GAUSSIAN PROCESSES FOR QUERY-EFFICIENT BOLTZMANN SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural samplers have achieved remarkable success in recent years, as efficient learning-based substitutes for traditional MCMC methods that significantly reduce the sampling cost after learning to sample. However, both neural samplers and MCMC methods remain sample-inefficient and therefore unsuitable for sampling problems where energy or likelihood evaluations are expensive. We develop a model-based sampling approach using a Gaussian process (GP) framework that trades computation for fewer evaluations by learning an energy surrogate and using its uncertainty to guide exploration. We distinguish two objectives: (1) producing accurate samples throughout exploration and (2) learning an accurate distribution for subsequent sampling. For (1), we analyze posterior-mean, optimistic and posterior-sampling policies using a KL-based notion of sampling regret and establish temperature-dependent bounds. For (2), we derive a Bayes-optimal one-step acquisition that selects evaluations to maximize the expected reduction in reverse KL divergence, together with practical approximations. Our experiments on low-dimensional sampling problems show the significant advantage of our model-based sampling method over neural samplers, MCMC methods, and existing GP-based sampling techniques.

1 INTRODUCTION

Often in the sciences, we have access to an energy function or a likelihood, and wish to generate samples from the corresponding Boltzmann distribution or Bayesian posterior. Both problems can be written as sampling from

$$p_\eta(dx) \propto \exp\big(-\eta E(x)\big)\, \mu(dx),$$

where $\eta > 0$ is an inverse temperature and μ is a reference measure; posterior sampling is recovered by taking E to be the negative log-likelihood, μ to be the prior, and $\eta = 1$.

Sampling problems vary substantially in both the cost of evaluating the energy and the dimensionality of the state space. For example, reported costs range from approximately 10^{-2} seconds for a gravitational-wave likelihood (Ashton, 2023), through 50 seconds for a forward solve in an eleven-parameter lung inverse problem (Dinkel et al., 2024), to 6.3×10^3 seconds for a density functional theory energy evaluation in an adsorption simulation (Goeminne et al., 2023). These examples span nearly six orders of magnitude in evaluation time. High dimensionality makes coverage of a multimodal target even harder, often requiring additional structure, a lower-dimensional representation or a multifidelity model (Peherstorfer et al., 2018). We focus on energies that can be modeled from a small number of expensive evaluations. In this regime, spending more computation deciding where to query can reduce the cost of the overall learning procedure.

Markov chain Monte Carlo (MCMC) methods construct a chain whose stationary distribution is the target. They generally do not learn a sampling policy, and generating more samples requires continuing the chain and making further energy evaluations. Neural samplers, such as Boltzmann generators, diffusion-based samplers, and GFlowNets, amortize sampling through the training of a generative model (Noé et al., 2019; Akhound-Sadegh et al., 2024; Bengio et al., 2023). Once trained, they can generate new samples efficiently, but their training can require a large number of energy evaluations. This trade-off is attractive when the energy is cheap to compute, and becomes more difficult to justify as its cost increases.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

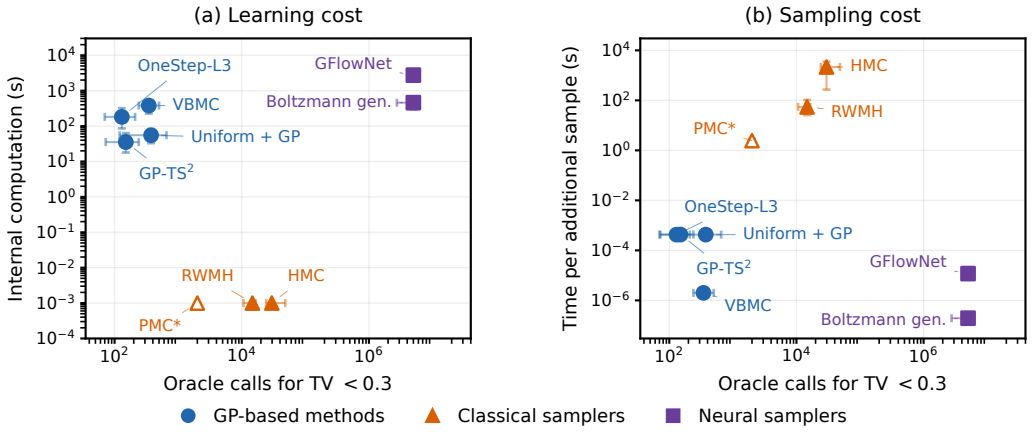

Figure 1: **Learning the energy trades computation for oracle efficiency.** The figure includes representative methods from GP-based, classical and neural sampler families and shows their performance on six 2D sampling tasks. Both panels use the number of oracle calls needed to first reach a total variation (TV) distance below 0.3. *The figures show that MCMC methods spend the least on internal computation but incur higher sampling costs when generating new samples. Neural samplers and GP-based methods take longer on internal computation but sample much faster, with GP-based methods being orders of magnitude more sample-efficient than the other two families in the problems considered.* **Left:** internal learning computation, excluding oracle time. **Right:** estimated cost per additional sample. GP surrogates need few evaluations and support sampling without the oracle; neural models amortize larger training budgets, while classical samplers continue to call the oracle. GP-based methods generate samples at the lowest cost when energy evaluation time dominates. Points and bars are medians and interquartile ranges over 18 runs; failures contribute terminal endpoints, and hollow/starred markers indicate fewer than half of the runs succeeded. Sampling costs assume 1 s per energy call. See Appendix C.4 for more details.

We advocate a model-based approach to sampling in this expensive-evaluation regime. The motivation is similar to that of model-based reinforcement learning (RL), where learning a model and using it for planning can substantially reduce the number of interactions with the environment required to learn a task (Deisenroth & Rasmussen, 2011; Chua et al., 2018). For sampling, modeling the energy can similarly allow the reuse of previous evaluations to guide the choice of the next query. This comes at the additional cost of fitting a model and computing the next query, but it can be a useful trade-off when energy evaluations are the main bottleneck. Figure 1 summarizes these trade-offs.

This principle is well established in Bayesian optimization (BO), where Gaussian process (GP) surrogates combine an estimate of an expensive objective with uncertainty about that estimate to select the next point to test. Acquisition functions use this uncertainty to balance the exploration of promising regions against the exploitation of good regions (Frazier, 2018). In low-dimensional control, PILCO similarly uses GP dynamics models to learn policies from only a few trials (Deisenroth & Rasmussen, 2011). GP surrogates have also been used for active posterior estimation and inference with expensive likelihoods (Kandasamy et al., 2015; El Gammal et al., 2023; Dinkel et al., 2024). With the growing interest in neural samplers, we argue that this model-based approach deserves particular attention when only a small number of energy evaluations is affordable. Our aim is to provide a theoretical foundation for choosing those evaluations, together with practical algorithms that use the uncertainty of a GP to learn the target distribution efficiently.

To choose where to evaluate, we need a notion of what a good query is. We measure the quality of the queries by their sampling regret: the cumulative reverse KL divergence between the distribution the queries are drawn from and the target. This objective can be related to KL-regularized RL through its correspondence with sampling. For instance, this relation is often used for fine-tuning foundation models, where the objective is to tilt a base model toward preferred outputs through a reward function (Korbak et al., 2022). Recent work on multi-armed sampling and KL-regularized RL studies the resulting exploration–exploitation trade-off (Pedramfar & Ravanbakhsh, 2026; Zhao

et al., 2025). At high temperatures, sampling from the current estimate can itself provide sufficient exploration, without an explicit exploration bonus. As the temperature decreases and the distribution concentrates near its optima, this implicit exploration can become inadequate and the problem approaches optimization. In the finite-armed setting, optimistic algorithms now have near-optimal cumulative-regret guarantees across temperature regimes (Ji et al., 2026).

There are two distinct objectives for an algorithm that learns to sample. In the cumulative setting, the queries themselves are samples of interest, and we measure their quality through the accumulated sampling regret, $\sum_t \eta^{-1} D_{\mathrm{KL}}(\pi_t \| p_\eta)$, where π_t is the distribution of the t-th query. In other settings, we allow an initial exploration phase and care primarily about the quality of the sampling distribution after that phase. The query distribution need not then resemble the target at each step. As in BO, an acquisition function can instead choose each query for its expected benefit to the final result. For sampling, this means targeting the error of the learned distribution, rather than the value of its best point.

Contributions. We explore algorithms derived from these two objectives using a GP surrogate of the energy.

- In the cumulative setting, we study posterior-mean sampling (GP-PM), upper-confidence sampling (GP-UCS), and double posterior sampling (GP-TS^2). These are GP realizations of greedy sampling, optimism, and posterior sampling, respectively. We establish cumulative sampling-regret bounds, forward-KL bounds for GP-PM and GP-TS^2, and final-policy guarantees under the stated assumptions.
- For learning a final sampling distribution, we derive the Bayes-optimal one-step acquisition for the expected reverse KL divergence of the updated posterior-mean Gibbs measure, together with three tractable approximations.
- We evaluate both families on six two-dimensional targets, alanine dipeptide in Ramachandran coordinates, a four-particle double-well system (DW-4), and a Planck-lite ΛCDM posterior. Our *OneStep* acquisition improves query efficiency on the toys, alanine and Planck-lite, while GP-UCS is more accurate on DW-4. We compare against MCMC and neural methods and show that our method can learn the energy with orders of magnitude fewer energy evaluations.

2 Related Work

Sampling methods differ in how they use evaluations of an unnormalized target. MCMC and SMC construct chains or particle populations through repeated target evaluations (Hastings, 1970; Del Moral et al., 2006), while neural samplers, including Boltzmann generators, diffusion samplers, and GFlowNets, learn generative models that amortize the cost of producing new samples (Noé et al., 2019; Vargas et al., 2023; Bengio et al., 2023). These approaches can also be combined with surrogate models, as in GFlowNet-based active learning (Jain et al., 2022). Our focus is on the regime where target evaluations are expensive and only a small number are affordable, making it useful to spend additional computation modeling the energy and deciding where to evaluate it. This regime further motivates the use of nonparametric models with uncertainty estimates.

This motivation is shared with Bayesian optimization and model-based RL, where probabilistic models guide the use of expensive observations or interactions (Frazier, 2018; Deisenroth & Rasmussen, 2011). For posterior estimation, BAPE, GPry, and VBMC already use GP surrogates to allocate likelihood evaluations and approximate the target distribution (Kandasamy et al., 2015; El Gammal et al., 2023; Acerbi, 2018). Related sequential-design methods derive acquisitions from uncertainty in the posterior approximation (Järvenpää et al., 2021), and theoretical work relates GP approximation error to posterior error (Stuart & Teckentrup, 2018).

Our work builds on this literature by studying query selection through the quality of the induced sampling distribution. BAPE's one-step expected-divergence criterion provides a direct precedent for this approach. Our OneStep acquisition targets the expected reverse KL divergence from the updated posterior-mean Gibbs measure to the unknown target. We derive this criterion and tractable approximations for selecting evaluations when the aim is to learn a sampler for subsequent use.

In the KL-regularized RL literature, recent work studies both simple and cumulative sampling regret, and how the need for exploration changes with temperature (Pedramfar & Ravanbakhsh, 2026; Zhao et al., 2025; Ji et al., 2026). We extend this analysis to GP models on continuous domains, prove sampling-regret bounds for GP-PM, GP-UCS and GP-TS^2, and derive a sample-efficient one-step policy that minimizes the KL divergence of the learned distribution.

3 METHOD

We learn a GP model of the energy and use it both to choose evaluations and to define a sampler. Gibbs policies sample from distributions based on the current model; OneStep selects queries for their expected improvement to the learned distribution.

Setup and objective. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the state space and $E : \mathcal{X} \to \mathbb{R}$ the unknown energy. We work with the reward $f = -E$, so larger rewards receive greater Gibbs weight. For a reference probability measure μ and inverse temperature $\eta > 0$, write

$$S_\eta(g)(\mathrm{d}x) = \frac{e^{\eta g(x)}\mu(\mathrm{d}x)}{\int e^{\eta g(x')}\mu(\mathrm{d}x')}.$$

Our goal is to sample from $p_\eta = S_\eta(f) = S_\eta(-E)$ using an unbiased energy oracle. Negating its output at x_t gives $y_t = f(x_t) + \epsilon_t$, with conditionally mean-zero noise. A policy selects $x_t \sim \pi_t$ from the observation history. When π_t is Gibbs, write $\pi_t = S_\eta(g_t)$; the policy may choose g_t explicitly or implicitly.

When the queries themselves are samples of interest, we measure their quality by cumulative sampling regret,

$$\mathrm{SRegret}(T) = \frac{1}{\eta}\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_t \,\big\|\, p_\eta\big). \tag{1}$$

GP surrogate. With a zero-mean GP prior, kernel k satisfying $k(x,x) \le 1$, and noise or regularization parameter $\lambda > 0$, the posterior mean and covariance after $t-1$ observations are

$$\begin{aligned} m_{t-1}(x) &= k_{t-1}(x)^\top (K_{t-1} + \lambda I)^{-1} y_{1:t-1}, \\ \Sigma_{t-1}(x,x') &= k(x,x') - k_{t-1}(x)^\top (K_{t-1} + \lambda I)^{-1} k_{t-1}(x'), \end{aligned} \tag{2}$$

For a nonzero prior mean, apply (2) to centered observations and add the prior mean back. Here $K_{t-1} = [k(x_i,x_j)]_{i,j<t}$, $k_{t-1}(x) = (k(x_i,x))_{i<t}$, and $s^2_{t-1}(x) = \Sigma_{t-1}(x,x)$. Let γ_T be the maximum information gain of the kernel after T observations, $\gamma_T = \max_{x_1,\dots,x_T} \frac{1}{2}\log\det(I + \lambda^{-1}K_T)$, and b_t the confidence radius of Appendix A, of order $\sqrt{\gamma_{t-1} + \log(1/\delta)}$ for an RKHS-bounded f; the *uniform confidence event* is the event $|f(x) - m_{t-1}(x)| \le b_t\, s_{t-1}(x)$ for all x and t simultaneously, which holds with probability at least $1-\delta$ under the RKHS and noise assumptions (Chowdhury & Gopalan, 2017).

For the analysis, we assume that f takes values in a known range of width C, say $[0, C]$, and write $\Pi_{[0,C]}$ for clipping to this range.

3.1 GIBBS POLICIES

We study three policies, corresponding to greedy sampling, optimism and posterior sampling. Their scores are

$$\begin{aligned} g_t^{\mathrm{PM}} &= \Pi_{[0,C]}(m_{t-1}), \quad & g_t^{\mathrm{UCS}} &= \Pi_{[0,C]}(m_{t-1} + b_t s_{t-1}), \\ g_t^{\mathrm{TS}^2} &= \tilde{f}_t, & \tilde{f}_t &\sim \mathcal{GP}(m_{t-1}, \Sigma_{t-1}). \end{aligned} \tag{3}$$

GP-PM samples from the current estimate of the target. It has no explicit exploration bonus, but can still visit uncertain states if the estimated distribution assigns them sufficient mass. **GP-UCS** adds an uncertainty bonus $b_t s_{t-1}$ to the reward, increasing the probability of states whose energy may be lower than predicted. **GP-TS^2** first draws a plausible reward function and then draws a state from its Gibbs distribution. Increasing η concentrates these policies on states with higher scores.

Theorem 1 (Regret of the Gibbs policies; informal)**.** *Under the confidence assumptions of Appendix A, and choosing $\delta = T^{-2}$,*

$$\mathbb{E}[\mathrm{SRegret}_{\mathrm{PM}}(T)] = \widetilde{O}\big(\eta e^{\eta C} b_T^2 \gamma_T\big),$$
$$\mathbb{E}[\mathrm{SRegret}_{\mathrm{UCS}}(T)] = \widetilde{O}\big(\min\{b_T\sqrt{T\gamma_T},\ \eta b_T^2 \gamma_T\}\big).$$

The bounded Bayesian version of GP-TS^2 has the same order of cumulative regret as GP-UCS. GP-PM and bounded GP-TS^2 also admit forward-KL guarantees. For fixed η and C, the expected KL error of each final query policy $\pi_{T+1} = S_\eta(g_{T+1})$ is $\widetilde{O}(b_{T+1}^2\gamma_T/T)$, with constants that can grow exponentially in ηC.

The bounds explain why temperature matters. For a radial basis function (RBF) kernel on a compact domain, b_T and γ_T grow only polylogarithmically, so average regret vanishes at fixed temperature (Srinivas et al., 2010). The GP-PM bound nevertheless deteriorates exponentially as the target concentrates, through $e^{\eta C}$. Optimism and posterior sampling avoid this factor in their cumulative bounds, reflecting the value of explicit exploration. This advantage does not remove the exponential constants in the final-policy bounds. Appendix A gives the precise statements and proofs. The Gaussian paths in (3) do not satisfy the bounded-posterior assumption of the GP-TS^2 theorem; its guarantee concerns the bounded model.

3.2 One-step lookahead

When queries are used to learn a sampler for later use, OneStep selects the evaluation with the largest expected reduction in the updated posterior-mean policy's reverse KL error.

Here we assume a correctly specified Gaussian regression model with noise $\varepsilon \sim \mathcal{N}(0, \lambda)$ and use the unclipped policy $\pi_t = S_\eta(m_{t-1})$. An observation at x updates the GP as follows, with $Z \sim \mathcal{N}(0,1)$ denoting its standardized prediction error:

$$m_t = m_{t-1} + Z r_x, \qquad \Sigma_t = \Sigma_{t-1} - r_x r_x^\top, \qquad r_x(x') = \frac{\Sigma_{t-1}(x', x)}{\sqrt{\Sigma_{t-1}(x,x) + \lambda}}. \tag{4}$$

The function r_x describes how the query can change predictions across the domain. The observation determines the mean update; the covariance reduction is known in advance.

Given the history $\mathcal{H}_t = \{(x_s, y_s)\}_{s<t}$, the one-step objective is

$$J_t(x) = \mathbb{E}\Big[\tfrac{1}{\eta} D_{\mathrm{KL}}\big(S_\eta(m_{t-1} + Z r_x) \,\big\|\, S_\eta(f)\big) \,\Big|\, \mathcal{H}_t\Big]. \tag{5}$$

The expectation is over the joint posterior of the function and observation. The following result expresses query value using the current GP.

Theorem 2 (One-step acquisition; see Appendix A.5.)**.** *Let $J_t^0 = \mathbb{E}[\eta^{-1} D_{\mathrm{KL}}(\pi_t \| S_\eta(f)) \mid \mathcal{H}_t]$ be the current expected error. Then $J_t(x) = J_t^0 - a_t(x)$, where*

$$a_t(x) = \frac{1}{\eta}\mathbb{E}_{Z\sim\mathcal{N}(0,1)} \log \mathbb{E}_{X'\sim\pi_t}\big[e^{\eta Z h_x(X')}\big] \geq 0, \qquad h_x = r_x - \mathbb{E}_{\pi_t}[r_x]. \tag{6}$$

Thus, maximizing a_t minimizes the expected error after one observation.

OneStep-L1 keeps the leading term in a small-η expansion,

$$a_t^{(1)}(x) = \frac{\eta}{2}\frac{\mathrm{Var}_{X'\sim\pi_t}\big(\Sigma_{t-1}(X', x)\big)}{\Sigma_{t-1}(x,x) + \lambda}. \tag{7}$$

It favors queries whose information affects different parts of the current distribution differently. **OneStep-L2** adds a fourth-cumulant correction, which captures departures from a Gaussian distribution of the centered update $h_x(X')$. This improves the formal expansion but need not improve accuracy at a fixed temperature. **OneStep-L3** evaluates the nonlinear expression in (6) by Gauss–Hermite quadrature, a weighted average over representative values of the Gaussian variable Z. All three approximate expectations over X' using samples from π_t.

3.3 IMPLEMENTATION

We fit exact RBF GPs using BoTorch and GPyTorch (Gardner et al., 2018; Balandat et al., 2020), with standardized observations and hyperparameters refitted after each batch. The implementation models energy, whose mean is the negative of the reward mean above. High observed energies are clipped before fitting, separately from the policy clipping used in the analysis. The experiments use dimensionless energies, $\eta = 1$ and uniform reference measures on bounded domains, with Lebesgue measure on the centered coordinate space for DW-4. GP-UCS uses a fixed exploration coefficient 0.5 in place of the theoretical b_t.

For the two-dimensional toys and alanine, we draw Gibbs queries from normalized grid probabilities and add uniform jitter within each selected cell. For DW-4 and Planck-lite, we use random-walk Metropolis on the surrogate. GP-TS^2 draws a fresh approximate posterior path for each query using Matheron's rule (Wilson et al., 2020). Once learning is complete, these same numerical samplers can generate samples from the posterior-mean density without further energy evaluations.

OneStep represents the current density with a fixed set of particles and optimizes its estimated acquisition from several particle-based and uniform starting points. To choose a batch, it reduces the working covariance after each selection while keeping the mean and particles fixed. This approximation, known as kriging believer, discourages redundant queries without requiring their observations in advance. We evaluate each selected batch and refit the GP before the next round. The guarantees concern sequential queries with fixed model assumptions; the experiments use batched queries, refitted hyperparameters and numerical approximations. Appendix B gives pseudocode, acquisition estimators and experimental settings.

4 EXPERIMENTS

We compare how efficiently Gibbs policies and OneStep learn sampling distributions on six two-dimensional toy densities, alanine dipeptide in Ramachandran coordinates, DoubleWell-4 (DW-4), and a Planck-lite ΛCDM posterior. These tasks test when selecting informative queries improves on sampling from the current estimate.

We compare with the LogExp query rule from GPry, a GP-based method for learning posteriors with expensive likelihoods (El Gammal et al., 2023). LogExp balances predicted density and uncertainty. For this comparison, we use LogExp to choose queries within our GP implementation and call the result GPry-LogExp. Other baselines include BAPE-EV (Kandasamy et al., 2015), which queries where the unnormalized density is most uncertain, and VBMC (Acerbi, 2018), which uses a GP to learn a variational posterior. We also compare with classical and neural samplers. "+ GP" means fitting a GP to the inputs where a classical sampler requested energy evaluations, including rejected proposals, then scoring the resulting learned density. This tests whether useful chain samples also provide useful data for learning the energy.

For learned densities, we report Jeffreys divergence $D_J(p, q) = D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|p)$, where p is the target and q the learned density. Its two directions penalize both missing target mass and assigning mass where the target density is low. We use grid integration on the toys and torsions, and reference-bank reweighting on Planck. To compare densities with sample histograms, we use total variation (TV). See Appendix B for details.

4.1 TWO-DIMENSIONAL TOY DENSITIES

The targets consist of a mixture of eight Gaussians, two moons, two spirals, checkerboard, rings and pinwheel (see Figure 5). GP acquisitions start with 20 uniform initial points for each of three seeds and use a batch size of 10 with a total budget of $2{,}000$ calls.

Across the six targets, OneStep generally learns accurate densities with fewer queries than the competing designs; Figure 2 illustrates this on rings. Its three approximation levels have similar accuracy, with no clear benefit from the more elaborate versions. GP-TS^2 explores more reliably than GP-PM and GP-UCS, which can sometimes miss target mass. We also compare online regret in Figure 6, which shows that GP-TS^2 scales generally better than GP-PM and GP-UCS. Classical "+ GP" designs are generally less query-efficient than OneStep for recovering the energy. Figure 1 il-

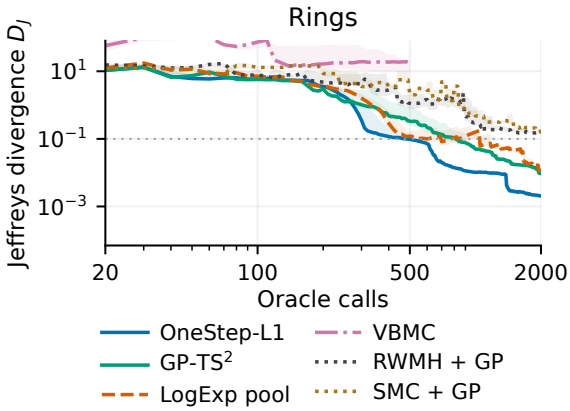

Figure 2: Density learning on rings. Jeffreys divergence D_J (lower is better), integrated on a 400×400 grid: median and interquartile range over three seeds. The dashed line marks $D_J = 0.1$; "+ GP" fits a surrogate to classical queries. VBMC stops at 500 calls; other runs use 2,000. Figure 7 in the appendix gives the full six-target comparison.

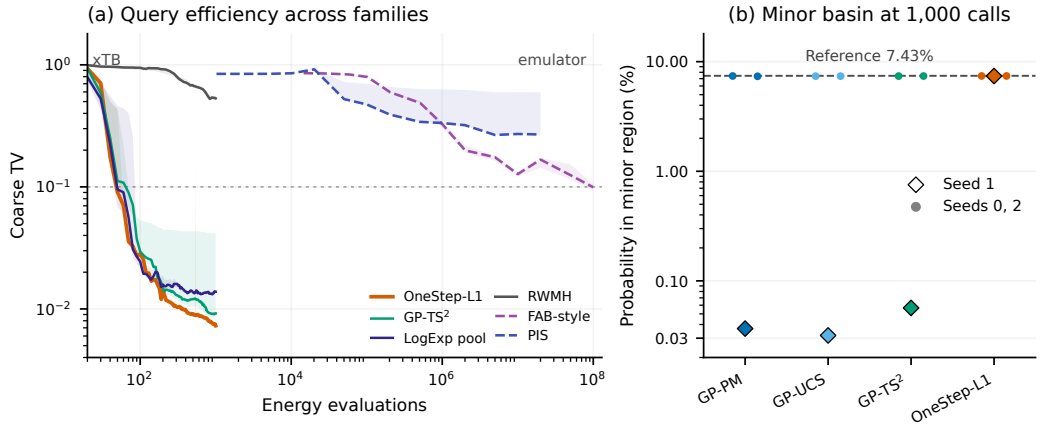

Figure 3: Learning the relaxed-energy alanine target at 300 K. Left: total variation on a 36×36 torsion grid, median and interquartile range over three seeds. GPs are scored as densities; random-walk Metropolis (RWMH) and neural methods use sample histograms. Solid curves count real energy queries, dashed curves emulator evaluations (excluding 20,736 real queries to build the emulator). Right: mass at 1,000 real queries in $30° < \phi < 120°$, $-120° < \psi < 0°$, whose reference mass is 7.43%. Diamonds identify the common seed in which the Gibbs policies miss this region. Appendix D gives complete results and protocols.

lustrates the computational trade-off: OneStep uses few oracle calls but spends more time fitting and selecting queries, while GP-TS^2 and LogExp pool require less internal computation. For classical samplers, internal computation is shown at a 10^{-3} s plotting floor, and their sampling cost assumes 1 s per oracle call. See Appendix C.4 for assumptions and measurement details.

4.2 Alanine dipeptide in Ramachandran coordinates

Alanine dipeptide provides a molecular target described by two backbone rotation angles, (ϕ, ψ), known as Ramachandran coordinates. We learn the Boltzmann density of its relaxed energy at 300 K, using the approximate quantum-chemical model GFN2-xTB (Bannwarth et al., 2019). Each real-oracle query fixes these angles and relaxes the remaining molecular coordinates. Because this procedure does not integrate over their possible configurations, the target omits their entropy and is not a torsional free-energy surface. GP and classical runs use three seeds, up to 1,000 real queries and a 72×72 reference grid. Neural runs use a differentiable emulator built from a denser scan costing 20,736 real queries; their training budgets count emulator evaluations. The emulator also supplies torsion derivatives, which are required for the training of some neural samplers.

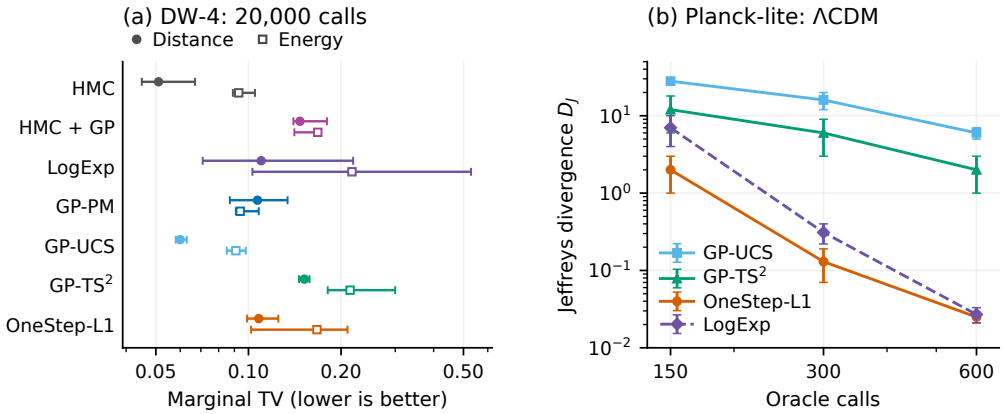

Figure 4: Oracle-budget comparisons on DW-4 and Planck-lite; lower is better. Left: TV of pair-distance and energy histograms at 20,000 calls, median and interquartile range (six seeds for acquisitions, five for HMC and HMC + GP). These marginal scores do not measure full-state TV. Right: Jeffreys divergence $D_J = D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|p)$ estimated from a Planck reference bank, mean $\pm$ standard error over five training seeds. Appendices E and F give complete results, protocols and estimator details.

OneStep learns an accurate torsion density from few real energy queries (Figure 3, left). The neural implementations use far more emulator evaluations and remain less accurate in this experiment.

We observe that Gibbs policies struggle with mode discovery. From one shared initial design, all three nearly miss a smaller conformational basin that OneStep consistently finds (Figure 3, right). Sampling from the current surrogate can leave a policy confined to the major basin when the smaller one is initially missed. This failure persists even with exact sampling, showing that it is not a side effect of an RWMH mixing issue.

4.3 DoubleWell-4

DW-4 describes four particles in two dimensions with pairwise double-well interactions (Köhler et al., 2020). The GP is fed the center-of-mass-subtracted coordinates. Note that this energy has many more symmetries but the GP does not account for them. We assess generated configurations through their pair distances and energies, two physically meaningful summaries of the target.

GP-UCS gives the strongest GP result, comparable to Hamiltonian Monte Carlo (HMC) on the measured metrics (Figure 4, left). Fitting a GP to the energy evaluations collected by HMC reduces the quality of the samples produced. Training on MCMC-generated samples is not sufficient to learn a high-fidelity energy surrogate. The best neural sampler, FAB, scores better, but at a much higher energy cost, and cannot be trained in low energy budget settings.

4.4 Planck-lite posterior inference

Our final task learns the posterior of the six parameters of ΛCDM, the standard cosmological model with a cosmological constant and cold dark matter, from Planck measurements of the cosmic microwave background. Each of the 600 Planck-lite oracle calls includes the cosmological forward calculation (Planck Collaboration, 2020b;a). We fix the search region to eight standard deviations around the reference means. We estimate divergence using a bank of reference samples generated by MCMC.

Figure 4 (right) shows that OneStep learns an accurate posterior with limited evaluations, reaching performance comparable to GPry-LogExp. GP-PM, GP-UCS, and GP-TS^2 struggle more when high efficiency is required.

5 DISCUSSION

Cost trade-off. Although we mainly experimented on cheap-to-compute energies, we stress that our methods should be applied when the energy is expensive. The trade-off becomes interesting when one energy call requires seconds or more of compute, as the GP can take a while to fit, and scales poorly with data and dimensionality. Our OneStep acquisition performs the best in three out of the four benchmarks tested, measured by final KL divergence, and performs the best across many of the different two-dimensional toy examples (see Figure 7).

Limitations. We study low-dimensional targets whose energy can be learned from a dataset small enough for exact GP fitting to remain practical. Fitting costs $\mathcal{O}(N^3)$ for N observations, which limits the training budget. Sparse approximations could extend this range, but this is beyond the scope of this paper. Neural samplers can work in much higher-dimensional spaces (Akhound-Sadegh et al., 2024), a regime that our experiments do not evaluate, but we expect our methods to perform poorly, as high-dimensional energies would require more data to learn.

Generating samples from the learned energy still requires a numerical sampler and can suffer from bad mixing. Starting the chain from previous training points, as we do, should help, as those are often near regions of low energy. As is often the case in the sciences, our theoretical results impose conditions on the target energy that may not be satisfied in practice.

AI STATEMENT

AI tools were used extensively throughout this work, including experimental design, code development, running experiments, analyzing results, developing theoretical arguments and proofs, generating figures, and drafting, revising, and polishing the manuscript. Their use extended to both substantive research contributions and presentation. The authors take full responsibility for the methods, proofs, experimental results, figures, and claims presented in the paper.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24, pp. 2312–2320. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

Luigi Acerbi. Variational Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8222–8232. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/747c1bcceb6109a4ef936bc70cfe67de-Abstract.html.

Luigi Acerbi. Variational Bayesian Monte Carlo with noisy likelihoods. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8211–8222. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/5d40954183d62a82257835477ccad3d2-Abstract.html.

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from Boltzmann densities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 760–786. PMLR, 2024. URL https://proceedings.mlr.press/v235/akhound-sadegh24a.html.

Michael Samuel Albergo and Eric Vanden-Eijnden. NETS: A non-equilibrium transport sampler. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 1026–1055. PMLR, 2025. URL https://proceedings.mlr.press/v267/albergo25a.html.

Gregory Ashton. Gaussian processes for glitch-robust gravitational-wave astronomy. *Monthly Notices of the Royal Astronomical Society*, 520(2):2983–2994, 2023. doi: 10.1093/mnras/stad341. URL https://doi.org/10.1093/mnras/stad341.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005. URL https://jmlr.org/papers/v6/banerjee05b.html.

Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. GFN2-xTB—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of Chemical Theory and Computation*, 15(3):1652–1671, 2019. doi: 10.1021/acs.jctc.8b01176.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27381–27394, Red Hook, NY, USA, 2021. Curran Associates, Inc. ISBN 9781713845393. URL https://proceedings.neurips.cc/paper/2021/hash/e614f646836aaed9f89ce58e837e2310-Abstract.html.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL https://www.jmlr.org/papers/v24/22-0364.html.

Diego Blas, Julien Lesgourgues, and Thomas Tram. The Cosmic Linear Anisotropy Solving System (CLASS). II. approximation schemes. *Journal of Cosmology and Astroparticle Physics*, 2011 (07):034, 2011. doi: 10.1088/1475-7516/2011/07/034. URL https://arxiv.org/abs/1104.2933.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://arxiv.org/abs/1601.00670.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled Langevin diffusions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a71c1931d3fb8ba564f7458d0657d0b1-Abstract-Conference.html.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 844–853. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/chowdhury17a.html.

J. Andrés Christen and Colin Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005. doi: 10.1198/106186005X76983. URL https://doi.org/10.1198/106186005X76983.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. URL https://arxiv.org/abs/cs/9603104.

Patrick R. Conrad, Youssef M. Marzouk, Natesh S. Pillai, and Aaron Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016. doi: 10.1080/01621459.2015.1096787. URL https://doi.org/10.1080/01621459.2015.1096787.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012. doi: 10.1111/j.1467-9469.2011.00756.x. URL https://doi.org/10.1111/j.1467-9469.2011.00756.x.

Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 465–472, 2011. URL https://icml.cc/2011/papers/323_icmlpaper.pdf.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. doi: 10.1111/j.1467-9868.2006.00553.x. URL https://doi.org/10.1111/j.1467-9868.2006.00553.x.

Maximilian Dinkel, Carolin M. Geitner, Gil Robalo Rei, Jonas Nitzler, and Wolfgang A. Wall. Solving Bayesian inverse problems with expensive likelihoods using constrained Gaussian processes and active learning. *Inverse Problems*, 40(9):095008, 2024. doi: 10.1088/1361-6420/ad5eb4. URL https://doi.org/10.1088/1361-6420/ad5eb4.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216–222, 1987. doi: 10.1016/0370-2693(87)91197-X.

Jonas El Gammal, Nils Schöneberg, Jesús Torrado, and Christian Fidler. Fast and robust Bayesian inference using Gaussian processes with GPry. *Journal of Cosmology and Astroparticle Physics*, 2023(10):021, 2023. doi: 10.1088/1475-7516/2023/10/021. URL https://arxiv.org/abs/2211.02045.

Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009. doi: 10.1287/ijoc.1080.0314. URL https://doi.org/10.1287/ijoc.1080.0314.

Peter I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. doi: 10.48550/arXiv.1807.02811. URL https://arxiv.org/abs/1807.02811.

Peter I. Frazier, Warren B. Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008. doi: 10.1137/070693424. URL https://doi.org/10.1137/070693424.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html.

Charles J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163. Interface Foundation of North America, 1991. URL https://hdl.handle.net/11299/58440.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In Yoel Tenne and Chi-Keong Goh (eds.), *Computational Intelligence in Expensive Optimization Problems*, pp. 131–162. Springer, 2010. doi: 10.1007/978-3-642-10701-6_6. URL https://link.springer.com/chapter/10.1007/978-3-642-10701-6_6.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11546–11583. PMLR, 2023. URL https://proceedings.mlr.press/v202/go23a.html.

Ruben Goeminne, Louis Vanduyfhuys, Veronique Van Speybroeck, and Toon Verstraelen. DFT-quality adsorption simulations in metal–organic frameworks enabled by machine learning potentials. *Journal of Chemical Theory and Computation*, 19(18):6313–6325, 2023. doi: 10.1021/acs.jctc.3c00495. URL https://doi.org/10.1021/acs.jctc.3c00495.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Tom Gunter, Michael A. Osborne, Roman Garnett, Philipp Hennig, and Stephen J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2789–2797. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/a0d08267a0fcee6970544a6d12286691-Abstract.html.

Michael U. Gutmann and Jukka Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47, 2016. URL https://www.jmlr.org/papers/v17/15-017.html.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334940.

Aaron J. Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S. Levine, Brandon M. Wood, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22204–22237. PMLR, 2025. URL https://proceedings.mlr.press/v267/havens25a.html.

Jiajun He, Wenlin Chen, Mingtian Zhang, David Barber, and José Miguel Hernández-Lobato. Training neural samplers with reverse diffusive KL divergence. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 5167–5175. PMLR, 2025. URL https://proceedings.mlr.press/v258/he25a.html.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL http://jmlr.org/papers/v15/hoffman14a.html.

Zhuoyue Huang, Freddie Bickford Smith, and Tom Rainforth. Loss-driven Bayesian active learning, 2026. URL https://arxiv.org/abs/2604.11995. arXiv:2604.11995.

Bobby Huggins, Chengkun Li, Marlon Tobaben, Mikko J. Aarnos, and Luigi Acerbi. PyVBMC: Efficient Bayesian inference in Python. *Journal of Open Source Software*, 8(86):5428, 2023. doi: 10.21105/joss.05428. URL https://doi.org/10.21105/joss.05428.

Himchan Hwang, Hyeokju Jeong, Dong Kyu Shin, Che-Sang Park, Sehee Kweon, Sangwoong Yoon, and Frank C. Park. Value gradient sampler: Learning invariant value functions for equivariant diffusion sampling. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3979–3987. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/hwang26a.html.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9786–9801. PMLR, 2022. URL https://proceedings.mlr.press/v162/jain22a.html.

Marko Järvenpää and Jukka Corander. Approximate Bayesian inference from noisy likelihoods with Gaussian process emulated MCMC. *Journal of Machine Learning Research*, 25(366):1–55, 2024. URL https://jmlr.org/papers/v25/21-0421.html.

Marko Järvenpää, Michael U. Gutmann, Arijus Pleska, Aki Vehtari, and Pekka Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *Bayesian Analysis*, 14(2):595–622, 2019. doi: 10.1214/18-BA1121. URL https://doi.org/10.1214/18-BA1121.

Marko Järvenpää, Michael U. Gutmann, Aki Vehtari, and Pekka Marttinen. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16(1):147–178, 2021. doi: 10.1214/20-BA1200. URL https://doi.org/10.1214/20-BA1200.

Kaixuan Ji, Qingyue Zhao, Heyang Zhao, Qiwei Di, and Quanquan Gu. Near-optimal regret for KL-regularized multi-armed bandits, 2026. URL https://arxiv.org/abs/2603.02155. arXiv:2603.02155.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, December 1998. ISSN 0925-5001. doi: 10.1023/A:1008306431147. URL https://doi.org/10.1023/A:1008306431147.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. Bayesian active learning for posterior estimation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3605–3611, 2015. URL https://www.ijcai.org/Proceedings/15/Papers/507.pdf.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: Exact likelihood generative learning for symmetric densities. In Hal Daumé, III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kohler20a.html.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16203–16220, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/67496dfa96afddab795530cc7c69b57a-Abstract-Conference.html.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-García, Lena Nehale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18269–18300. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lahlou23a.html.

Johannes U. Lange. nautilus: boosting Bayesian importance nested sampling with deep learning. *Monthly Notices of the Royal Astronomical Society*, 525(2):3181–3194, 2023. doi: 10.1093/mnras/stad2441. URL https://doi.org/10.1093/mnras/stad2441.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/b3ba8f1bee1238a2f37603d90b58898d-Abstract.html.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 06 1953. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XCTVFJwS9LJ.

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001. doi: 10.1023/A:1008923215028. URL https://doi.org/10.1023/A:1008923215028.

Willie Neiswanger, Lantao Yu, Shengjia Zhao, Chenlin Meng, and Stefano Ermon. Generalizing Bayesian optimization with decision-theoretic entropies. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/file/8451a20c5a7e0ee5671dda28f7daf7f3-Paper-Conference.pdf.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. URL https://doi.org/10.1126/science.aaw1147.

A. O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991. doi: 10.1016/0378-3758(91)90002-V. URL https://www.sciencedirect.com/science/article/pii/037837589190002V.

Mohammad Pedramfar and Siamak Ravanbakhsh. Multi-armed sampling problem and the end of exploration. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3691–3699. PMLR, 2026. URL https://proceedings.mlr.press/v300/pedramfar26a.html.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018. doi: 10.1137/16M1082469. URL https://doi.org/10.1137/16M1082469.

Planck Collaboration. Planck 2018 results. VIII. gravitational lensing. *Astronomy & Astrophysics*, 641:A8, 2020a. doi: 10.1051/0004-6361/201833886. URL https://www.aanda.org/articles/aa/full_html/2020/09/aa33886-18/aa33886-18.html.

Planck Collaboration. Planck 2018 results. V. CMB power spectra and likelihoods. *Astronomy & Astrophysics*, 641:A5, 2020b. doi: 10.1051/0004-6361/201936386. URL https://www.aanda.org/articles/aa/full_html/2020/09/aa36386-19/aa36386-19.html.

Jixiang Qing, Henry Moss, and Matthias Sachs. Active learning for Gaussian process regression under self-induced Boltzmann weights, 2026. URL https://arxiv.org/abs/2605.10654. arXiv:2605.10654.

Carl Edward Rasmussen. Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In *Bayesian Statistics 7*. Oxford University Press, 2003. doi: 10.1093/oso/9780198526155.003.0045. URL https://mlg.eng.cam.ac.uk/pub/pdf/Ras03.pdf.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011. doi: 10.1137/100801275. URL https://doi.org/10.1137/100801275.

Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Nikolay Malkin. Improved off-policy training of diffusion samplers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/93c936b9e492def9c00782cab79dbc6d-Paper-Conference.pdf.

Michael Sinsbeck and Wolfgang Nowak. Sequential design of computer experiments for the solution of Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):640–664, 2017. doi: 10.1137/15M1047659. URL https://doi.org/10.1137/15M1047659.

John Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, 2006.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, ICML'10, pp. 1015–1022, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Andrew M. Stuart and Aretha L. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018. doi: 10.1090/mcom/3244. URL https://doi.org/10.1090/mcom/3244.

Robert H. Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607–2609, Nov 1986. doi: 10.1103/PhysRevLett.57.2607. URL https://link.aps.org/doi/10.1103/PhysRevLett.57.2607.

Jesús Torrado and Antony Lewis. Cobaya: Code for Bayesian analysis of hierarchical physical models. *Journal of Cosmology and Astroparticle Physics*, 2021(05):057, 2021. doi: 10.1088/1475-7516/2021/05/057. URL https://arxiv.org/abs/2005.05290.

Jesús Torrado, Nils Schöneberg, and Jonas El Gammal. Parallelized acquisition for active learning using Monte Carlo sampling, 2023. URL https://arxiv.org/abs/2305.19267. arXiv:2305.19267.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10292–10302. PMLR, 2020. URL https://proceedings.mlr.press/v119/wilson20a.html.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y.

Heyang Zhao, Chenlu Ye, Wei Xiong, Quanquan Gu, and Tong Zhang. Logarithmic regret for online KL-regularized reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77864–77884. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhao25ac.html.

A GP SAMPLING THEORY

This appendix establishes the guarantees for the two objectives in the main text: producing useful samples while learning the energy, and choosing evaluations to improve a sampler for later use. For the Gibbs policies, the central argument is that uncertainty in the energy bounds the sampling error, while each observation reduces that uncertainty. We first prove the required confidence bound, then analyze the three policies and the quality of their final distributions. We finish by deriving the one-step acquisition and its practical approximations.

Model and common notation. We use bandit notation in the regret analysis: $\mathcal{A}$ denotes the measurable state space $\mathcal{X}$ of the main text, and an action A_t is a query location x_t. Let μ be a probability measure on $\mathcal{A}$. The reward is the negative energy, $f = -E$, so larger rewards receive greater Gibbs weight. For the regret bounds, we assume that f has a known range of width C and shift that range to $[0, C]$; this shift leaves its Gibbs distribution unchanged. The norm and confidence assumptions below apply to this shifted function. An evaluation at A_t returns

$$Y_t = f(A_t) + \varepsilon_t.$$

Let k be a positive-semidefinite kernel satisfying $k(a, a) \le 1$, let $\mathcal{H}_k$ be its reproducing kernel Hilbert space (RKHS), and fix a regularization parameter $\lambda > 0$. The RKHS specifies the class of reward functions whose smoothness is controlled by the kernel; its norm measures the size of a function within that class. Write $H_t = \{(A_s, Y_s)\}_{s<t}$ for the history available before round t. The GP mean and pointwise variance computed from this history are

$$m_{t-1}(a) = k_{t-1}(a)^\top (K_{t-1} + \lambda I)^{-1} Y_{1:t-1},$$
$$s^2_{t-1}(a) = k(a, a) - k_{t-1}(a)^\top (K_{t-1} + \lambda I)^{-1} k_{t-1}(a),$$

where $K_{t-1} = [k(A_i, A_j)]_{i,j<t}$ and $k_{t-1}(a) = (k(A_1, a), \dots, k(A_{t-1}, a))^\top$. These expressions use a zero prior mean. The kernel and regularization parameter are held fixed throughout the analysis, so each update adds an observation to the same regression model.

The frequentist results treat f as fixed and assume a bound B on its RKHS norm. They also assume that the observation noise has sub-Gaussian scale R, as specified below. The resulting confidence interval must contain f at every location and every round, including locations chosen adaptively. Define the maximum information gain γ_n and confidence radius b_t by

$$\gamma_n = \max_{a_1,\dots,a_n \in \mathcal{A}} \frac{1}{2} \log\det(I + \lambda^{-1} K_n), \qquad b_t = B + \frac{R}{\sqrt{\lambda}} \sqrt{2\big(\gamma_{t-1} + \log(1/\delta)\big)}.$$

The quantity γ_n summarizes how much n observations can reveal about a function modeled by the kernel. It also controls the total posterior variance at queried locations, which is the link to sampling regret used below. The parameter $\delta \in (0, 1)$ sets the probability that the confidence bound fails.

Theorem 3 (Uniform GP confidence bound)**.** *Suppose that $\|f\|_{\mathcal{H}_k} \le B$ and that, conditionally on H_t and A_t, the noise is mean zero and R-sub-Gaussian:*

$$\mathbb{E}[\exp(z\varepsilon_t) \mid H_t, A_t] \le \exp(z^2 R^2/2) \qquad \textit{for every } z \in \mathbb{R}.$$

Then, with probability at least $1 - \delta$, simultaneously for every $t \ge 1$ and $a \in \mathcal{A}$,

$$|f(a) - m_{t-1}(a)| \le b_t s_{t-1}(a).$$

The noise condition is imposed after conditioning on the chosen query. It therefore permits adaptive query selection and does not require the observation noise to be exactly Gaussian. The conclusion is a single confidence event covering all rounds and locations, rather than a separate probability statement for each prediction.

Proof. The prediction error has two sources: regularization bias and accumulated observation noise. We bound them separately in the kernel's feature space. Let $\varphi(a)$ be a feature map for k, so that $k(a, a') = \langle \varphi(a), \varphi(a') \rangle$, and identify f with an element of $\mathcal{H}_k$. Put

$$V_{t-1} = \lambda I + \sum_{s<t} \varphi(A_s) \otimes \varphi(A_s), \qquad S_{t-1} = \sum_{s<t} \varepsilon_s \varphi(A_s).$$

Here $u \otimes u$ denotes the outer-product operator sending a vector v to $u\langle u, v\rangle$. Thus V_{t-1} records the observed feature directions, with regularization ensuring invertibility, and S_{t-1} is the accumulated noise in those directions. We use the weighted norm $\|v\|_M^2 = \langle v, Mv\rangle$ for a positive operator M. The kernel-ridge identities then give

$$m_{t-1}(a) = \left\langle \varphi(a), V_{t-1}^{-1} \sum_{s<t} Y_s \varphi(A_s) \right\rangle, \qquad s_{t-1}^2(a) = \lambda \|\varphi(a)\|_{V_{t-1}^{-1}}^2.$$

Consequently,

$$|f(a) - m_{t-1}(a)| \le \left(\sqrt{\lambda}\|f\|_{\mathcal{H}_k} + \|S_{t-1}\|_{V_{t-1}^{-1}}\right) \|\varphi(a)\|_{V_{t-1}^{-1}}. \tag{A.1}$$

Indeed, the regularization bias is $\lambda V_{t-1}^{-1} f$, and Cauchy–Schwarz in the V_{t-1} norm bounds its contribution by $\sqrt{\lambda}\|f\|_{\mathcal{H}_k}\|\varphi(a)\|_{V_{t-1}^{-1}}$; the remaining contribution is the noise term $V_{t-1}^{-1} S_{t-1}$.

The noise term is controlled by the self-normalized sub-Gaussian inequality (Abbasi-Yadkori et al., 2011; Chowdhury & Gopalan, 2017). Here "self-normalized" means that the accumulated noise is measured using V_{t-1}^{-1}, which accounts for the locations already observed. With probability at least $1 - \delta$, simultaneously for all t,

$$\|S_{t-1}\|_{V_{t-1}^{-1}} \le R\sqrt{2\log\left(\frac{\det(V_{t-1})^{1/2}}{\det(\lambda I)^{1/2}\delta}\right)}. \tag{A.2}$$

Here and below, determinants of operators on $\mathcal{H}_k$ are understood on the finite-dimensional span of $\{\varphi(A_1), \ldots, \varphi(A_{t-1})\}$; the orthogonal complement cancels from the determinant ratio. To see why this bound remains valid for adaptive queries, the standard proof starts from the conditional sub-Gaussian assumption. For a feature-space vector h, it constructs the exponential-supermartingale $\exp(\langle h, S_t\rangle/R - \|h\|_{V_t - \lambda I}^2/2)$: a nonnegative process whose conditional expectation cannot increase. Integrating over h against a centered Gaussian with precision λI combines these directional bounds. Ville's inequality then controls the probability that the resulting process ever crosses a threshold, which gives a bound uniform in time. Gaussian integration produces the determinant ratio in (A.2); the same event holds for every stopping time and hence for all t.

The determinant ratio is written in feature space, whereas γ_{t-1} is defined using the Gram matrix at observed locations. The matrix determinant lemma and the primal–dual determinant identity connect these two forms:

$$\frac{1}{2}\log\frac{\det(V_{t-1})}{\det(\lambda I)} = \frac{1}{2}\log\det(I + \lambda^{-1}K_{t-1}) \le \gamma_{t-1}.$$

Insert this inequality and $\|f\|_{\mathcal{H}_k} \le B$ into (A.1)–(A.2), and use $s_{t-1}(a) = \sqrt{\lambda}\|\varphi(a)\|_{V_{t-1}^{-1}}$. The resulting coefficient is exactly b_t, proving the claim. □

Confidence bounds and Gibbs regret. We clip the confidence intervals to the known reward range so that both the policies and their errors remain bounded, even when the confidence event fails. Let $\Pi_{[0,C]}(z) = \min\{C, \max\{0, z\}\}$. For every history, define

$$L_t(a) = \Pi_{[0,C]}(m_{t-1}(a) - b_t s_{t-1}(a)), \qquad U_t(a) = \Pi_{[0,C]}(m_{t-1}(a) + b_t s_{t-1}(a)).$$

Monotonicity of the projection ensures $0 \le L_t(a) \le U_t(a) \le C$ for every history, including outside the confidence event.

For any measurable $g : \mathcal{A} \to [0, C]$, let

$$S_\eta(g)(da) = \frac{\exp(\eta g(a))\mu(da)}{\int_{\mathcal{A}} \exp(\eta g(a'))\mu(da')}.$$

A sampling function g_t is the score used to construct the query distribution at round t. For a sequence of such functions chosen from the history, the reverse-KL sampling regret is

$$\mathrm{SRegret}(T) = \frac{1}{\eta}\sum_{t=1}^{T} D_{\mathrm{KL}}(S_\eta(g_t) \,\|\, S_\eta(f)).$$

This accumulates the discrepancy between each query distribution and the true target. Reversing the arguments of KL gives the forward-KL counterpart,

$$\text{FKL-SRegret}(T) = \frac{1}{\eta}\sum_{t=1}^{T} D_{\mathrm{KL}}(S_\eta(f) \,\|\, S_\eta(g_t))\,.$$

We add a subscript PM, UCS, or TS^2 when g_t is chosen by the corresponding algorithm below. For finite $\mathcal{A}$, a draw from $S_\eta(g)$ is a categorical draw with probabilities proportional to $\exp(\eta g(a))\mu(\{a\})$. Unless an expectation is explicitly conditional, it includes the observation noise and all randomization used by the relevant algorithm. In the Bayesian results it also includes the prior draw of f. The posterior-mean and GP-UCS bounds use the fixed-function assumptions above; the GP-TS^2 and one-step results state their separate Bayesian assumptions.

A.1 Posterior-mean GP sampling

Posterior-mean sampling uses the current reward estimate directly, without an uncertainty bonus. It still explores because a Gibbs distribution at finite temperature assigns probability throughout the support of μ. The bound below quantifies this implicit exploration and its dependence on temperature.

Definition 1 (Posterior-mean GP sampling)**.** *Fix an inverse temperature $\eta > 0$. At round t, compute m_{t-1} and s_{t-1}, set*

$$g_t^{\mathrm{PM}}(a) = \Pi_{[0,C]}(m_{t-1}(a)),$$

sample $A_t \sim S_\eta(g_t^{\mathrm{PM}})$, observe Y_t, and update the GP posterior. The projection may be omitted when m_{t-1} is known to take values in $[0, C]$.

The proof uses the Sampling With Confidence (SWC) property: the score defining the query distribution and the true reward lie in the same confidence interval. This lets us control their induced KL divergence through the interval's width.

Proposition 1 (Sampling With Confidence property)**.** *On the uniform confidence event, for every t and a,*

$$L_t(a) \le f(a), g_t^{\mathrm{PM}}(a) \le U_t(a).$$

Thus posterior-mean GP sampling satisfies the SWC property.

Proof. The confidence theorem and $f(a) \in [0, C]$ imply $L_t(a) \le f(a) \le U_t(a)$, so the clipped interval is nonempty. Projection onto an interval is monotone. Projecting $m_{t-1}(a) - b_t s_{t-1}(a) \le m_{t-1}(a) \le m_{t-1}(a) + b_t s_{t-1}(a)$ onto $[0, C]$ therefore gives $L_t(a) \le g_t^{\mathrm{PM}}(a) \le U_t(a)$. □

Theorem 4 (Regret of posterior-mean GP sampling)**.** *Under the assumptions of the uniform confidence theorem,*

$$\mathbb{E}[\text{SRegret}_{\mathrm{PM}}(T)] \le \frac{4\eta e^{\eta C} b_T^2}{\log(1+\lambda^{-1})}\gamma_T + 2CT\delta.$$

With $\delta = T^{-2}$ and fixed η, C, and λ,

$$\mathbb{E}[\text{SRegret}_{\mathrm{PM}}(T)] = \widetilde{O}\big(\eta e^{\eta C} b_T^2 \gamma_T\big)\,.$$

The forward-KL regret satisfies the same statements with $e^{\eta C}$ replaced by $e^{3\eta C}$.

The first term accounts for uncertainty while the confidence intervals are valid. The additive failure term accounts for the remaining event, on which clipping still bounds the regret. Choosing a decreasing failure probability keeps that contribution from accumulating linearly with the query budget.

Proof. Let $\mathcal{E}$ be the uniform confidence event. We first bound each round's regret by the squared confidence width at the sampled location. We then sum these widths using the information-gain bound. The first step gives

$$\mathbb{E}[\text{SRegret}_{\mathrm{PM}}(T)] \le \frac{\eta e^{\eta C}}{2}\mathbb{E}\left[\sum_{t=1}^{T}(U_t(A_t) - L_t(A_t))^2\right] + 2CT\mathbb{P}(\mathcal{E}^c). \tag{A.3}$$

Fix t and work on $\mathcal{E}$. Let $\pi_t = S_\eta(g_t^{\mathrm{PM}})$, and put $h = g_t^{\mathrm{PM}} - f$. The Gibbs density ratio gives the exact identity

$$D_{\mathrm{KL}}(\pi_t \| S_\eta(f)) = \eta \mathbb{E}_{\pi_t}[h] + \log \mathbb{E}_{\pi_t}[e^{-\eta h}]. \tag{A.4}$$

Because $g_t^{\mathrm{PM}}, f \in [0, C]$, we have $\eta h \geq -\eta C$. Taylor's theorem implies $e^{-x} \leq 1 - x + e^{\eta C} x^2/2$ for every $x \geq -\eta C$. Apply this to the exponential in (A.4), then use $\log u \leq u - 1$. The two terms linear in h cancel, leaving a bound quadratic in the score error. Since both scores lie in the same confidence interval, this yields

$$\eta^{-1} D_{\mathrm{KL}}(S_\eta(g_t^{\mathrm{PM}}) \| S_\eta(f)) \leq \frac{\eta e^{\eta C}}{2} \mathbb{E}_{A_t \sim S_\eta(g_t^{\mathrm{PM}})}[(U_t(A_t) - L_t(A_t))^2]$$

because $|h| \leq U_t - L_t$ on $\mathcal{E}$. Outside $\mathcal{E}$, both f and g_t^{PM} still lie in $[0, C]$. Each of the two terms in the scaled KL identity—the reward difference and the difference of log normalizing constants—is then at most C in absolute value. The instantaneous regret is therefore at most $2C$. Summing over rounds and averaging over the success and failure events proves (A.3), with $\mathbb{P}(\mathcal{E}^c) \leq \delta$ supplied by the confidence theorem.

We next relate the confidence widths to the GP variances. Clipping cannot enlarge an interval, and the confidence radii are nondecreasing, hence

$$(U_t(A_t) - L_t(A_t))^2 \leq 4 b_t^2 s_{t-1}^2(A_t) \leq 4 b_T^2 s_{t-1}^2(A_t). \tag{A.5}$$

It remains to bound the sum of posterior variances at the locations actually queried. The following identity expresses their total information gain as a sum of per-query contributions. Repeated use of the matrix determinant lemma gives

$$\frac{1}{2} \log \det(I + \lambda^{-1} K_T) = \frac{1}{2} \sum_{t=1}^{T} \log(1 + \lambda^{-1} s_{t-1}^2(A_t)). \tag{A.6}$$

Since $0 \leq s_{t-1}^2(A_t) \leq k(A_t, A_t) \leq 1$, concavity of $x \mapsto \log(1 + x/\lambda)$ implies

$$\log(1 + \lambda^{-1} x) \geq x \log(1 + \lambda^{-1}) \quad (0 \leq x \leq 1).$$

Combining this inequality with (A.6) and the definition of γ_T yields

$$\sum_{t=1}^{T} s_{t-1}^2(A_t) \leq \frac{2\gamma_T}{\log(1 + \lambda^{-1})}. \tag{A.7}$$

Substituting (A.5)–(A.7) into (A.3) proves the displayed reverse-KL bound and its stated order. This step is valid for any realized query sequence: no independence or fixed design is required by the determinant identity.

For forward KL, the same Taylor argument applies after exchanging $S_\eta(f)$ and $S_\eta(g_t^{\mathrm{PM}})$. It again gives a factor $e^{\eta C}$, but now averages the squared width under $S_\eta(f)$. To reuse the variance bound at the algorithm's query locations, we must change this expectation back to the query distribution. The Gibbs density-ratio identity and $|f - g_t^{\mathrm{PM}}| \leq C$ imply

$$\frac{dS_\eta(f)}{dS_\eta(g_t^{\mathrm{PM}})}(a) \leq e^{2\eta C}.$$

Changing the squared-width expectation from $S_\eta(f)$ to $S_\eta(g_t^{\mathrm{PM}})$ therefore multiplies the bound by at most $e^{2\eta C}$. The product is $e^{3\eta C}$, and the remainder of the proof is unchanged. The width and variance bounds (A.5)–(A.7) are pathwise: they hold for each realized sequence of queries. They do not depend on the posterior-mean sampling rule and will also be used for GP-UCS and GP-TS[2]. □

A.2 GP-UCS

GP-UCS adds an uncertainty bonus to the reward estimate. On the confidence event, the resulting score is an upper bound on the true reward at every location. This one-sided error is what removes the exponential temperature factor from the cumulative-regret bound.

Definition 2 (Gaussian-process upper-confidence sampling)**.** *Fix an inverse temperature $\eta > 0$. At round t, compute m_{t-1} and s_{t-1} and set*

$$g_t^{\mathrm{UCS}}(a) = U_t(a) = \Pi_{[0,C]}(m_{t-1}(a) + b_t s_{t-1}(a)).$$

Sample $A_t \sim S_\eta(g_t^{\mathrm{UCS}})$, observe Y_t, and update the GP posterior. We call this procedure GP-UCS.

Proposition 2 (Optimism of GP-UCS)**.** *On the uniform confidence event,*

$$0 \le f(a) \le g_t^{\mathrm{UCS}}(a) \le C \quad \textit{and} \quad 0 \le g_t^{\mathrm{UCS}}(a) - f(a) \le U_t(a) - L_t(a)$$

for every t and a.

Proof. The uniform confidence theorem gives $m_{t-1}(a) - b_t s_{t-1}(a) \le f(a) \le m_{t-1}(a) + b_t s_{t-1}(a)$. Intersecting this interval with $[0, C]$ gives $L_t(a) \le f(a) \le U_t(a)$. Since $g_t^{\mathrm{UCS}} = U_t$, both displayed conclusions follow. □

Theorem 5 (Regret of GP-UCS)**.** *Under the assumptions of the uniform confidence theorem, GP-UCS satisfies*

$$\mathbb{E}[\mathrm{SRegret}_{\mathrm{UCS}}(T)] \le \min\left\{2b_T\sqrt{\frac{2T\gamma_T}{\log(1+\lambda^{-1})}}, \frac{4\eta b_T^2\gamma_T}{\log(1+\lambda^{-1})}\right\} + 2CT\delta. \tag{A.8}$$

Equivalently,

$$\mathbb{E}[\mathrm{SRegret}_{\mathrm{UCS}}(T)] = \widetilde{O}\Big(\min\{b_T\sqrt{T\gamma_T}, \eta b_T^2\gamma_T\}\Big) + 2CT\delta.$$

In particular, unlike the general SWC guarantee for posterior-mean sampling, the GP-UCS bound has no factor exponential in ηC.

Proof. Let $\mathcal{E}$ be the uniform confidence event and fix a round t. On $\mathcal{E}$, put $h = g_t^{\mathrm{UCS}} - f$. By the optimism proposition, $0 \le h \le U_t - L_t$. Applying the Gibbs KL identity (A.4) with $\pi_t = S_\eta(g_t^{\mathrm{UCS}})$ gives

$$\eta^{-1} D_{\mathrm{KL}}(S_\eta(g_t^{\mathrm{UCS}}) \| S_\eta(f)) = \mathbb{E}_{\pi_t}[h] + \eta^{-1}\log \mathbb{E}_{\pi_t}[e^{-\eta h}]. \tag{A.9}$$

There are two useful ways to bound this expression. Since $h \ge 0$, dropping the nonpositive logarithmic term gives a bound linear in the confidence width. Alternatively, $e^{-x} \le 1 - x + x^2/2$ for $x \ge 0$ and $\log u \le u - 1$ give a quadratic bound. Together they yield

$$\begin{aligned}
\eta^{-1} D_{\mathrm{KL}}(S_\eta(g_t^{\mathrm{UCS}}) \| S_\eta(f)) &\le \mathbb{E}_{\pi_t}[U_t - L_t], \\
\eta^{-1} D_{\mathrm{KL}}(S_\eta(g_t^{\mathrm{UCS}}) \| S_\eta(f)) &\le \frac{\eta}{2}\mathbb{E}_{\pi_t}[(U_t - L_t)^2].
\end{aligned} \tag{A.10}$$

On $\mathcal{E}^c$, the instantaneous regret is at most $2C$, by the same bounded Gibbs KL calculation used in the posterior-mean proof. Therefore the two inequalities in (A.10), summed over time and averaged, give

$$\begin{aligned}
\mathbb{E}[\mathrm{SRegret}_{\mathrm{UCS}}(T)] &\le \mathbb{E}\left[\sum_{t=1}^{T}(U_t(A_t) - L_t(A_t))\right] + 2CT\delta, \\
\mathbb{E}[\mathrm{SRegret}_{\mathrm{UCS}}(T)] &\le \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}(U_t(A_t) - L_t(A_t))^2\right] + 2CT\delta.
\end{aligned} \tag{A.11}$$

For the first line, Cauchy–Schwarz, the width bound in (A.5), and the variance bound (A.7) yield

$$\sum_{t=1}^{T}(U_t(A_t) - L_t(A_t)) \le 2b_T\sqrt{T\sum_{t=1}^{T} s_{t-1}^2(A_t)} \le 2b_T\sqrt{\frac{2T\gamma_T}{\log(1+\lambda^{-1})}}.$$

For the second line of (A.11), equations (A.5) and (A.7) yield

$$\frac{\eta}{2}\sum_{t=1}^{T}(U_t(A_t) - L_t(A_t))^2 \le \frac{4\eta b_T^2\gamma_T}{\log(1+\lambda^{-1})}.$$

The linear-width argument produces the square-root dependence on T; the squared-width argument instead retains a factor of η. Both hold for the same policy and query sequence, so taking their minimum proves (A.8) and its stated asymptotic form. □

A.3 GP-TS^2

GP-TS^2 explores by drawing a possible reward function and sampling from its Gibbs distribution. Its analysis uses a well-specified Bayesian model: the prior and likelihood used by the algorithm also generate the true function and observations. If ρ_t is the posterior law of f given H_t, the true function and an independent posterior draw have the same conditional distribution. This identity, often called probability matching, is the key to the regret proof.

We additionally assume that ρ_t is supported on functions taking values in $[0, C]$. This bounds the KL error when the confidence event fails. Such a model may use a bounded prior or apply the same bounded link to the latent true function and posterior draws. The confidence event in the theorem must still hold for that model; we use the GP quantities above to form its confidence intervals. Ordinary, unclipped Gaussian posterior paths do not satisfy this boundedness assumption. The theorem therefore concerns the stated bounded model, with exact posterior sampling, rather than providing a guarantee for those Gaussian paths. The Gaussian model used for the one-step calculation below has different assumptions.

Definition 3 (GP-TS^2)**.** *At round t, draw a function $\widetilde{f}_t \sim \rho_t$, set $g_t^{\mathrm{TS}^2} = \widetilde{f}_t$, and then draw*

$$A_t \sim S_\eta(g_t^{\mathrm{TS}^2}).$$

After observing Y_t, update the GP posterior. Thus GP-TS^2 contains two sampling operations: it first samples a reward function from the GP posterior and then samples an action from the Gibbs distribution induced by that function.

Define the reverse-KL Bayesian sampling regret by

$$\mathrm{BSRegret}_{\mathrm{TS}^2}(T) = \frac{1}{\eta}\mathbb{E}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\Big(S_\eta(g_t^{\mathrm{TS}^2}) \,\Big\|\, S_\eta(f)\Big)\right],$$

where the expectation includes the prior draw of f, the observation noise, the posterior function draws, and the action draws. The forward-KL Bayesian sampling regret is defined by reversing the two arguments of D_{KL} in this display.

Theorem 6 (Bayesian regret of GP-TS^2)**.** *Suppose that the assumptions above hold and that the simultaneous confidence event*

$$\mathcal{E} = \{L_t(a) \le f(a) \le U_t(a) \text{ for every } t \le T, a \in \mathcal{A}\}$$

has probability at least $1 - \delta$. Then

$$\mathrm{BSRegret}_{\mathrm{TS}^2}(T) \le \min\left\{2b_T\sqrt{\frac{2T\gamma_T}{\log(1+\lambda^{-1})}}, \frac{8\eta b_T^2\gamma_T}{\log(1+\lambda^{-1})}\right\} + 4CT\delta. \tag{A.12}$$

The identical bound holds for forward-KL Bayesian sampling regret. In asymptotic notation,

$$\mathrm{BSRegret}_{\mathrm{TS}^2}(T) = \widetilde{O}\Big(\min\{b_T\sqrt{T\gamma_T}, \eta b_T^2\gamma_T\}\Big) + 4CT\delta.$$

Proof. Fix t and condition on H_t. We first analyze two posterior functions that both lie in the confidence interval, then account for the probability that either falls outside it. Let F_1, F_2 be independent draws from ρ_t conditioned to lie in the confidence set $\{q : L_t \le q \le U_t\}$. The pair is exchangeable: swapping F_1 and F_2 does not change its joint distribution. We will use this symmetry to obtain the following linear- and squared-width bounds:

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}D_{\mathrm{KL}}(S_\eta(F_1)\|S_\eta(F_2)) \le \mathbb{E}_{F_1, A\sim S_\eta(F_1)}[U_t(A) - L_t(A)],\\
&\frac{1}{\eta}\mathbb{E}D_{\mathrm{KL}}(S_\eta(F_1)\|S_\eta(F_2)) \le \eta\mathbb{E}_{F_1, A\sim S_\eta(F_1)}[(U_t(A) - L_t(A))^2].
\end{aligned} \tag{A.13}$$

For the first inequality, expand the KL divergence as in (A.4). The difference of log normalizing constants has expectation zero because F_1 and F_2 have the same distribution. What remains is the reward difference at an action drawn using F_1. Both functions lie inside the same interval, so that difference is at most $U_t(A) - L_t(A)$.

For the second inequality, it is useful to add the two directions of KL before averaging over the functions. Fix q_1, q_2 in the confidence set, write $\pi_i = S_\eta(q_i)$, $r = q_2 - q_1$, and $c = \eta^{-1} \log \mathbb{E}_{\pi_1}[e^{\eta r}]$. The constant c accounts for the change in normalization between the two Gibbs distributions. Their density ratio is $d\pi_2 = e^u d\pi_1$, where $u = \eta(r - c)$, so adding the two KL expressions gives

$$D_{\mathrm{KL}}(\pi_1 \| \pi_2) + D_{\mathrm{KL}}(\pi_2 \| \pi_1) = \int (e^u - 1) u \, d\pi_1.$$

Apply the pointwise inequality $(e^u - 1)u \le (e^u + 1)u^2/2$ to bound this by second moments of $r - c$ under the two distributions. To remove c, consider the log moment-generating function of r under π_1. Convexity places its secant slope c, over the interval from zero to η, between its endpoint derivatives: the means of r under π_1 and π_2. Expanding the two centered squares and using this ordering bounds them by twice the sum of the uncentered second moments. Consequently,

$$D_{\mathrm{KL}}(\pi_1 \| \pi_2) + D_{\mathrm{KL}}(\pi_2 \| \pi_1) \le \eta^2 \left(\int r^2 d\pi_1 + \int r^2 d\pi_2 \right).$$

Now use $|r| \le U_t - L_t$ and average over the exchangeable pair (F_1, F_2). Swapping the two functions exchanges the two KL terms, so their expectations agree. The two squared-width terms likewise have equal expectations. Dividing by two proves the second inequality in (A.13). The same symmetry proves both bounds with the KL direction reversed.

Conditionally on H_t, f and $g_t^{\mathrm{TS}^2}$ are independent and have the same posterior law. Hence (A.13) applies whenever both functions lie in the confidence set. If $\mathcal{E}_t$ denotes the event that f lies in the round-t confidence set, probability matching and a union bound give

$$\mathbb{P}\big(f \text{ or } g_t^{\mathrm{TS}^2} \text{ is outside the confidence set}\big) \le 2\mathbb{P}(\mathcal{E}_t^c) \le 2\delta.$$

These probabilities are averaged over histories; they need not be bounded by 2δ separately for every history. On the failure event, bounded rewards give instantaneous scaled KL regret at most $2C$. On the event that both functions are inside the interval, the conditional bounds in (A.13) apply. Weighting those bounds by the probability of this event only reduces the nonnegative width terms, so they are bounded by the corresponding unconditional expectations at the algorithm's queries. Summing the success and failure contributions over t gives

$$\begin{aligned} \mathrm{BSRegret}_{\mathrm{TS}^2}(T) &\le \mathbb{E}\left[\sum_{t=1}^{T} (U_t(A_t) - L_t(A_t)) \right] + 4CT\delta, \\ \mathrm{BSRegret}_{\mathrm{TS}^2}(T) &\le \eta \mathbb{E}\left[\sum_{t=1}^{T} (U_t(A_t) - L_t(A_t))^2 \right] + 4CT\delta. \end{aligned} \tag{A.14}$$

At this point the argument has the same form as for GP-UCS. The width calculations (A.5)–(A.7) apply to the realized GP-TS^2 actions as well. They bound the first term in (A.14) by

$$2b_T \sqrt{\frac{2T\gamma_T}{\log(1 + \lambda^{-1})}}$$

and the second by

$$\frac{8\eta b_T^2 \gamma_T}{\log(1 + \lambda^{-1})}.$$

Taking the smaller bound proves (A.12). Exchangeability in (A.13) proves the same result for forward KL. □

A.4 Integrated posterior contraction and simple regret

Cumulative regret measures the quality of the queries made during learning. We now ask how accurate the policy is after all T observations. The first step is to show that uncertainty decreases across the state space, not only at previously queried locations. A bounded Gibbs score ensures that every region with positive μ-mass retains a positive chance of being sampled. We use this to control the posterior variance averaged under μ,

$$V_t = \int_{\mathcal{A}} s_t^2(a) \, \mu(da).$$

Here V_t is a scalar measure of remaining uncertainty, distinct from the feature-space operator used in the confidence proof. "Contraction" refers to this average variance decreasing as observations accumulate.

Theorem 7 (Integrated posterior contraction)**.** *Suppose every sampling function used by the algorithm takes values in $[0, C]$. This holds for posterior-mean GP sampling and GP-UCS by construction, and for GP-TS^2 under its bounded-prior assumption. Then*

$$\mathbb{E}[V_T] \le \frac{2e^{\eta C}\gamma_T}{T\log(1+\lambda^{-1})}. \tag{A.15}$$

If, after observing the first T rewards, the algorithm constructs its next sampling function g_{T+1} and draws $A_{T+1} \sim S_\eta(g_{T+1})$, then

$$\mathbb{E}[s_T^2(A_{T+1})] \le \frac{2e^{2\eta C}\gamma_T}{T\log(1+\lambda^{-1})}. \tag{A.16}$$

Consequently, if $\gamma_T = o(T)$, both the integrated posterior variance and the posterior variance encountered by the final policy converge to zero in expectation.

Proof. For any measurable $g : \mathcal{A} \to [0, C]$, its Gibbs density relative to μ satisfies

$$e^{-\eta C} \le \frac{dS_\eta(g)}{d\mu}(a) = \frac{e^{\eta g(a)}}{\int e^{\eta g(a')}\mu(da')} \le e^{\eta C}. \tag{A.17}$$

The numerator and normalizing constant both lie in $[1, e^{\eta C}]$. Thus the query distribution cannot underweight any region by more than the factor $e^{-\eta C}$ relative to μ. In particular, uncertainty averaged at query locations controls uncertainty averaged across the domain. Applying the lower bound in (A.17), conditional on both the history and the possibly random sampling function, gives

$$\mathbb{E}[s_{t-1}^2(A_t) \mid H_t, g_t] = \int s_{t-1}^2(a) S_\eta(g_t)(da) \ge e^{-\eta C} V_{t-1}. \tag{A.18}$$

The GP covariance update is

$$s_t^2(a) = s_{t-1}^2(a) - \frac{\mathrm{cov}_{t-1}(a, A_t)^2}{\lambda + s_{t-1}^2(A_t)},$$

where $\mathrm{cov}_{t-1}(a, A_t)$ is the posterior covariance between the prediction at a and the reward at the queried location. The subtracted term is nonnegative, so $s_t^2(a) \le s_{t-1}^2(a)$ for every a, and hence $V_t \le V_{t-1}$. Averaging (A.18) over the history and score, summing over rounds, and applying the information-gain bound (A.7) yields

$$e^{-\eta C}\sum_{t=1}^{T}\mathbb{E}[V_{t-1}] \le \mathbb{E}\left[\sum_{t=1}^{T} s_{t-1}^2(A_t)\right] \le \frac{2\gamma_T}{\log(1+\lambda^{-1})}. \tag{A.19}$$

Because the average variance decreases after every observation, its final value is no larger than its average over earlier rounds. Thus $T\mathbb{E}[V_T] \le \sum_{t=1}^{T}\mathbb{E}[V_{t-1}]$. This converts the cumulative bound in (A.19) into the final-variance bound (A.15).

To obtain the variance seen by the final policy, change from the base measure back to that policy. This uses the upper density bound in (A.17) and contributes one more factor of $e^{\eta C}$: conditionally after the first T observations,

$$\mathbb{E}[s_T^2(A_{T+1}) \mid H_{T+1}, g_{T+1}] \le e^{\eta C} V_T.$$

Taking expectations and applying (A.15) proves (A.16). The convergence claim follows when $\gamma_T/T \to 0$. □

The preceding theorem is a statement about the surrogate's uncertainty. Combining it with the per-round KL bounds gives a statement about the final sampler. Its *simple sampling regret* measures only the quality of this final policy, without summing over earlier queries:

$$r_{T+1}(g_{T+1}) = \frac{1}{\eta} D_{\mathrm{KL}}(S_\eta(g_{T+1}) \,\|\, S_\eta(f)).$$

Corollary 1 (Final-policy simple regret)**.** *Assume the conditions of the integrated posterior contraction theorem and that the simultaneous confidence event also holds at time $T+1$ with probability at least $1-\delta$. Posterior-mean GP sampling satisfies*

$$\mathbb{E}[r_{T+1}^{\mathrm{PM}}] \le \frac{4\eta e^{3\eta C} b_{T+1}^2 \gamma_T}{T\log(1+\lambda^{-1})} + 2C\delta. \tag{A.20}$$

GP-UCS satisfies

$$\mathbb{E}[r_{T+1}^{\mathrm{UCS}}] \le \min\left\{2e^{\eta C} b_{T+1}\sqrt{\frac{2\gamma_T}{T\log(1+\lambda^{-1})}}, \frac{4\eta e^{2\eta C} b_{T+1}^2 \gamma_T}{T\log(1+\lambda^{-1})}\right\} + 2C\delta. \tag{A.21}$$

Under the Bayesian assumptions of GP-TS^2,

$$\mathbb{E}[r_{T+1}^{\mathrm{TS}^2}] \le \min\left\{2e^{\eta C} b_{T+1}\sqrt{\frac{2\gamma_T}{T\log(1+\lambda^{-1})}}, \frac{8\eta e^{2\eta C} b_{T+1}^2 \gamma_T}{T\log(1+\lambda^{-1})}\right\} + 4C\delta. \tag{A.22}$$

The GP-TS^2 statement also holds for forward-KL simple regret. Posterior-mean sampling satisfies the forward-KL bound

$$\mathbb{E}[r_{T+1}^{\mathrm{PM,FKL}}] \le \frac{4\eta e^{5\eta C} b_{T+1}^2 \gamma_T}{T\log(1+\lambda^{-1})} + 2C\delta. \tag{A.23}$$

These bounds concern the next query policy generated by each algorithm: the posterior-mean policy for GP-PM, the optimistic policy for GP-UCS, and a fresh posterior-function policy for GP-TS^2. No additional observation at A_{T+1} is needed to define or use that final policy. The exponential constants arise from comparing each query distribution with μ, even for the algorithms whose cumulative-regret bounds avoid such constants.

Proof. Each claim follows by applying the corresponding per-round regret bound to the final policy and substituting the posterior-variance bound. For posterior-mean sampling, the SWC argument gives

$$\mathbb{E}[r_{T+1}^{\mathrm{PM}}] \le 2\eta e^{\eta C} b_{T+1}^2 \mathbb{E}[s_T^2(A_{T+1})] + 2C\delta.$$

Substitution of (A.16) proves (A.20). The two one-step GP-UCS bounds give

$$\mathbb{E}[r_{T+1}^{\mathrm{UCS}}] \le \min\left\{2b_{T+1}\sqrt{\mathbb{E}[s_T^2(A_{T+1})]}, 2\eta b_{T+1}^2 \mathbb{E}[s_T^2(A_{T+1})]\right\} + 2C\delta,$$

where Jensen's inequality bounds the expected standard deviation by the square root of the expected variance in the first term. Substituting (A.16) in each branch proves (A.21).

For GP-TS^2, the per-round probability-matching bounds have the same linear and quadratic form, with coefficients $2b_{T+1}$ and $4\eta b_{T+1}^2$ and failure term $4C\delta$. Substitution of (A.16) therefore proves (A.22). Finally, use the forward-KL per-round bounds from the corresponding algorithm proofs: the GP-TS^2 bound is unchanged by exchangeability, while the posterior-mean bound carries the additional density-ratio factor. □

A.5 One-step lookahead

The one-step objective values a query by the expected quality of the sampler after observing its result. We first show why the posterior-mean Gibbs measure is the best current policy under expected reverse KL. We then use the GP update to calculate the expected improvement from a candidate query, proving Theorem 2. The remaining results explain the three approximations used in the experiments and their relation to existing acquisition functions.

Here we assume a well-specified Gaussian regression model: f is drawn from the GP prior, the observation noise is $\varepsilon_t \sim \mathcal{N}(0,\lambda)$, and expectations are taken under the resulting posterior. This differs from both the fixed-function assumptions for GP-PM and GP-UCS and the bounded posterior required by the GP-TS^2 theorem. We assume that $\mathbb{E}\,|\log\int e^{\eta f}\,\mathrm{d}\mu| < \infty$, which holds for a continuous kernel on a compact $\mathcal{A}$. This integrability condition allows us to average the log normalizing constant of the unknown target. The posterior mean is not clipped, so the current policy

is $\pi_t = S_\eta(m_{t-1})$. In the experiments, the GP is fitted to clipped energies, but its mean is used without further clipping.

We return to the notation x, x' for query locations, reserving a_t for the acquisition. Write Σ_{t-1} for the covariance given H_t, with $s^2_{t-1}(x) = \Sigma_{t-1}(x, x)$. Before evaluating x, its observation has the predictive distribution $Y = m_{t-1}(x) + (\Sigma_{t-1}(x, x) + \lambda)^{1/2} Z$, where $Z \sim \mathcal{N}(0, 1)$. Conditioning on that observation gives

$$m_t = m_{t-1} + Z\, r_x, \qquad \Sigma_t = \Sigma_{t-1} - r_x \otimes r_x, \qquad r_x(x') = \frac{\Sigma_{t-1}(x', x)}{\sqrt{\Sigma_{t-1}(x, x) + \lambda}}.$$

The function r_x describes how information at x changes predictions at other locations. The random scalar Z is the observation's standardized prediction error and determines the direction and size of the mean update. In the covariance expression, $r_x \otimes r_x$ is the function of two locations whose value is the product of their two r_x values. The covariance reduction depends only on the query location, so it can be computed before the observation is available.

Free energy and the best current policy. For a function g with $\int e^{\eta g}\, \mathrm{d}\mu < \infty$, define the soft maximum

$$A_\eta(g) = \frac{1}{\eta} \log \int e^{\eta g}\, \mathrm{d}\mu. \tag{A.24}$$

This is the log normalizing constant of the Gibbs distribution, scaled by $1/\eta$. It is convex, and its derivative at g in a direction u is $\mathbb{E}_{S_\eta(g)}[u]$. Expanding the KL divergence gives the identity below for any policy π that has a density relative to μ, written $\pi \ll \mu$:

$$\frac{1}{\eta} D_{\mathrm{KL}}\big(\pi \,\|\, S_\eta(g)\big) = A_\eta(g) - \mathbb{E}_\pi[g] + \frac{1}{\eta} D_{\mathrm{KL}}(\pi \,\|\, \mu), \tag{A.25}$$

and, since KL is nonnegative, $\mathbb{E}_\pi[g] - \eta^{-1} D_{\mathrm{KL}}(\pi\|\mu) \le A_\eta(g)$, with equality if and only if $\pi = S_\eta(g)$. This is the Gibbs variational principle: the Gibbs policy maximizes expected reward with a KL penalty for departing from the base measure. Taking $\pi = S_\eta(g')$ gives

$$\frac{1}{\eta} D_{\mathrm{KL}}\big(S_\eta(g') \,\|\, S_\eta(g)\big) = A_\eta(g) - A_\eta(g') - \mathbb{E}_{S_\eta(g')}[g - g'], \tag{A.26}$$

whose right-hand side is a Bregman divergence: the difference between a convex function and its linear approximation at g'. This form makes it possible to average over the unknown reward function without computing its Gibbs distribution for every posterior draw.

Lemma 1 (Bayes-optimal policy)**.** *Let π be a probability measure chosen as a function of H_t. Then*

$$\begin{aligned}\mathbb{E}\Big[\tfrac{1}{\eta} D_{\mathrm{KL}}\big(\pi \,\|\, S_\eta(f)\big) \,\Big|\, H_t\Big] &= \mathbb{E}\big[A_\eta(f) \mid H_t\big] - \mathbb{E}_\pi[m_{t-1}] + \tfrac{1}{\eta} D_{\mathrm{KL}}(\pi \,\|\, \mu) \\ &\ge \mathbb{E}\big[A_\eta(f) \mid H_t\big] - A_\eta(m_{t-1}),\end{aligned}$$

with equality if and only if $\pi = \pi_t$. The posterior-mean Gibbs measure therefore minimizes the posterior expected regret of the next policy, and $J_t^0 = \mathbb{E}[A_\eta(f) \mid H_t] - A_\eta(m_{t-1})$.

Proof. Apply (A.25) with $g = f$ and average over the posterior. Given H_t, the proposed policy π is fixed, while the posterior mean of the unknown function is m_{t-1}. We can therefore interchange the expectation over the policy's state and the expectation over the function: Fubini's theorem gives $\mathbb{E}[\mathbb{E}_\pi f \mid H_t] = \mathbb{E}_\pi[m_{t-1}]$. This proves the first equality. The only policy-dependent terms that remain are expected posterior-mean reward and the KL penalty relative to μ. Applying the Gibbs variational principle with $g = m_{t-1}$ gives the inequality and its equality case. □

The lemma says that, if no further observations will be made, the posterior-mean Gibbs policy minimizes expected reverse KL to the unknown target. The term $\mathbb{E}[A_\eta(f) \mid H_t]$ is independent of the chosen policy, so only the posterior mean is needed to find this optimum. This is an instance of the general result that the posterior mean is a Bayes estimate under a Bregman loss (Banerjee et al., 2005). It does not imply that posterior-mean sampling chooses the most informative next query.

Proof of Theorem 2. For a candidate query x, we apply Lemma 1 to the updated policy $S_\eta(m_t)$ and average over the possible observation Y. The posterior expected error after observing Y has the form given by that lemma. Averaging it over Y returns the current expectation of $A_\eta(f)$, because observing and then averaging does not change our current belief about this quantity. The law of total expectation gives

$$J_t(x) = \mathbb{E}\Big[\mathbb{E}\big[A_\eta(f) \mid H_t, Y\big] - A_\eta(m_t) \,\Big|\, H_t\Big] = \mathbb{E}\big[A_\eta(f) \mid H_t\big] - \mathbb{E}_Z\big[A_\eta(m_{t-1} + Z r_x)\big],$$

where the first term is independent of x. The reduction in expected regret is therefore the expected increase in the soft maximum of the posterior mean: $J_t(x) = J_t^0 - a_t(x)$ with

$$a_t(x) = \mathbb{E}_Z\big[A_\eta(m_{t-1} + Z r_x)\big] - A_\eta(m_{t-1}). \tag{A.27}$$

To express this value as an expectation under the current policy, factor out its normalizing constant and center the update using $h_x = r_x - \mathbb{E}_{\pi_t}[r_x]$. Since $\int e^{\eta(m_{t-1}+Z r_x)}\,\mathrm{d}\mu = \int e^{\eta m_{t-1}}\,\mathrm{d}\mu\,\mathbb{E}_{\pi_t}[e^{\eta Z r_x}]$,

$$A_\eta(m_{t-1} + Z r_x) - A_\eta(m_{t-1}) = \frac{1}{\eta}\log \mathbb{E}_{\pi_t}\big[e^{\eta Z r_x}\big] = Z\,\mathbb{E}_{\pi_t}[r_x] + \frac{1}{\eta}\log \mathbb{E}_{\pi_t}\big[e^{\eta Z h_x}\big],$$

and the first term has mean zero under Z, which gives (6). This also makes nonnegativity transparent. For each fixed Z, Jensen's inequality gives $\log \mathbb{E}_{\pi_t}[e^{\eta Z h_x}] \geq \eta Z\,\mathbb{E}_{\pi_t}[h_x] = 0$, since h_x is centered. Equality holds for every Z if and only if $h_x = 0$ π_t-almost surely: the possible update then changes all relevant rewards by the same constant. □

The next bounds connect the size of this improvement to the variation in r_x and to the remaining posterior uncertainty. All suprema in the statement concern the support of the current policy; $\|h_x\|_\infty$ is the largest absolute update there.

Proposition 3 (Basic properties)**.** *Let* $\operatorname{osc}(h_x) = \sup_{x' \in \operatorname{supp}\pi_t} h_x(x') - \inf_{x' \in \operatorname{supp}\pi_t} h_x(x')$ *be the oscillation (range) of* h_x *on the support of* π_t*; it equals that of* r_x*, and* $\|h_x\|_\infty \leq \operatorname{osc}(h_x)$ *because* h_x *has mean zero. For every* x*:*

1. $a_t(x)$ *depends on* r_x *only through* h_x*; adding a constant to* r_x *does not change it.*
2. $|r_x(x')| \leq s_{t-1}(x')$ *for every* x'*, so* $\|h_x\|_\infty \leq 2\sup_{x' \in \operatorname{supp}\pi_t} s_{t-1}(x')$.
3. $0 \leq a_t(x) \leq \min\big\{\frac{\eta}{8}\operatorname{osc}(h_x)^2,\ \sqrt{2/\pi}\,\|h_x\|_\infty\big\}$.

Proof. For item (1), adding a constant to r_x adds the same constant to its mean, leaving the centered function h_x unchanged.

For item (2), the Cauchy–Schwarz inequality bounds posterior covariance by the product of posterior standard deviations: $|\Sigma_{t-1}(x', x)| \leq s_{t-1}(x')s_{t-1}(x)$. Divide by the denominator in r_x and use $s_{t-1}(x) \leq (s_{t-1}^2(x) + \lambda)^{1/2}$. Centering can at most double the largest absolute value, which gives the stated bound on h_x.

For item (3), apply Hoeffding's lemma to the centered bounded variable $h_x(X')$, $X' \sim \pi_t$. This controls its log moment-generating function by the square of its range: $\log \mathbb{E}_{\pi_t}[e^{s h_x}] \leq s^2 \operatorname{osc}(h_x)^2/8$. Setting $s = \eta Z$, averaging with $\mathbb{E}Z^2 = 1$, and dividing by η gives the quadratic bound. For the linear bound, bound every exponential by its largest possible value. This gives $\log \mathbb{E}_{\pi_t}[e^{s h_x}] \leq |s|\,\|h_x\|_\infty$; averaging now uses $\mathbb{E}|Z| = \sqrt{2/\pi}$. Nonnegativity was established in the preceding proof. □

An observation is therefore useful only insofar as it can change the relative probabilities assigned by the posterior-mean Gibbs measure. An update that adds the same constant everywhere leaves those probabilities unchanged. Likewise, if the model is already certain on the support of the policy, there is nothing further to learn there. The two bounds in item (3) describe how the acquisition is controlled by the size of the possible update: one is quadratic and the other is linear.

Approximating the acquisition. The logarithm in the exact acquisition is a cumulant-generating function. Cumulants describe features of a distribution: the second is its variance, and the fourth measures the departure of its fourth central moment from the Gaussian value. Expanding this function gives the L1 and L2 approximations, with errors controlled by the range of the update.

Corollary 2 (Cumulant expansion)**.** *Let $\kappa_j(x)$ be the j-th cumulant of $h_x(X')$, $X' \sim \pi_t$; for $j \geq 2$ it is also the j-th cumulant of $r_x(X')$. There are universal constants c, c' such that*

$$\left|a_t(x) - \tfrac{\eta}{2}\kappa_2(x)\right| \leq c'\,\eta^3\,\mathrm{osc}(h_x)^4, \qquad \left|a_t(x) - \tfrac{\eta}{2}\kappa_2(x) - \tfrac{\eta^3}{8}\kappa_4(x)\right| \leq c\,\eta^5\,\mathrm{osc}(h_x)^6.$$

Formally, $a_t(x) = \sum_{k\geq 1} \eta^{2k-1}\kappa_{2k}(x)/(2^k k!)$.

The finite-order bounds control the errors of L1 and L2 directly; they do not require the formal infinite series to converge. The relevant scale is the possible update multiplied by inverse temperature, so uncertainty reduction can improve these approximations even when temperature is fixed.

Proof. Let $K(s) = \log \mathbb{E}_{\pi_t}[e^{sh_x}]$; here K is a scalar function, not the GP Gram matrix. To bound the Taylor remainder, we need to control its derivatives away from zero. Its j-th derivative at u is the j-th cumulant of h_x under the tilted measure proportional to $e^{uh_x}\pi_t$. Tilting changes the probabilities assigned to update values, but their range still has width at most $\mathrm{osc}(h_x)$. Cumulants of the orders used here are polynomials in central moments, so $|K^{(j)}(u)| \leq c_j\,\mathrm{osc}(h_x)^j$ for every u, with c_j depending only on j.

Centering gives $\kappa_1 = 0$. Taylor's theorem to order five therefore gives $K(s) = \sum_{j=2}^{5} \kappa_j s^j/j! + \rho(s)$ with $|\rho(s)| \leq c_6\,\mathrm{osc}(h_x)^6 s^6/720$. Substitute $s = \eta Z$ and average over the standard Gaussian. Its symmetry eliminates the odd powers, while $\mathbb{E}Z^4 = 3$ and $\mathbb{E}Z^6 = 15$ give $\mathbb{E}_Z K(\eta Z) = \eta^2\kappa_2/2 + \eta^4\kappa_4/8 + \mathbb{E}\rho(\eta Z)$ with $|\mathbb{E}\rho(\eta Z)| \leq c_6\,\eta^6 \mathrm{osc}(h_x)^6/48$. Dividing by η gives the second error bound with $c = c_6/48$. Repeating the argument with a third-order Taylor expansion gives the first bound with $c' = c_4/8$. Continuing the same coefficient calculation formally to all orders uses $\mathbb{E}Z^{2k} = (2k-1)!!$ and yields the stated series. □

This yields the three computational choices in Section 3.2: two truncated expansions, and a numerical approximation to the exact objective.

- **OneStep-L1**: $a_t^{(1)} = \frac{\eta}{2}\kappa_2 = \frac{\eta}{2}\mathrm{Var}_{X'\sim\pi_t}\big(\Sigma_{t-1}(X', x)\big)/\big(\Sigma_{t-1}(x,x) + \lambda\big)$, with error $O(\eta^3 \mathrm{osc}(h_x)^4)$. It rewards queries whose possible updates vary across the states currently favored by the policy.
- **OneStep-L2**: $a_t^{(2)} = a_t^{(1)} + \frac{\eta^3}{8}\kappa_4$, with error $O(\eta^5 \mathrm{osc}(h_x)^6)$. Unlike a_t, it can be negative. For a symmetric two-point $h_x = \pm c$, $\kappa_4 = -2c^4$ and $a_t^{(2)} = \frac{\eta c^2}{2}\big(1 - \frac{\eta^2 c^2}{2}\big)$, which is negative once $\eta c > \sqrt{2}$, whereas $a_t = \eta^{-1}\mathbb{E}_Z \log\cosh(\eta c Z) > 0$. The higher-order error bound motivates L2 when $\eta\,\mathrm{osc}(h_x)$ is small. A threshold such as $\eta\,\mathrm{osc}(h_x) \lesssim 2$ is only a heuristic for this regime; it does not guarantee that L2 is more accurate than L1 at a fixed temperature.
- **OneStep-L3**: numerical integration of the exact a_t in (6), without truncating the cumulant expansion. The Gaussian expectation is computed with an L-node Gauss–Hermite rule, $\mathbb{E}_Z\varphi(Z) \approx \sum_{l=1}^{L} w_l \varphi(z_l)$ with $z_l = \sqrt{2}\,\xi_l$ and $w_l = \nu_l/\sqrt{\pi}$, where (ξ_l, ν_l) are the nodes and weights for the weight function $e^{-\xi^2}$. The integrand $\varphi(z) = \eta^{-1}\log \mathbb{E}_{\pi_t}[e^{\eta z h_x}]$ is smooth and convex and grows at most linearly in $|z|$; we use $L = 16$. The rescaling of nodes and weights converts the usual Gauss–Hermite weight to the standard Gaussian density. The resulting approximation retains the full nonlinear dependence on each possible update.

All three use the variance $\Sigma_{t-1}(x,x)$ at the candidate query and its covariances $\Sigma_{t-1}(X', x)$ with points representing the current policy. They do not require a covariance matrix between those policy points, which reduces the cost of using a large particle set.

Relation to the knowledge gradient and to integrated variance reduction. The knowledge gradient values a query by the expected improvement in the best posterior-mean reward after observing it. Equation (A.27) has the same structure, with the maximum replaced by the soft maximum A_η. Both follow a decision-theoretic view of acquisition functions: a query is valuable when it reduces the expected loss of the next decision (Neiswanger et al., 2022; Huang et al., 2026). Here that loss is reverse KL, written as the Bregman divergence in (A.26).

Temperature connects the sampling objective to two familiar design criteria. At low temperature, the policy concentrates near the best reward and the acquisition approaches the knowledge gradient. At high temperature, the policy is close to μ and the acquisition approaches a criterion for reducing average uncertainty, after removing changes that only shift the reward by a constant.

Proposition 4 (High and low temperature limits)**.** *Let $\mathcal{A}$ be compact, let μ have full support, and let m_{t-1} and Σ_{t-1} be continuous. Fix H_t and x.*

1. *As $\eta \to \infty$, $a_t(x)$ tends to the knowledge gradient (Frazier et al., 2008; 2009; Scott et al., 2011)*
$$\mathrm{KG}_t(x) = \mathbb{E}_Z\Big[\max_{x'}\big(m_{t-1}(x') + Z\, r_x(x')\big)\Big] - \max_{x'} m_{t-1}(x').$$

2. *As $\eta \to 0$, $a_t(x)/\eta \to \frac{1}{2}\mathrm{Var}_\mu(r_x) = \frac{1}{2}\big\{\int r_x^2 \,\mathrm{d}\mu - \big(\int r_x \,\mathrm{d}\mu\big)^2\big\}$, where*
$$\int r_x^2 \,\mathrm{d}\mu = \int \Sigma_{t-1}(x',x')\,\mu(\mathrm{d}x') - \int \Sigma_t(x',x')\,\mu(\mathrm{d}x'),$$
$$\Big(\int r_x \,\mathrm{d}\mu\Big)^2 = \mathrm{Var}\Big[\int f \,\mathrm{d}\mu \,\Big|\, H_t\Big] - \mathrm{Var}\Big[\int f \,\mathrm{d}\mu \,\Big|\, H_t, Y\Big].$$

Proof. For the low-temperature limit in item (1), we first show that the soft maximum approaches the ordinary maximum. For continuous g, $A_\eta(g) \le \max g$. For every $\epsilon > 0$, restricting the integral to points within ϵ of the maximum gives $A_\eta(g) \ge \max g - \epsilon + \eta^{-1}\log\mu(\{g > \max g - \epsilon\})$. The set inside this logarithm is open and nonempty, so the full-support assumption gives it positive μ-mass. Its logarithm is therefore finite, and its contribution vanishes as inverse temperature increases. Hence $A_\eta(g) \to \max g$.

Apply this limit to $g = m_{t-1} + Zr_x$ for each fixed Z. To pass the limit through the Gaussian expectation in (A.27), use $|A_\eta(m_{t-1}+Zr_x)| \le \|m_{t-1}\|_\infty + |Z|\,\|r_x\|_\infty$. Continuity on the compact domain bounds both functions on the right, and the standard Gaussian has a finite first absolute moment. Dominated convergence then gives the knowledge-gradient limit.

For the high-temperature limit in item (2), Corollary 2 gives $a_t(x)/\eta = \kappa_2(x)/2 + O(\eta^2 \mathrm{osc}(h_x)^4)$. The leading term also changes with temperature because π_t does. As η tends to zero, its density with respect to μ converges uniformly to 1, so $\kappa_2(x) = \mathrm{Var}_{\pi_t}(r_x) \to \mathrm{Var}_\mu(r_x)$. The remainder vanishes, proving the claimed limit.

The two variance-reduction identities are direct consequences of $\Sigma_t = \Sigma_{t-1} - r_x \otimes r_x$. On the diagonal, this update subtracts the square of r_x from the pointwise variance; integrating gives the first identity. For the second, the variance of an integrated function is the double integral of its covariance: $\mathrm{Var}[\int f \,\mathrm{d}\mu \mid \cdot\,] = \iint \Sigma \,\mathrm{d}\mu\,\mathrm{d}\mu$. Integrating the rank-one covariance reduction in both arguments gives the square of the integral of r_x, as stated. □

The low-temperature limit recovers the optimization objective of selecting a state with high reward, for which the knowledge gradient is the Bayes-optimal one-step rule. The high-temperature limit has two parts. The first, $\int r_x^2 \,\mathrm{d}\mu$, is the reduction in posterior variance averaged across the domain, a standard active-learning objective for regression (Cohn et al., 1996). The second is the reduction in uncertainty about the average reward $\int f \,\mathrm{d}\mu$, the quantity targeted by Bayesian quadrature (O'Hagan, 1991). Subtracting it removes uncertainty in the constant component of the reward, which has no effect on the normalized sampling distribution. Thus learning the high-temperature target amounts, to first order, to learning relative rewards under a squared-error criterion weighted by μ.

OneStep-L1 is exactly this expression with μ replaced by the current Gibbs measure:

$$a_t^{(1)}(x) = \frac{\eta}{2}\Big\{\int r_x^2 \,\mathrm{d}\pi_t - \Big(\int r_x \,\mathrm{d}\pi_t\Big)^2\Big\},$$

which is integrated variance reduction weighted by the current policy, with the constant component removed. The uncentered first term is the Boltzmann-weighted integrated variance of Qing et al. (2026), derived for regression accuracy. Related integrated-variance criteria have also been used for approximate Bayesian computation (ABC) posteriors (Järvenpää et al., 2019). The exact objective

approximated by OneStep-L3 retains both temperature regimes. For small updates it behaves like a variance, while Proposition 3 bounds its growth by a linear function for larger updates, as expected from its connection to a maximum.

Estimation and optimization. We approximate expectations under π_t using M equally weighted sample locations, or particles, $X_1, \ldots, X_M$. For the toys and alanine, we draw these from a grid representation of the policy (Appendices C and D). For DW-4 and Planck-lite, we use the final states of random-walk Metropolis chains targeting the posterior-mean Gibbs distribution (Appendix B).

Much of the covariance calculation can be shared across candidate queries. With a Cholesky factor $LL^\top = K_{t-1} + \lambda I$ and $v(x) = L^{-1}k_{t-1}(x)$, we have $\Sigma_{t-1}(X_i, x) = k(X_i, x) - v(X_i)^\top v(x)$. We compute $v(X_i)$ once per round; each candidate then requires one triangular solve. Once these covariances are available, evaluating $\widehat{a}_t^{(1)}$ or $\widehat{a}_t^{(2)}$ costs $O(M)$, and evaluating $\widehat{a}_t^{(3)}$ costs $O(LM)$, where L is the number of quadrature nodes. These costs refer to evaluating the acquisition after the covariances have been computed; they do not include fitting the GP or producing the particles.

For independent exact draws from π_t, replacing population expectations by particle averages gives consistent estimates as $M \to \infty$, with bias $O(1/M)$. For L3, this convergence is to the fixed-quadrature approximation; quadrature error remains. The logarithm is concave, so its expectation when applied to a finite particle average is below the logarithm of the population average. The empirical centering term has the correct expectation, so Jensen's inequality makes L3's particle-estimation bias downward. Finite grids and finite-length chains introduce further approximation error beyond these independent-draw calculations.

We hold the particles fixed while optimizing the acquisition. This gives a deterministic, differentiable objective as a function of x, avoiding new Monte Carlo noise at every optimization step. We initialize searches from particles with high acquisition values and from uniform draws in the domain, refine each candidate with Adam, and select the best result. The numbers of starts and optimization steps are given with the individual experiment protocols.

Batches. To select $B > 1$ queries before receiving their observations, we build the batch one location at a time. After selecting $x_{(1)}$, we update the covariance by $\Sigma \leftarrow \Sigma - r_{x_{(1)}} \otimes r_{x_{(1)}}$. This update is exact because it does not depend on the observed value. We keep the posterior mean, the policy π_t, and its particles fixed. This is the kriging-believer approximation (Ginsbourger et al., 2010): it treats each pending observation as though it equaled its current predicted mean. At each selection, the update direction is computed using the current working covariance, including the reductions from earlier selections in the batch.

The next query maximizes the acquisition with the reduced covariance. It therefore accounts for information already expected from the pending queries, while assuming that those queries will not change the mean. This greedy construction is a practical batch approximation; it need not optimize the joint value of all B observations. The one-step optimality result above applies to $B = 1$.

B GP IMPLEMENTATION AND COMMON EXPERIMENTAL SETUP

All experiments share the GP, query rules and evaluation conventions described here. Table 1 lists the settings that differ between experiments; Appendices C, D, E and F give the target, reference, baseline configurations and complete results of each experiment.

Oracle accounting. One oracle call evaluates the energy (and its gradient, for gradient-based samplers) at one input. Budgets count every call a method makes, including the initial design, rejected proposals and all replicas; reference construction and evaluation-only calls are not counted. Neural budgets on alanine count emulator calls (Appendix D.4).

Surrogate. The implementation fits the energy; we write m_{t-1}^E for its posterior mean, so the reward mean of the main text is $m_{t-1} = -m_{t-1}^E$ and the surrogate's Gibbs measure is $S_\eta(-m_{t-1}^E) \propto e^{-\eta m_{t-1}^E}$. The GP is an exact BoTorch `SingleTaskGP` (Balandat et al., 2020; Gardner et al., 2018) with inputs mapped to the unit cube and standardized outputs; an RBF kernel with one length-scale per input dimension and no output scale (length-scale prior log-normal with location $\sqrt{2} + \frac{1}{2}\log d$,

Table 1: Target and GP-method settings for each experiment. IQR denotes interquartile range, SEM standard error of the mean, and s.d. standard deviation.

	2-D toys	Alanine dipeptide	DW-4	Planck-lite ΛCDM
State	$x \in [-4, 4]^2$	torsions (ϕ, ψ)	8 centered Cartesian coordinates	6 rescaled parameters in $[-8, 8]^6$
Oracle	analytic energy	relaxed GFN2-xTB energy at 300 K	analytic pair energy	CLASS and Planck-lite likelihoods
GP inputs	box coordinates	cos and sin of ϕ and ψ	range of the training inputs	the fixed box
Initial design	20 uniform points	20 uniform points	50 centered draws from $\mathcal{N}(0, 1.5^2 I)$	18 draws from Cobaya's reference distribution
Batch size	10	10	50	6
Budget	2,000 calls	1,000 calls	20,000 calls	600 calls
Seeds	0–2	0–2	0–5	42–46
Energy clip	best observed $+50$	best observed $+50\, k_B T$	2,000	best observed $+300$
Initial mean curvature	–	–	0.69	40
Surrogate sampling	categorical, 400×400 grid	categorical, 72×72 grid	10,000 RWMH steps, proposal s.d. 0.1	5,000 adapted RWMH steps
Chain starts	–	–	training points within 50 of the best, plus noise of s.d. 0.01	training points within 50 of the best
Adam learning rate	0.05	0.05	0.01	0.05
Optimization box	the domain	$(\phi, \psi)/45^\circ \in [-4, 4]^2$	particle range ± 3	$[-8, 8]^6$
Reference	exact on the grid	72×72 relaxed scan	10^4 long-chain states	258,948-row bank from 20 chains
Metric	D_J; TV on 50×50 cells	D_J; TV on 36×36 cells	TV of pair distances and energies	D_J
Reported	median [IQR]	median [IQR]	median [IQR] at 20,000	mean ± SEM at 150, 300, 600

scale $\sqrt{3}$ and lower bound 0.025); Gaussian noise with a $\mathrm{LogNormal}(-4, 1)$ prior and floor 10^{-4} in standardized units; and a learned diagonal quadratic mean $c + \frac{1}{2}\sum_i a_i (z_i - \mu_i)^2$ with $a_i \geq 10^{-3}$ (constant for the two LogExp variants on the toys and alanine). Energies are clipped from above before each fit (Table 1), the hyperparameters are refitted after each round by maximizing the marginal likelihood plus the log prior, warm-started from the previous values, and $\eta = 1$ throughout.

Sampling from the surrogate. GP-PM samples $S_\eta(-m^E_{t-1})$, GP-UCS samples $S_\eta(-(m^E_{t-1} - \beta s_{t-1}))$ with $\beta = 0.5$, and GP-TS^2 samples the Gibbs measure of a fresh Matheron path (Wilson et al., 2020) for each query. On the toys and alanine, the Gibbs measure is sampled exactly on a grid (a cell is drawn with probability proportional to e^{-m^E} at its center, plus uniform jitter within the cell); on DW-4 and Planck-lite, by random-walk Metropolis (Table 1). OneStep draws its particles in the same way.

Algorithm 1 GP learning with Gibbs policies or OneStep acquisitions

Require: Initial observations $\mathcal{D}$ of $f = -E$, rounds R, batch size B, inverse temperature η; rule GP-PM, GP-UCS, GP-TS^2, or OneStep-Lℓ with $\ell \in \{1, 2, 3\}$.

```
Fit the GP to D; denote its reward mean and covariance by m and Σ.
for r = 1, ..., R do
    if the rule is OneStep then
        Draw particles X_{1:M} ~ S_η(m) and set Σ^work ← Σ.
    end if
    for j = 1, ..., B do
        if the rule is OneStep-Lℓ then
            Choose x_j by maximizing â^(ℓ) in (B.2) using Σ^work.
            Reduce Σ^work at x_j using (4); keep m and particles fixed.
        else
            Form g_j for the chosen Gibbs rule, drawing a fresh posterior path for GP-TS^2.
            Draw the query x_j ~ S_η(g_j).
        end if
    end for
    Query the oracle at x_{1:B} to obtain reward observations y_{1:B}.
    Append {(x_j, y_j)}_{j=1}^B to D and refit the GP, updating m and Σ.
end for
return the fitted GP; its mean defines the learned density S_η(m).
```

OneStep. With $M = 4{,}096$ particles $X_i \sim S_\eta(-m^E_{t-1})$ and the centered update $\widehat{h}_x(X_i) = r_x(X_i) - M^{-1}\sum_j r_x(X_j)$ from (4), the empirical cumulants are

$$\widehat{\kappa}_2(x) = \frac{1}{M}\sum_i \widehat{h}_x(X_i)^2, \qquad \widehat{\kappa}_4(x) = \frac{1}{M}\sum_i \widehat{h}_x(X_i)^4 - 3\widehat{\kappa}_2(x)^2, \tag{B.1}$$

and the three acquisitions are

$$\begin{aligned} \widehat{a}^{(1)}_t(x) &= \tfrac{\eta}{2}\widehat{\kappa}_2(x), \\ \widehat{a}^{(2)}_t(x) &= \widehat{a}^{(1)}_t(x) + \tfrac{\eta^3}{8}\widehat{\kappa}_4(x), \\ \widehat{a}^{(3)}_t(x) &= \frac{1}{\eta}\sum_{l=1}^{L} w_l \log\left(\frac{1}{M}\sum_i e^{\eta z_l \widehat{h}_x(X_i)}\right), \end{aligned} \tag{B.2}$$

with (z_l, w_l) the $L = 16$ Gauss–Hermite nodes and weights of a standard Gaussian (Appendix A.5). Each acquisition is maximized from 64 starts, the 48 highest-scoring particles and 16 uniform points in the optimization box, by 60 projected Adam steps; the unoptimized starts remain candidates. Batches are chosen greedily with the rank-one covariance downdate of (4) and fixed mean and particles (kriging believer). Algorithm 1 gives the loop.

GP baselines. With $u = -m^E_{t-1}$ and posterior variance s^2, BAPE-EV (Kandasamy et al., 2015) maximizes $2u + s^2 + \log(e^{s^2} - 1)$ and GPry-LogExp (El Gammal et al., 2023) maximizes $2\zeta u + \frac{1}{2}\log(s^2 - \sigma_n^2)$ with $\zeta = d^{-0.85}$ and σ_n equal to 10^{-2} output standard deviations, using the same GP, particles, optimizer and batching as OneStep. LogExp pool (toys and alanine) ranks the same score over the particles instead of optimizing it (Torrado et al., 2023). Uniform + GP fits the GP to uniform queries.

MCMC samplers. RWMH, MALA, HMC and PT run independent chains, adapt their step sizes by Robbins–Monro updates during an initial phase, and discard an initial fraction of each chain from raw histograms; every PT replica is charged and only cold replicas are scored. SMC (Del Moral et al., 2006) uses $N = \min(200, \max(20, \lfloor b/20 \rfloor))$ particles at budget b, temperature increments targeting an effective sample size of $N/2$, systematic resampling and one Metropolis move per level. PMC (Cappé et al., 2004; Cornuet et al., 2012) adapts eight Gaussian components plus a 10% uniform component with 20 draws per round. "+ GP" fits the shared GP to every finite-energy in-domain input a sampler evaluated, including rejected proposals and all replicas, and scores the fitted density like our methods. VBMC (Acerbi, 2018) uses PyVBMC 1.0.4 (Huggins et al., 2023) with hard bounds $[-4, 4]^2$, plausible bounds $[-3.5, 3.5]^2$ and its own 20-point uniform design.

Neural samplers. Neural samplers are trained from energy evaluations alone, and their budgets count training evaluations. On DW-4 we use released or reference implementations (Appendix E.4); on the toys and alanine, local implementations of a Boltzmann generator (Noé et al., 2019), a FAB-style flow (Midgley et al., 2023), iDEM (Akhound-Sadegh et al., 2024), GFlowNet-TB (Sendera et al., 2024) and PIS (Zhang & Chen, 2022): eight-layer RealNVP flows with two 128-unit hidden layers (Adam, learning rate 10^{-3}, batches of 256); iDEM with 50 energy samples per score estimate, batches of 128, a 10,000-state buffer and 50 inner updates; and 100-step controlled SDEs for GFlowNet-TB and PIS. The FAB-style flow anneals toward p rather than FAB's p^2/q, training uses a soft wall outside the domain, and all five are scored by histograms of 200,000 samples. These runs compare the stated implementations, not the published methods.

Scoring. Learned densities are scored by $D_J = D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|p)$, with both densities normalized on the reference grid and log probabilities floored at -690 (toys, alanine), or estimated by reweighting a reference bank (Planck-lite, Appendix F.3). Samplers without a density are scored by the TV between a sample histogram and the reference on a coarse grid (on DW-4, the pair-distance and energy histograms). Curves use the latest checkpoint at or below each budget; threshold counts are the first recorded crossing, with runs that never cross counted as $+\infty$ and a dash meaning that fewer than two seeds crossed. Results are medians and interquartile ranges over seeds (means and standard errors on Planck-lite).

C TWO-DIMENSIONAL TOY DENSITIES

Six analytic targets on $[-4, 4]^2$, three seeds, and 2,000 oracle calls per density-learning run (500 for VBMC); separate classical runs extend the raw-sample comparison to 10^6 calls. The surrogate is sampled exactly on a grid, so the comparison isolates query selection from chain mixing. This is the only experiment with a computation and sampling-cost comparison (Appendix C.4).

C.1 TARGETS

Each target is a Boltzmann density $p(x) \propto e^{-E(x)}$ on $\mathcal{X} = [-4, 4]^2$ (Figure 5):

- *8 Gaussians*: eight equally weighted isotropic Gaussians ($\sigma = 0.35$) on a circle of radius 2.8.
- *Two moons*: the interleaving half-circles of `sklearn`, scaled by 2.5, each represented by 100 equally weighted isotropic Gaussians ($\sigma = 0.2$).
- *Two spirals*: the arms $\pm(-n\cos n,\ n\sin n)/3$, $n \in (0, 3\pi]$, each represented by 200 Gaussians ($\sigma = 0.15$) with weights $\propto n$.
- *Checkerboard*: $E(x) = \beta(1 - o(x))$ with $\beta = 12$, $o = s_1 s_2 + (1 - s_1)(1 - s_2)$ and $s_i = \sigma(20\sin(\pi x_i/2))$ for the logistic sigmoid σ: squares of side 2 alternating between energies 0 and 12.
- *Rings*: radii $r_i \in \{1, 2, 3\}$ with Gaussian radial profiles ($\sigma = 0.12$), $p(x) = \sum_i w_i \mathcal{N}(\|x\|; r_i, \sigma)/(2\pi\|x\|)$, mass proportional to circumference.
- *Pinwheel*: five curved blades (Grathwohl et al., 2019), each represented by 60 Gaussians ($\sigma = 0.2$) with Gaussian radial weights.

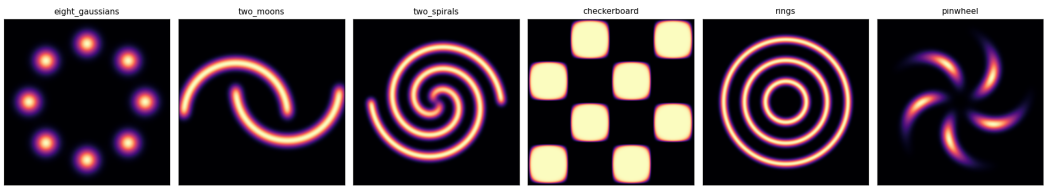

Figure 5: The six two-dimensional targets on $[-4, 4]^2$.

Table 2: Jeffreys divergence after 300 calls for all toy methods. Entries are medians over three seeds, using the latest available checkpoint at or below the budget.

Method	Checkerboard	Two spirals	Two moons	8 Gaussians	Rings	Pinwheel
GP-PM	2.31	2.96	1.66	1.5	5.64	1.06
GP-UCS	2.66	2.11	0.103	0.717	4.23	0.566
GP-TS^2	1.21	0.812	0.041	0.032	1.03	0.048
OneStep-L1	0.829	0.082	0.0082	0.005	0.353	0.0093
OneStep-L2	0.861	0.073	0.008	0.0073	0.36	0.013
OneStep-L3	0.767	0.09	0.0085	0.0082	0.239	0.012
GPry-LogExp	1.76	0.568	0.0063	0.826	1.09	0.036
LogExp pool	1.59	0.106	0.0057	0.816	1.62	0.027
BAPE-EV	1.09	0.945	0.065	0.027	1.01	0.075
Uniform + GP	4.47	1.76	0.239	0.058	2.89	0.553
VBMC	25.73	0.982	0.057	4.95	19.23	5.12
RWMH + GP	5.37	1.33	0.887	0.096	4.11	0.378
MALA + GP	5.46	7.99	0.903	2.06	8.22	2.12
HMC + GP	4.55	2.16	1.38	0.784	9.14	1.41
PT + GP	6.58	6.21	4.63	1.14	10.35	1.46
SMC + GP	4.02	1.83	1.07	0.257	3.53	0.632
PMC + GP	4.89	4.49	4.9	1.17	3.38	1.67

C.2 Protocol

GP methods start from the same 20 uniform points per seed and query batches of ten. Gibbs queries and particles are drawn on a 400×400 grid, on which D_J is computed; TV is aggregated to 50×50 cells. RWMH, MALA, HMC (five leapfrog steps) and PT (four geometric temperatures up to $T = 10$) use ten chains from uniform starts, adapt during the first 10% of the budget and discard the first quarter of each chain; the 10^6-call runs are separate experiments. SMC starts from the uniform base measure with one random-walk move per level, and VBMC uses 500 evaluations. The neural implementations of Appendix B are trained with 2×10^7 calls (flows), 5×10^7 (iDEM) and 5×10^6 (GFlowNet-TB and PIS).

C.3 Complete results

Figure 7 and Tables 2–4 give all density estimators, Figures 8 and 9 the two KL directions, and Figure 6 the per-step sampling regret of the Gibbs policies. The three OneStep levels have similar curves. On checkerboard, BAPE-EV has the lower terminal D_J (0.041 versus 0.046 for L1) while L1 reaches $D_J < 0.1$ earlier.

C.4 Computation, oracle and sampling costs

Figures 10 and 11 and Tables 5 and 6 extend Figure 1, where OneStep denotes L3, to all 22 methods at TV < 0.3 and < 0.1. For each of the 18 target–seed runs we take the first checkpoint below the threshold, or the final budget if none; medians and interquartile ranges include all endpoints, and hollow (starred) markers mean that fewer than half the runs succeeded. Internal computation is the fitting and acquisition time through that checkpoint for the GP methods and the training time for the neural methods (RTX A5500 GPUs); classical methods are shown at a 10^{-3} s floor, as they only involve simple operations. At TV < 0.3, L1/L2/L3 use median $115/110/130$ calls and $93/145/182$ s, and LogExp pool 190 calls and 17 s; the value of trading speed for better queries depends on the cost of an energy evaluation. Sampling cost is the time per state of $1{,}000$ batched RWMH transitions on the final surrogate ($4{,}096$ chains), the time per sample of the final neural model, a $2\,\mu$s proxy for VBMC, and, for the classical samplers, an assumed oracle cost of 1 s per evaluation multiplied by the integrated autocorrelation time (RWMH, MALA, HMC; four times the random-walk value for PT) or divided by the particle count (SMC) or the effective sample size (PMC). These proxies do not give the cost of equally accurate independent draws.

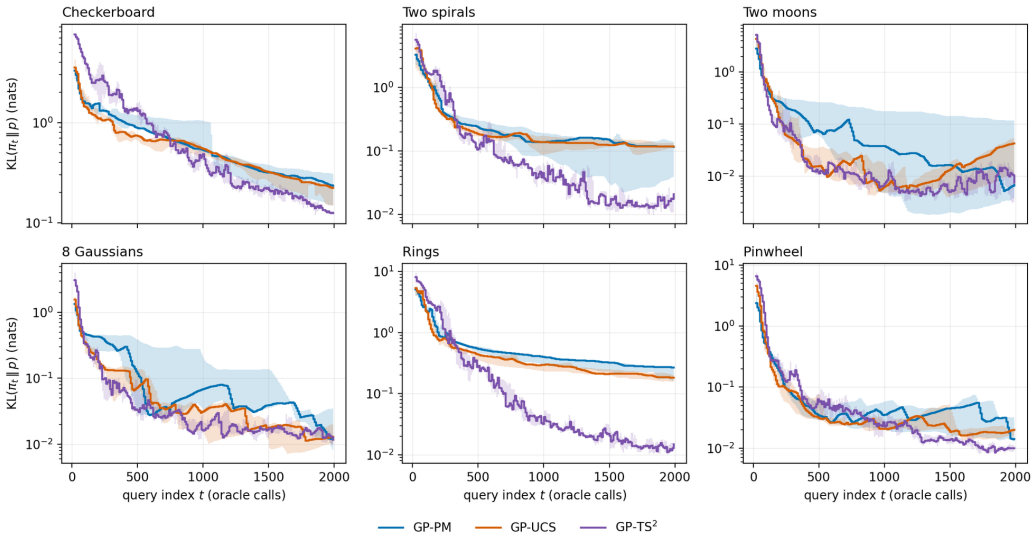

Figure 6: Per-step sampling regret of the Gibbs policies. GP-PM has the lowest regret at first; the order reverses as the budget grows, and GP-TS^2 ends lowest without a tuned exploration bonus. Only these three stochastic policies have a finite KL regret.

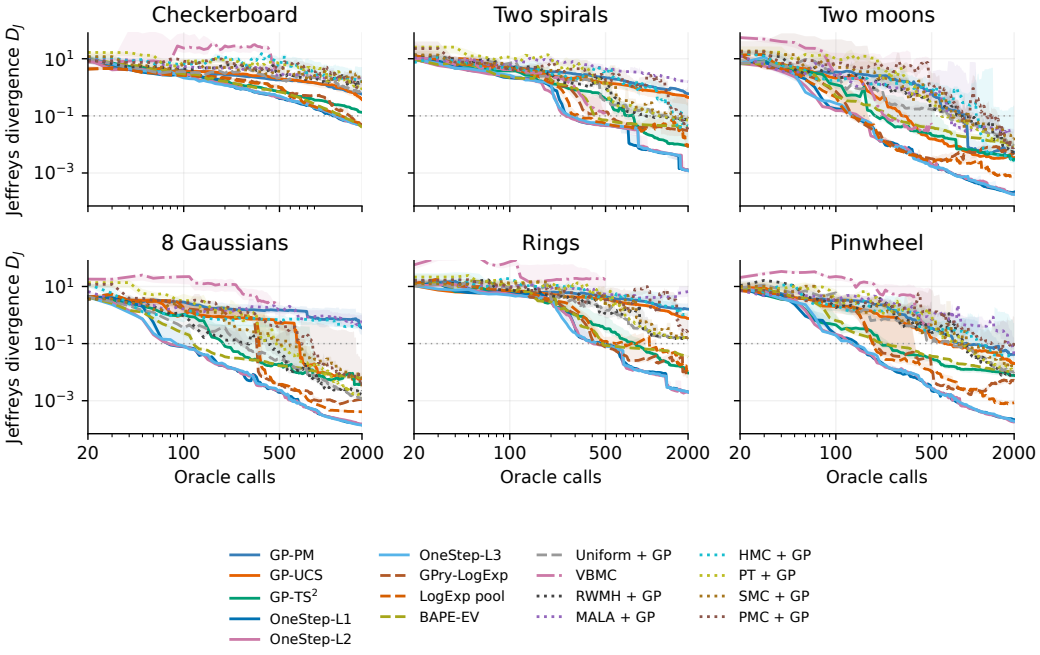

Figure 7: Jeffreys divergence for all toy density estimators: medians and interquartile ranges over three seeds. "+ GP" evaluates the fitted density, and VBMC stops at 500 calls. The horizontal line marks $D_J = 0.1$.

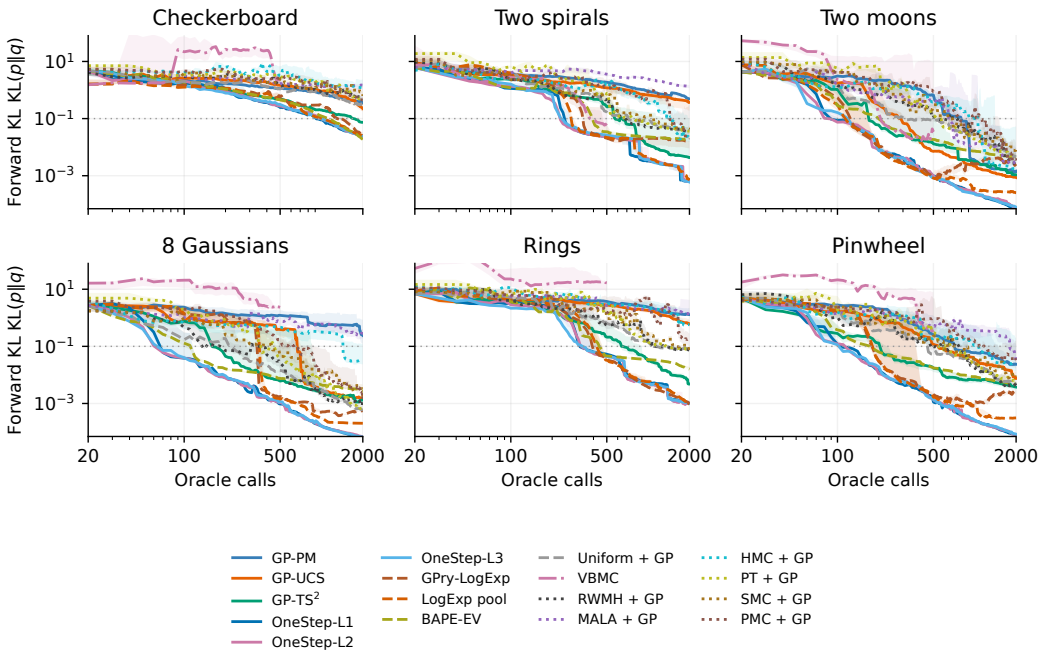

Figure 8: Forward KL $\mathrm{KL}(p\|q)$ for all toy density estimators (medians and interquartile ranges over three seeds); this direction penalizes missing target mass.

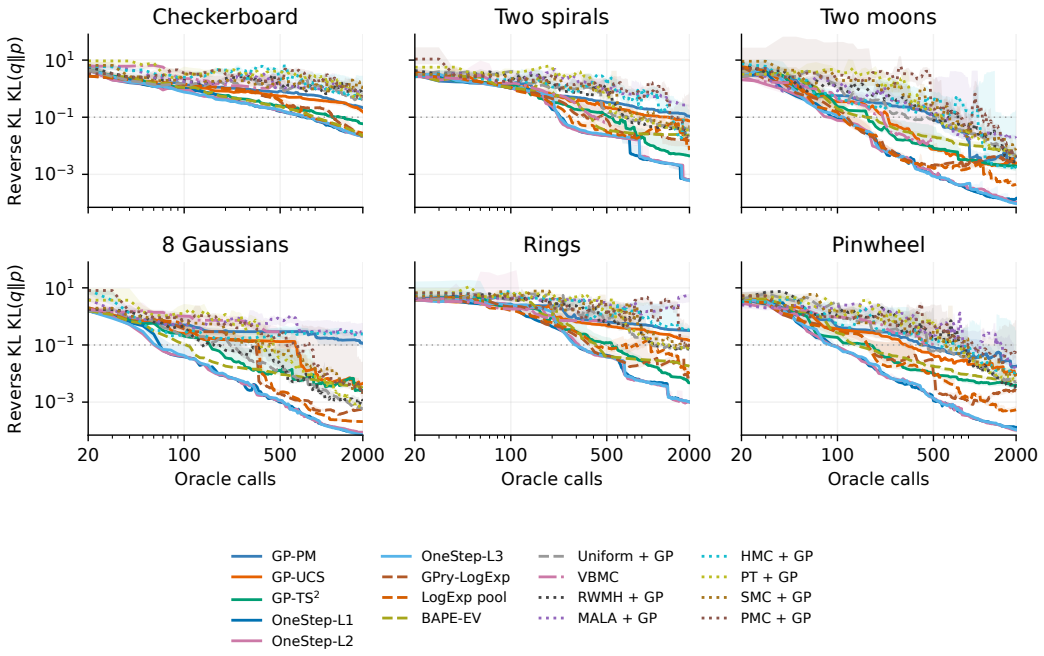

Figure 9: Reverse KL $\mathrm{KL}(q\|p)$ for all toy density estimators (medians and interquartile ranges over three seeds); this direction penalizes surrogate mass in low target-density regions.

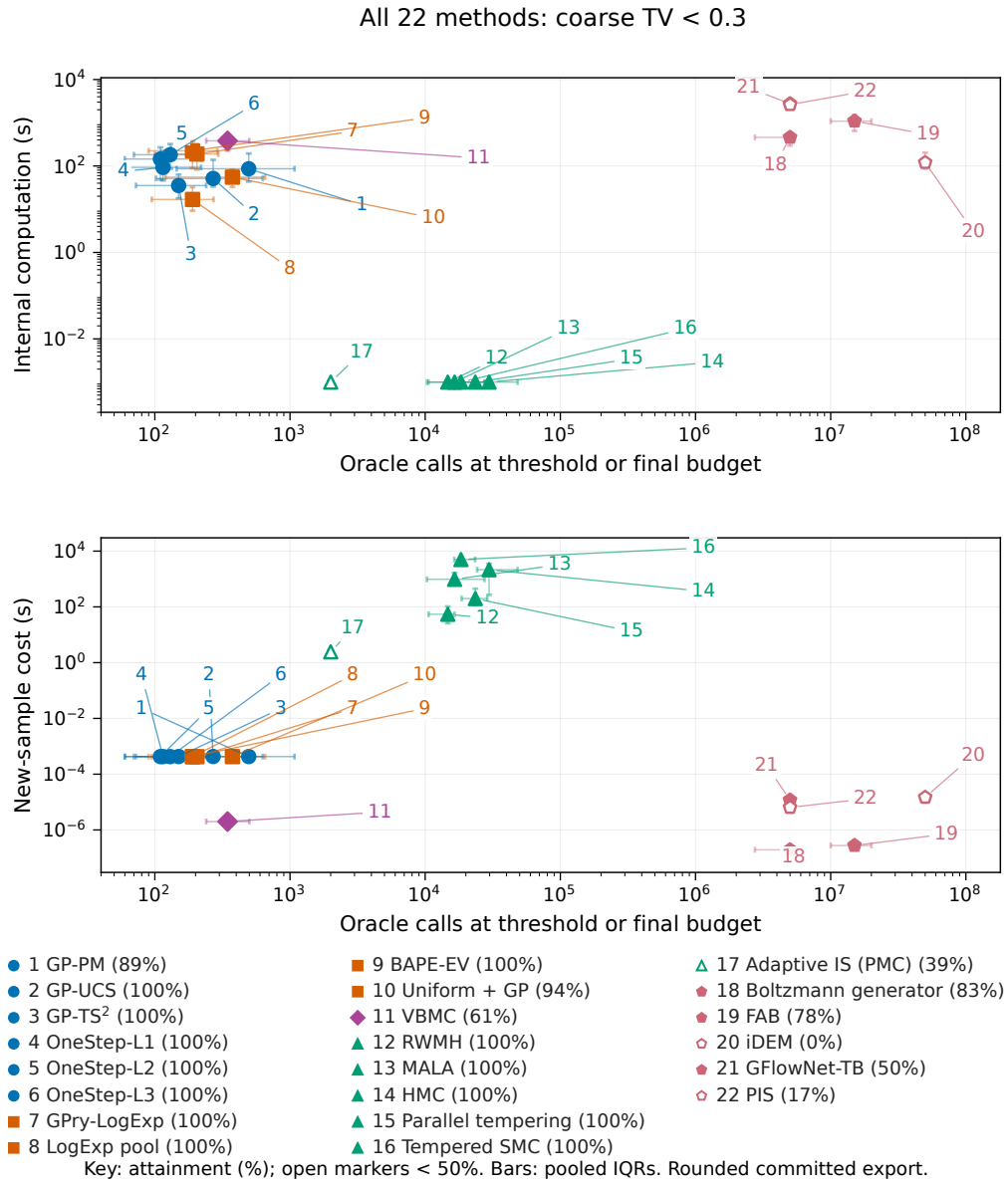

Figure 10: All 22 methods at TV < 0.3 on the 50×50 grid: oracle calls versus internal computation (top) and the cost of an additional sample (bottom). Medians and interquartile ranges over six targets and three seeds, including the final endpoints of runs that did not reach the threshold; hollow markers mean that fewer than half the runs succeeded. The timing proxies are described in Appendix C.4; Table 5 gives the values.

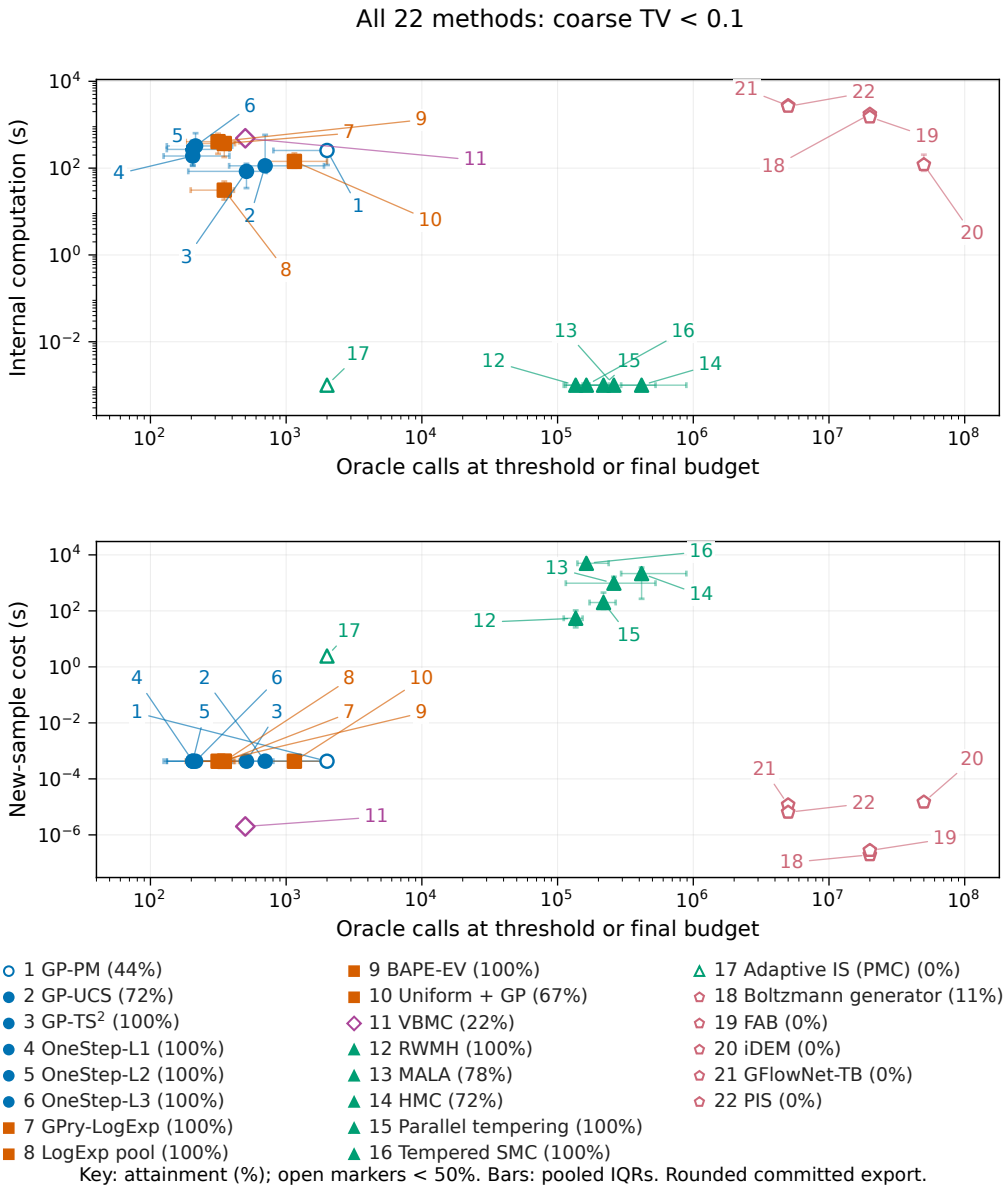

Figure 11: As Figure 10, at the stricter threshold TV < 0.1. Table 6 gives the values.

Table 3: Final Jeffreys divergence for all toy methods, with medians over three seeds evaluated on the 400×400 grid. Classical + GP rows score the density fitted to the queried inputs and their observed energies, rather than the sampler output. VBMC uses its separate 500-call budget.

Method	Calls	Checkerboard	Two spirals	Two moons	8 Gaussians	Rings	Pinwheel
GP-PM	2,000	0.579	0.597	0.0052	0.36	1.61	0.042
GP-UCS	2,000	0.378	0.443	0.0034	0.0061	0.758	0.02
GP-TS^2	2,000	0.133	0.0087	0.003	0.0039	0.0094	0.0077
OneStep-L1	2,000	0.046	0.0013	0.00022	0.00014	0.002	0.00022
OneStep-L2	2,000	0.046	0.0013	0.00018	0.00015	0.002	0.00018
OneStep-L3	2,000	0.047	0.0012	0.00018	0.00014	0.002	0.00019
GPry-LogExp	2,000	0.047	0.031	0.0052	0.0011	0.0086	0.0051
LogExp pool	2,000	0.053	0.0077	0.0007	0.00041	0.012	0.00085
BAPE-EV	2,000	0.041	0.042	0.011	0.0056	0.034	0.01
Uniform + GP	2,000	0.781	0.075	0.0092	0.0012	0.165	0.01
VBMC	500	3.57	0.133	0.05	2.64	17.67	0.468
RWMH + GP	2,000	1.3	0.07	0.0054	0.0022	0.142	0.0081
MALA + GP	2,000	2.23	1.58	0.033	0.565	6.63	0.041
HMC + GP	2,000	1.62	0.046	0.0025	0.433	1.1	0.036
PT + GP	2,000	0.95	0.071	0.0086	0.0013	0.158	0.012
SMC + GP	2,000	1.29	0.095	0.017	0.0032	0.245	0.018
PMC + GP	2,000	0.958	0.04	0.008	0.0028	0.236	0.087

Table 4: Oracle calls at the first recorded crossing of $D_J < 0.1$, with medians over three seeds. A run that never crosses contributes $+\infty$ to the median. A dash therefore means that fewer than two seeds crossed within 2,000 calls, or 500 calls for VBMC.

Method	Checkerboard	Two spirals	Two moons	8 Gaussians	Rings	Pinwheel
GP-PM	–	–	880	–	–	830
GP-UCS	–	–	310	710	–	620
GP-TS^2	–	730	200	200	830	220
OneStep-L1	1220	270	140	80	480	150
OneStep-L2	1270	260	150	80	410	130
OneStep-L3	1290	280	140	90	440	140
GPry-LogExp	1630	540	140	350	440	190
LogExp pool	1430	320	130	360	600	190
BAPE-EV	1420	430	250	120	480	230
Uniform + GP	–	1320	660	270	–	690
VBMC	–	–	230	–	–	–
RWMH + GP	–	990	800	290	–	870
MALA + GP	–	–	700	–	–	1820
HMC + GP	–	1560	1060	–	–	1090
PT + GP	–	1230	590	460	–	870
SMC + GP	–	1600	700	500	–	740
PMC + GP	–	1800	840	720	–	1380

C.5 Limitations of the toy comparisons

VBMC completes all runs, but four end with D_J between 16.9 and 19.9, and PMC's extension to longer budgets did not finish. The neural energies use a soft wall outside the box while the benchmark restricts the target to it, so out-of-box samples are discarded and the histogram renormalized: the reported TV is conditional on the box and can hide escaped mass. Together with the FAB-style objective (Appendix B), the neural scores therefore describe the tested implementations rather than the published methods.

Table 5: Toy cost comparison at coarse $\mathrm{TV} < 0.3$. Entries report the median [interquartile range] over six targets and three seeds. A run that does not reach the threshold contributes its final budget. Appendix C.4 explains the aggregation and timing proxies. Each power of ten multiplies all three reported values; identical quartiles are omitted.

No. / method	Successes	Oracle calls	Internal (s)	Sampling (s)
GP query rules				
1. GP-PM	16/18	$495\,[145, 1080]$	$86.2\,[43.5, 195]$	$428\,[426, 431]\times10^{-6}$
2. GP-UCS	18/18	$270\,[102, 628]$	$51.5\,[33.5, 139]$	$430\,[428, 432]\times10^{-6}$
3. GP-TS^2	18/18	$150\,[72.5, 240]$	$35.3\,[17.9, 63.2]$	$429\,[428, 431]\times10^{-6}$
4. OneStep-L1	18/18	$115\,[60, 220]$	$92.7\,[46.1, 222]$	$428\,[427, 429]\times10^{-6}$
5. OneStep-L2	18/18	$110\,[60, 220]$	$145\,[48.2, 270]$	$427\,[427, 429]\times10^{-6}$
6. OneStep-L3	18/18	$130\,[70, 210]$	$182\,[86.2, 322]$	$427\,[427, 428]\times10^{-6}$
Prior GP acquisitions				
7. GPry-LogExp	18/18	$205\,[100, 295]$	$189\,[82.6, 298]$	$423\,[422, 425]\times10^{-6}$
8. LogExp pool	18/18	$190\,[95, 272]$	$16.8\,[9.24, 32.2]$	$425\,[425, 426]\times10^{-6}$
9. BAPE-EV	18/18	$190\,[90, 290]$	$225\,[89.5, 376]$	$427\,[427, 428]\times10^{-6}$
10. Uniform + GP	17/18	$375\,[120, 655]$	$55.2\,[32.9, 82.7]$	$427\,[427, 428]\times10^{-6}$
Variational inference				
11. VBMC	11/18	$345\,[240, 500]$	$381\,[226, 492]$	2×10^{-6}
Classical samplers				
12. RWMH	18/18	$14.7\,[10.6, 16.5]\times10^{3}$	1×10^{-3}	$54.4\,[25.8, 104]$
13. MALA	18/18	$16.5\,[10.3, 27.5]\times10^{3}$	1×10^{-3}	$974\,[727, 1650]$
14. HMC	18/18	$29.7\,[24.2, 48.3]\times10^{3}$	1×10^{-3}	$2.15\,[0.27, 3.52]\times10^{3}$
15. PT	18/18	$23.5\,[18.6, 28.8]\times10^{3}$	1×10^{-3}	$199\,[117, 450]$
16. SMC	18/18	$18.4\,[16.4, 23.4]\times10^{3}$	1×10^{-3}	5×10^{3}
17. PMC	7/18	$2\,[1.92, 2]\times10^{3}$	1×10^{-3}	$2.44\,[1.96, 2.91]$
Neural samplers				
18. Boltzmann generator	15/18	$5\,[2.75, 5]\times10^{6}$	$459\,[296, 597]$	$194\,[179, 208]\times10^{-9}$
19. FAB-style	14/18	$15\,[10, 20]\times10^{6}$	$1.09\,[0.647, 1.52]\times10^{3}$	$277\,[209, 410]\times10^{-9}$
20. iDEM	0/18	50×10^{6}	$120\,[86.7, 204]$	$14.8\,[9.53, 21.3]\times10^{-6}$
21. GFlowNet-TB	9/18	5×10^{6}	$2.75\,[2.61, 2.89]\times10^{3}$	$11.8\,[11, 13.7]\times10^{-6}$
22. PIS	3/18	5×10^{6}	$2.62\,[2.54, 2.75]\times10^{3}$	$6.43\,[4.15, 9.05]\times10^{-6}$

Table 6: Toy cost comparison at coarse TV < 0.1. Entries report the median [interquartile range] over six targets and three seeds. A run that does not reach the threshold contributes its final budget. Appendix C.4 explains the aggregation and timing proxies. Each power of ten multiplies all three reported values; identical quartiles are omitted.

No. / method	Successes	Oracle calls	Internal (s)	Sampling (s)
GP query rules				
1. GP-PM	8/18	$2\,[0.805, 2]\times10^{3}$	$255\,[118, 298]$	$428\,[426, 431]\times10^{-6}$
2. GP-UCS	13/18	$700\,[380, 1900]$	$113\,[93.2, 584]$	$430\,[428, 432]\times10^{-6}$
3. GP-TS^2	18/18	$510\,[190, 720]$	$84.2\,[34.7, 126]$	$429\,[428, 431]\times10^{-6}$
4. OneStep-L1	18/18	$205\,[125, 382]$	$190\,[113, 428]$	$428\,[427, 429]\times10^{-6}$
5. OneStep-L2	18/18	$205\,[132, 382]$	$270\,[116, 386]$	$427\,[427, 429]\times10^{-6}$
6. OneStep-L3	18/18	$215\,[132, 400]$	$324\,[189, 638]$	$427\,[427, 428]\times10^{-6}$
Prior GP acquisitions				
7. GPry-LogExp	18/18	$350\,[198, 420]$	$369\,[180, 435]$	$423\,[422, 425]\times10^{-6}$
8. LogExp pool	18/18	$350\,[198, 412]$	$31.1\,[18.8, 49.1]$	$425\,[425, 426]\times10^{-6}$
9. BAPE-EV	18/18	$315\,[182, 450]$	$407\,[214, 621]$	$427\,[427, 428]\times10^{-6}$
10. Uniform + GP	12/18	$1.15\,[0.685, 2]\times10^{3}$	$143\,[102, 219]$	$427\,[427, 428]\times10^{-6}$
Variational inference				
11. VBMC	4/18	500	$487\,[413, 580]$	2×10^{-6}
Classical samplers				
12. RWMH	18/18	$136\,[111, 153]\times10^{3}$	1×10^{-3}	$54.4\,[25.8, 104]$
13. MALA	14/18	$260\,[115, 527]\times10^{3}$	1×10^{-3}	$974\,[727, 1650]$
14. HMC	13/18	$416\,[294, 889]\times10^{3}$	1×10^{-3}	$2.15\,[0.27, 3.52]\times10^{3}$
15. PT	18/18	$218\,[172, 268]\times10^{3}$	1×10^{-3}	$199\,[117, 450]$
16. SMC	18/18	$163\,[140, 238]\times10^{3}$	1×10^{-3}	5×10^{3}
17. PMC	0/18	2×10^{3}	1×10^{-3}	$2.44\,[1.96, 2.91]$
Neural samplers				
18. Boltzmann generator	2/18	20×10^{6}	$1.74\,[1.61, 1.79]\times10^{3}$	$194\,[179, 208]\times10^{-9}$
19. FAB-style	0/18	20×10^{6}	$1.5\,[1.45, 1.59]\times10^{3}$	$277\,[209, 410]\times10^{-9}$
20. iDEM	0/18	50×10^{6}	$120\,[86.7, 204]$	$14.8\,[9.53, 21.3]\times10^{-6}$
21. GFlowNet-TB	0/18	5×10^{6}	$2.75\,[2.66, 2.89]\times10^{3}$	$11.8\,[11, 13.7]\times10^{-6}$
22. PIS	0/18	5×10^{6}	$2.65\,[2.6, 2.75]\times10^{3}$	$6.43\,[4.15, 9.05]\times10^{-6}$

D ALANINE DIPEPTIDE IN RAMACHANDRAN COORDINATES

This experiment uses a real oracle in two periodic coordinates, with a small secondary basin that tests mode discovery. The GP and classical methods query the real oracle; the neural samplers train on an emulator (Appendix D.4).

D.1 TARGET AND ORACLE

We sample the backbone torsions $(\phi, \psi) \in [-180^\circ, 180^\circ)^2$ of Ace–Ala–NMe (22 atoms). A query sets the torsions in a fixed reference conformer (built with RDKit with trans amides and relaxed with GFN2-xTB) and relaxes the remaining coordinates with GFN2-xTB (Bannwarth et al., 2019) through `tblite` and ASE's constrained BFGS optimizer (force tolerance 0.02 eV/Å, at most 300 steps). The target is

$$p(\phi, \psi) \propto \exp\left[-\frac{E_{\mathrm{rel}}(\phi, \psi) - E_0}{k_B T}\right], \qquad T = 300 \text{ K}, \tag{D.1}$$

a gas-phase relaxed-energy surface rather than a torsional free energy. The reference is the same energy at the centers of a 72×72 grid (5° cells; all $5{,}184$ relaxations converged). One query is one requested torsion pair; runs use seeds 0–2 and up to $1{,}000$ queries.

D.2 PROTOCOL

The GP input is $(\cos\phi, \sin\phi, \cos\psi, \sin\psi)$ mapped to $[0, 1]^4$, acquisitions are optimized in $(\phi, \psi)/45^\circ \in [-4, 4]^2$, training energies are clipped at the best observed value plus $50\, k_B T$, and Gibbs queries and particles are drawn on the 72×72 grid. RWMH uses ten chains with an initial proposal s.d. of 0.8 in scaled torsions; PT uses ten chains at each of four log-spaced temperatures from 1 to 10 with proposal scale $\propto \sqrt{T}$; both wrap proposals onto the torus, adapt toward acceptance 0.234 during the first 10% of the budget and discard the first quarter of each chain. SMC uses covariance-scaled Metropolis moves, with each plotted budget an independent trial ($10{,}760$ queries per seed for the whole curve). PMC weights its wrapped proposals with unwrapped Gaussian densities, so only "PMC + GP" is reported. VBMC uses $1{,}000$ evaluations and does not encode periodicity. MALA and HMC are omitted because the oracle provides no torsion derivatives.

D.3 COMPLETE RESULTS

Densities $q \propto \exp(-m^E)$ are normalized on the reference grid for D_J; TV is computed on the grid and on its 36×36 aggregation. Figure 12 and Table 7 give the complete real-oracle comparison.

Mode discovery. In seed 1, all three Gibbs policies underweight the positive-ϕ basin through the final budget. In the rectangle $30^\circ < \phi < 120^\circ$, $-120^\circ < \psi < 0^\circ$ (reference mass 7.43%), GP-PM, GP-UCS and GP-TS^2 place 0.0367%, 0.0318% and 0.0567% of their mass at $1{,}000$ calls, and OneStep-L1 7.40% (Figure 13): exact sampling of the surrogate does not ensure that every mode is discovered.

Unstable fits. Some refits produce abrupt error spikes: at the final budget, PT + GP puts 77% and 81% of its mass in one grid cell in seeds 0 and 1 ($D_J \simeq 50$ and 26), and the uniform and RWMH designs show large transient errors. We retain these runs. VBMC never reaches $D_J < 0.1$.

D.4 NEURAL SAMPLERS ON AN EMULATOR

The neural runs need 10^7–10^8 evaluations and torsion derivatives, so they train on a periodic bicubic interpolant of a 144×144 relaxed-energy scan ($20{,}736$ real queries; RMSE $0.075\, k_B T$ on 300 fresh queries; coarse TV 0.0052 from the reference). Training budgets count emulator evaluations only. All methods use scaled torsions with a quadratic wall $100 \sum_j (|x_j| - 4)_+^2$ outside one period; the Boltzmann generator anneals the inverse temperature from 0.2 to 1 over the first 30% of training, and the FAB-style flow uses eight annealing levels with five leapfrog steps and no replay. All five methods are scored by wrapped histograms of $200{,}000$ samples on the 36×36 grid, and each curve

Table 7: Alanine Jeffreys divergence, reported as median [interquartile range] over three seeds. The count $n_{0.1}$ is the median first budget with $D_J < 0.1$; unsuccessful runs are censored beyond $1{,}000$ calls. "Reached counts how many seeds crossed at least once. Classical + GP rows evaluate the fitted surrogate.

Method	D_J at 100	D_J at 1,000	$n_{0.1}$	Reached
GP-PM	0.614 [0.33, 5.3]	0.00592 [0.0053, 0.29]	240	2/3
GP-UCS	0.498 [0.26, 5]	0.00343 [0.0033, 0.3]	180	2/3
GP-TS^2	0.0322 [0.026, 0.29]	0.00362 [0.0036, 0.27]	70	2/3
OneStep-L1	0.00817 [0.0068, 0.0085]	0.00147 [0.0015, 0.0016]	60	3/3
OneStep-L2	0.00661 [0.0065, 0.0078]	0.00195 [0.0018, 0.002]	60	3/3
OneStep-L3	0.0057 [0.0054, 0.008]	0.00151 [0.0015, 0.0016]	50	3/3
GPry-LogExp	0.0114 [0.011, 0.014]	0.00236 [0.0022, 0.0025]	70	3/3
LogExp pool	0.0105 [0.0096, 0.023]	0.0024 [0.0024, 0.0025]	70	3/3
BAPE-EV	0.0352 [0.029, 0.038]	0.00959 [0.0067, 0.0096]	70	3/3
Uniform + GP	0.186 [0.13, 0.21]	0.0787 [0.044, 1.2]	110	3/3
RWMH + GP	0.0994 [0.099, 0.28]	0.00246 [0.0023, 0.042]	100	3/3
PT + GP	0.48 [0.34, 0.97]	26.2 [13, 38]	170	3/3
SMC + GP	0.227 [0.2, 0.39]	0.0249 [0.014, 0.034]	200	3/3
PMC + GP	1.16 [1, 4.3]	0.00333 [0.0032, 0.0066]	320	3/3
VBMC	3.47 [3, 5.5]	0.738 [0.72, 1.3]	$> 1{,}000$	0/3

Table 8: Alanine neural results at the last budget available for all three seeds of each method. Coarse TV reports the median [interquartile range] from wrapped histograms of $200{,}000$ samples. Counts are actual emulator evaluations, not real xTB queries. The $20{,}736$ real queries used to construct the emulator and 300 validation queries are excluded.

Method	Evaluations	Coarse TV
Boltzmann generator	$100,000,000$	0.205 [0.123, 0.266]
FAB-style	$100,013,568$	0.0992 [0.0932, 0.106]
iDEM	$100,000,000$	0.577 [0.574, 0.577]
GFlowNet-TB	$10,000,128$	0.273 [0.271, 0.273]
PIS	$20,000,000$	0.269 [0.265, 0.597]

stops at the last budget shared by all three seeds (10^7 evaluations for GFlowNet-TB, 2×10^7 for PIS, about 10^8 otherwise). Table 8 and Figure 14 give the results: PIS ends at TV 0.925 in one seed (median 0.269), and the FAB-style flow reaches a median TV of 0.099.

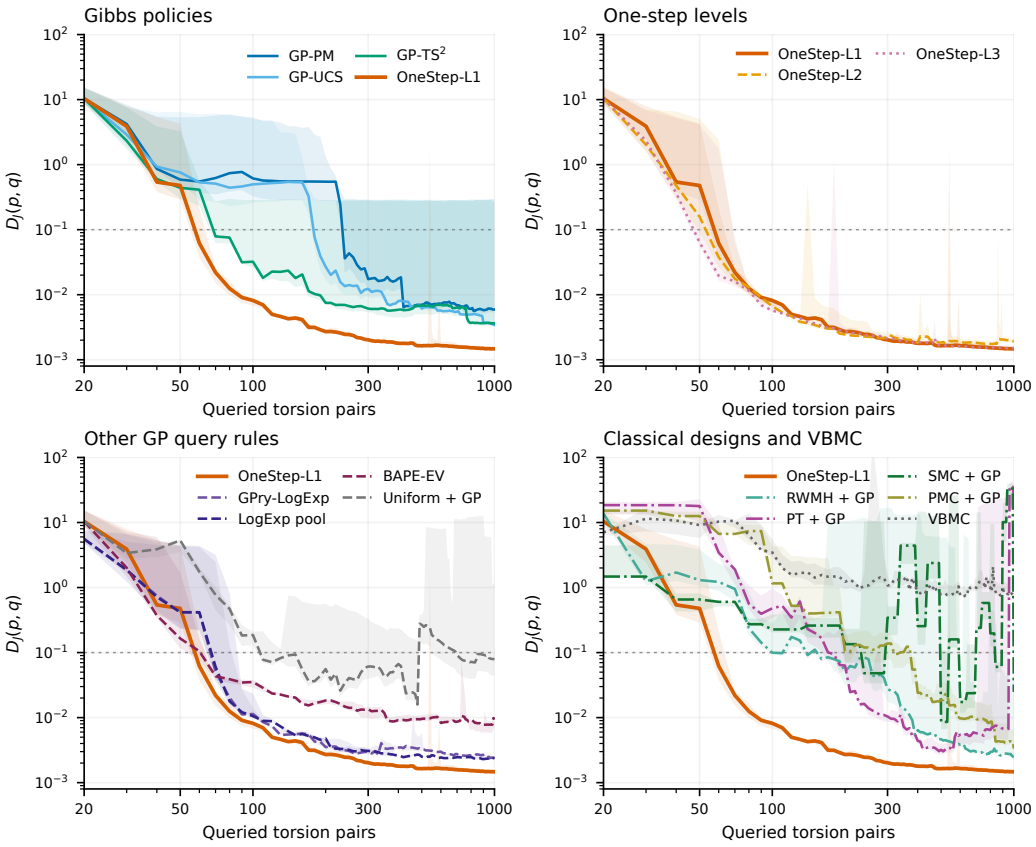

Figure 12: Complete alanine comparison: median Jeffreys divergence and interquartile range over three seeds. OneStep-L1 is repeated in each panel for comparison; the dashed line marks $D_J = 0.1$. Each SMC budget is an independent trial, and PMC + GP scores the query design only.

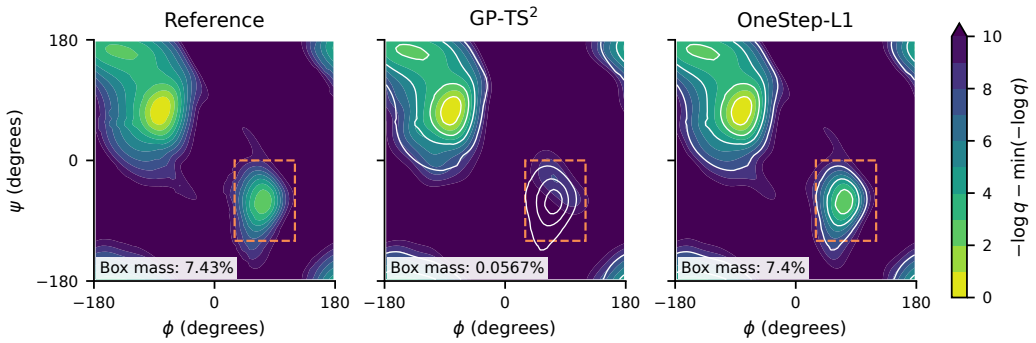

Figure 13: Minor-basin diagnostic for seed 1 at 1,000 calls. Panels show $-\log q - \min(-\log q)$, capped at 10; white contours mark reference levels 1, 3, 5 and 8. The dashed rectangle is $30° < \phi < 120°$, $-120° < \psi < 0°$, with its mass in each panel.

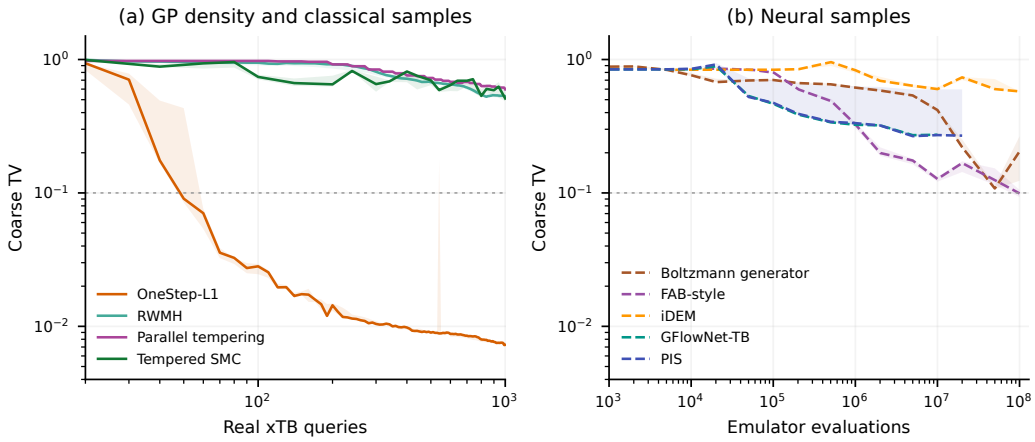

Figure 14: Alanine total variation on a 36×36 torsion grid, median and interquartile range over three seeds. Left: the OneStep-L1 density and raw classical samples on the real oracle. Right: the five neural implementations on the emulator, scored with wrapped histograms of $200{,}000$ samples; emulator-call counts exclude the $20{,}736$-query construction scan. The dashed line marks TV 0.1.

E DOUBLEWELL-4

DW-4 is the largest experiment (eight coordinates, 20,000 calls). The surrogate is sampled by Markov chains and compared with tuned classical samplers at the same budget and with neural samplers at their full training budgets.

E.1 TARGET, REFERENCE AND SCORING

For four planar particles with pair distances $r_{ij} = \|x_i - x_j\|$,

$$E(x) = \sum_{i<j} \left[-4(r_{ij} - 4)^2 + 0.9(r_{ij} - 4)^4\right], \qquad p(x) \propto e^{-E(x)}.$$

The GP acts on the eight coordinates after subtracting the center of mass. GP methods use 50 initial states and batches of 50 up to 20,000 calls, with seeds 0–5. The reference is 10^4 terminal states of long Metropolis chains. A fitted GP is scored by running 10^4 RWMH chains of 20,000 steps (proposal s.d. 0.1) on its posterior mean from $\mathcal{N}(0, 1.5^2 I)$ starts and comparing the pair-distance and energy histograms of the final states with the reference (200 equal-width bins over the reference range; out-of-range mass counts as error). These marginal scores, $\mathrm{TV}_{\mathrm{dist}}$ and $\mathrm{TV}_{\mathcal{E}}$, do not measure full-state TV.

E.2 SETTINGS SPECIFIC TO DW-4

GP inputs are rescaled to the current range of the training points, query and particle chains use a fixed isotropic proposal, acquisitions are optimized within the particle range extended by 3 on each side, and GPry-LogExp uses $\zeta = 8^{-0.85} \approx 0.171$ (Table 1).

E.3 CLASSICAL MCMC AND MCMC + GP

RWMH, MALA, HMC and PT receive 2×10^4 evaluations, including each chain's initial $\mathcal{N}(0, 1.5^2 I)$ draw; step sizes start at a multiple of an equilibrium-tuned value and adapt during burn-in; each chain contributes 1,000 evenly spaced post-burn-in states; forces are clipped at norm 10^3. On seeds 0–2 we tune chain counts $\{10, 50, 100\}$, burn-in fractions $\{0.25, 0.5, 0.75\}$, step multipliers $\{0.25, 0.5, 1, 2\}$ (0.1 also for MALA) and HMC leapfrog steps $\{1, 2, 3, 5, 10\}$, and for PT 4 or 8 temperatures up to 3 or 10, step multipliers $\{0.5, 1, 2\}$, 10 or 50 chains and burn-in 0.25 or 0.5. Among settings with acceptance within a factor 1.6 of its target and energy $W_2 < 1$, the lowest median $\mathrm{TV}_{\mathrm{dist}} + \mathrm{TV}_{\mathcal{E}}$ is run on seeds 0–4 (Table 9). MCMC + GP fits the GP once to all 2×10^4 evaluated states and is scored like our methods.

Table 9: Tuned classical MCMC on DW-4, 2×10^4 energy evaluations per run.

Sampler	Chains	Energy calls per step	Steps	Burn-in	Initial step multiple	Target acceptance	Other
RWMH	10	1	1,999	25%	2	0.234	–
MALA	10	1	1,999	50%	0.25	0.574	–
HMC	10	2	999	25%	0.25	0.65	$L = 2$
PT	10	4	499	25%	1	0.234	4 temperatures up to 3

E.4 NEURAL SAMPLERS

FAB and iDEM use their released configurations, PIS the iDEM code base and DDS a PyTorch reimplementation, with five seeds each (Table 10). Budgets count full-schedule training evaluations, excluding initial-buffer population and diagnostics. For each seed we report the checkpoint with the lowest $\mathrm{TV}_{\mathrm{dist}}$, chosen with the reference scores themselves.

Table 10: Neural-sampler runs on DW-4. FAB and iDEM use their released configurations, PIS uses the iDEM code base, and DDS a PyTorch reimplementation.

	FAB	PIS	DDS	iDEM
Model	SE(3)-augmented coupling flow, 12 layers, EGNN conditioner	EGNN drift (128 units, 3 layers) plus a time-weighted $\nabla \log p$ term	EGNN drift (128 units, 3 layers)	EGNN score network (128 units, 3 layers)
Base	Gaussian, center of mass removed	SDE started at $x = 0$	$\mathcal{N}(0, 2^2 I)$, center of mass removed	$\mathcal{N}(0, 3^2 I)$, center of mass removed
Objective	AIS toward p^2/q, prioritized replay buffer	KL between path measures	reverse KL between path measures, sticking the landing	regression on Monte Carlo score estimates, $K = 1{,}000$
Integration	AIS: 2 intermediate distributions, HMC with 5 leapfrog steps	Euler–Maruyama, 100 steps	Euler, 100 steps, OU reference	VE SDE, σ from 10^{-5} to 3, 1,000 steps
Training	20,000 iterations, 128 AIS samples each	1,000 epochs of 100 steps, 512 paths	100,000 iterations, 512 paths	1,000 epochs of 100 steps, 512 points
Optimizer	Adam, lr 2×10^{-4} with warm-up and decay	Adam, lr 5×10^{-4}	Adam, lr 10^{-4}	Adam, lr 10^{-3}
Gradient clipping	adaptive	0.5	norm 0.5	0.5
Evaluations per iteration	1,408	5.2×10^4	512	5.1×10^5
Total evaluations	2.8×10^7	5.3×10^9	5.1×10^7	5.1×10^{10}
Checkpoints scored	14	13	15	6
Scored samples	10^4 flow samples, not reweighted	10^4, same SDE	10^4, same SDE	10^4, same SDE
Seeds	0–4	0–4	0–4	0–4

E.5 Complete results

Table 11 gives all methods; bold marks the lowest median at 2×10^4 calls. The budget-adapted neural rows come from a 20,000-call configuration screen and do not establish a minimum training budget.

Table 11: DW-4 marginal TV, median [interquartile range]. Acquisition policies use six seeds; classical methods, their GP fits and full-schedule neural methods use five; budget-adapted neural methods use three. Neural results use the best scored checkpoints. Bold marks the lowest median at 2×10^4 calls.

Method	Evaluations	$\mathrm{TV}_{\mathrm{dist}} \downarrow$	$\mathrm{TV}_{\mathcal{E}} \downarrow$
Neural samplers, full training schedule			
FAB	2.8×10^7	0.034 [0.033, 0.035]	0.080 [0.077, 0.080]
PIS	5.3×10^9	0.053 [0.052, 0.055]	0.132 [0.127, 0.133]
DDS	5.1×10^7	0.246 [0.246, 0.255]	0.357 [0.313, 0.441]
iDEM	5.1×10^{10}	0.058 [0.041, 0.078]	0.195 [0.086, 0.202]
Neural samplers, 2×10^4 evaluations			
FAB	2×10^4	0.550 [0.549, 0.550]	0.940 [0.940, 0.941]
PIS	2×10^4	0.408 [0.399, 0.414]	0.800 [0.790, 0.807]
DDS	2×10^4	0.338 [0.338, 0.364]	0.784 [0.782, 0.811]
iDEM	2×10^4	0.287 [0.217, 0.317]	0.211 [0.190, 0.258]
Classical MCMC			
RWMH	2×10^4	0.056 [0.055, 0.074]	0.110 [0.105, 0.117]
MALA	2×10^4	0.059 [0.051, 0.127]	0.102 [0.081, 0.107]
HMC	2×10^4	**0.051** [0.045, 0.067]	0.093 [0.089, 0.105]
PT	2×10^4	0.077 [0.073, 0.080]	0.164 [0.162, 0.174]
MCMC + GP			
RWMH + GP	2×10^4	0.144 [0.141, 0.144]	0.195 [0.164, 0.215]
MALA + GP	2×10^4	0.198 [0.175, 0.207]	0.192 [0.157, 0.219]
HMC + GP	2×10^4	0.147 [0.140, 0.180]	0.168 [0.141, 0.172]
PT + GP	2×10^4	0.309 [0.300, 0.372]	0.636 [0.612, 0.665]
GP baselines			
GPry-LogExp	2×10^4	0.110 [0.071, 0.219]	0.217 [0.103, 0.529]
BAPE-EV	2×10^4	0.771 [0.688, 0.834]	0.998 [0.997, 1.000]
Ours			
GP-PM	2×10^4	0.107 [0.087, 0.134]	0.094 [0.091, 0.108]
GP-UCS	2×10^4	0.060 [0.058, 0.063]	**0.091** [0.085, 0.098]
GP-TS^2	2×10^4	0.152 [0.146, 0.158]	0.214 [0.181, 0.300]
OneStep-L1	2×10^4	0.108 [0.099, 0.125]	0.167 [0.102, 0.210]
OneStep-L2	2×10^4	0.149 [0.141, 0.196]	0.280 [0.140, 0.403]
OneStep-L3	2×10^4	0.192 [0.165, 0.220]	0.409 [0.259, 0.500]

F PLANCK-LITE ΛCDM

This experiment learns a six-parameter cosmological posterior within 600 calls; accuracy is estimated against a reference bank.

F.1 POSTERIOR AND COORDINATES

The energy is the negative log posterior of flat ΛCDM, computed with CLASS (Blas et al., 2011) and Cobaya (Torrado & Lewis, 2021) from the Planck 2018 low-ℓ TT and EE, plik-lite TTTEEE and lensing likelihoods (Planck Collaboration, 2020b;a), with $A_{\rm Planck} = 1$ and neutrino mass $0.06\,\mathrm{eV}$. Parameters are mapped to $z_i = (\theta_i - \ell_i)/s_i$ with the centers and scales of Table 12, and all methods operate on $[-8, 8]^6$. Each run uses 18 initial draws from the Gaussian reference distribution in the table, then batches of six up to 600 calls; non-finite oracle values are clipped, and proposals outside the box are rejected.

Table 12: Planck-lite priors, coordinate centers and scales, and the standard deviations of the Gaussian reference draws used for initialization.

Parameter	Uniform prior	Center ℓ_i	Scale s_i	Initial s.d.
$\log(10^{10}A_s)$	$[1.61, 3.91]$	3.05	0.014	0.001
n_s	$[0.8, 1.2]$	0.965	0.0042	0.004
$100\theta_s$	$[0.5, 10]$	1.0416	0.0003	0.0004
ω_b	$[0.005, 0.1]$	0.0224	0.00015	0.0001
$\omega_{\rm cdm}$	$[0.001, 0.99]$	0.120	0.0012	0.001
$\tau_{\rm reio}$	$[0.01, 0.8]$	0.055	0.0073	0.006

F.2 SETTINGS SPECIFIC TO PLANCK-LITE

Because the initial cloud is narrow (especially in $\log A_s$), inputs are scaled by the fixed box rather than by the data, the prior-mean curvature is initialized at 40, and energies are clipped at the best observed value plus 300. Gibbs queries and particles use 5,000 RWMH steps with a proposal covariance estimated from the low-energy training points (shrunk 5% toward its diagonal) and a scale adapted toward 23.4% acceptance during the first 2,500 steps. GPry-LogExp uses $\zeta = 6^{-0.85} \approx 0.218$.

F.3 REFERENCE BANK AND DIVERGENCE ESTIMATOR

The reference bank holds 258,948 rows from 20 chains, with multiplicities totaling 704,376 states (parameter-wise effective sample sizes 2.9–3.2 $\times 10^4$). With multiplicities a_i, $b_i = a_i / \sum_j a_j$ and $r_i = E(x_i) - m^E(x_i)$, let

$$L = \log \sum_i b_i e^{r_i}, \qquad u_i = \frac{b_i e^{r_i}}{\sum_j b_j e^{r_j}};$$

then

$$\widehat{\mathrm{KL}}(p\|q) = L - \sum_i b_i r_i, \qquad \widehat{\mathrm{KL}}(q\|p) = \sum_i u_i r_i - L.$$

Table 13 reports their sum D_J as mean $\pm$ standard error over five seeds; scoring uses no oracle calls. Both estimates require the bank to cover the learned density: mass placed outside the bank goes undetected, so a low divergence does not certify the tails, and the standard errors exclude bank uncertainty.

Table 13: Planck-lite Jeffreys divergence from reference-bank reweighting, mean ± standard error over five seeds. Bold marks the lowest mean. GPry-LogExp is the optimized LogExp acquisition in the shared GP. Settings and bank-overlap limitations are given in Appendix F.

	Oracle calls		
Acquisition	150	300	600
GP-PM	37 ± 4	20 ± 3	8 ± 2
GP-UCS	28 ± 2	16 ± 4	6 ± 1
GP-TS^2	12 ± 6	6 ± 3	2 ± 1
OneStep-L1	2 ± 1	0.13 ± 0.06	0.025 ± 0.004
OneStep-L2	$\mathbf{0.9} \pm 0.3$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.022} \pm 0.003$
OneStep-L3	1.4 ± 0.8	0.08 ± 0.01	$\mathbf{0.022} \pm 0.002$
GPry-LogExp	7 ± 3	0.31 ± 0.09	0.027 ± 0.006
BAPE-EV	68 ± 5	67 ± 8	58 ± 7

G Extended related work

Monte Carlo methods. MCMC constructs a chain whose stationary distribution is the target, as in Metropolis–Hastings (Metropolis et al., 1953; Hastings, 1970), Hamiltonian Monte Carlo (Duane et al., 1987), and the No-U-Turn Sampler (Hoffman & Gelman, 2014). Parallel tempering exchanges states between chains at different temperatures to improve mixing (Swendsen & Wang, 1986; Geyer, 1991). Sequential Monte Carlo (SMC) propagates weighted particles through intermediate targets using reweighting, resampling, and often MCMC moves (Del Moral et al., 2006); annealed importance sampling instead forms weights along annealed trajectories without population resampling (Neal, 2001). Population Monte Carlo and AMIS adapt proposals from previous weighted draws (Cappé et al., 2004; Cornuet et al., 2012). Nested sampling estimates evidence and produces weighted posterior samples by exploring regions of increasing likelihood (Skilling, 2006); nautilus improves its efficiency through learned proposals (Lange, 2023). These methods reuse evaluations or adapt proposals, but further sampling or importance correction generally requires new target evaluations. An energy surrogate supports approximate sampling without those additional evaluations.

Variational and neural samplers. Variational inference fits a tractable distribution by optimizing a divergence or an evidence bound (Blei et al., 2017), while Stein variational gradient descent transports particles using the target score and particle interactions (Liu & Wang, 2016). Boltzmann generators learn invertible maps from a simple distribution to the target (Noé et al., 2019), and Flow Annealed Importance Sampling Bootstrap (FAB) trains normalizing flows using annealed importance sampling (Midgley et al., 2023). Path Integral Sampler and Denoising Diffusion Sampler learn controlled stochastic dynamics toward the target (Zhang & Chen, 2022; Vargas et al., 2023). Iterated Denoising Energy Matching (iDEM) learns a score model from Monte Carlo estimates using energy evaluations, without an initial dataset of target samples (Akhound-Sadegh et al., 2024). GFlowNets learn policies whose terminal outcomes have probabilities proportional to a reward (Bengio et al., 2021; 2023), including in continuous and hybrid spaces (Lahlou et al., 2023) and for diffusion sampling (Sendera et al., 2024). These methods learn a sampling distribution or transport, whereas our GP models the energy that defines the distribution.

Reducing oracle evaluations. Adjoint Sampling reuses energy and score evaluations through a replay buffer (Havens et al., 2025); Value Gradient Sampler uses temporal-difference learning with off-policy training (Hwang et al., 2026). Sequential Controlled Langevin Diffusions combines learned dynamics with SMC (Chen et al., 2025), NETS learns transport that reduces importance-weight variance (Albergo & Vanden-Eijnden, 2025), and DiKL uses a reverse diffusive KL objective (He et al., 2025). GFlowNet-based active learning combines surrogate uncertainty with neural sampling to propose diverse candidates for expensive biological sequence evaluations (Jain et al., 2022). We focus on small evaluation budgets and queries that improve a specified target distribution, accounting separately for the computation needed to fit and use the surrogate.

Gaussian processes for expensive functions. GPs represent an unknown function and uncertainty about its values (Rasmussen & Williams, 2006). Bayesian optimization uses this information to acquire evaluations of an expensive objective (Jones et al., 1998; Frazier, 2018); GP-UCB and GP Thompson sampling have regret guarantees governed by information gain (Srinivas et al., 2010; Chowdhury & Gopalan, 2017). Model-based RL similarly reduces environment interactions: PILCO learns GP dynamics (Deisenroth & Rasmussen, 2011), while PETS uses probabilistic neural dynamics models (Chua et al., 2018). Multifidelity methods exploit cheaper oracle approximations when available (Peherstorfer et al., 2018). We share the objective of learning from few evaluations, but seek a distribution rather than a maximizer or control policy.

Active posterior estimation and Bayesian quadrature. The closest methods actively learn expensive likelihoods or log densities. Bayesian Active Posterior Estimation (BAPE) proposes an exponentiated-variance acquisition and a one-step expected-divergence criterion (Kandasamy et al., 2015). GPry balances predicted posterior mass and uncertainty in a GP of the log posterior (El Gammal et al., 2023), with subsequent work on parallel acquisition (Torrado et al., 2023). Active GP surrogates have also been applied to inverse problems with expensive likelihoods (Dinkel et al., 2024).

Bayesian quadrature selects evaluations to reduce uncertainty in an integral. WSABI uses a transformed GP for nonnegative integrands (Gunter et al., 2014); Variational Bayesian Monte Carlo (VBMC) combines a GP surrogate, variational posterior, and evidence estimate, including with noisy likelihoods (Acerbi, 2018; 2020). Sequential design also targets uncertainty in posterior approximations (Sinsbeck & Nowak, 2017; Järvenpää et al., 2021). In likelihood-free inference, BOLFI and related acquisitions model simulator discrepancies to learn an approximate Bayesian computation posterior (Gutmann & Corander, 2016; Järvenpää et al., 2019). Our OneStep criterion minimizes expected reverse KL from the updated posterior-mean Gibbs measure to the unknown target. It differs from BAPE's forward-KL instantiation, integrated-variance criteria, and evidence-estimation objectives. We derive tractable approximations to this acquisition and separately analyze the regret of GP sampling policies.

Surrogate-assisted Monte Carlo and approximation guarantees. GP-based HMC constructs proposals from an approximate log density (Rasmussen, 2003), while delayed-acceptance MCMC screens proposals with a cheap approximation before an exact correction (Christen & Fox, 2005). Locally refined surrogate MCMC can be asymptotically exact under suitable refinement conditions (Conrad et al., 2016). GP-emulated MCMC selects evaluations to control uncertainty in acceptance decisions (Järvenpää & Corander, 2024). Other guarantees relate GP surrogate error to the Hellinger distance between exact and approximate posteriors (Stuart & Teckentrup, 2018). Our analysis connects sequential query policies to cumulative sampling regret and the distribution learned within an evaluation budget.

KL-regularized learning and sampling regret. For a reference distribution μ, the connection to RL follows from

$$J_\eta(\pi) = \mathbb{E}_{x\sim\pi}[f(x)] - \eta^{-1} D_{\mathrm{KL}}(\pi\|\mu),$$

whose optimizer is the Gibbs measure $p_\eta(dx) \propto e^{\eta f(x)}\mu(dx)$. Its suboptimality gap is $J_\eta(p_\eta) - J_\eta(\pi) = \eta^{-1} D_{\mathrm{KL}}(\pi\|p_\eta)$. This connects sampling to KL-regularized decision making and distribution matching for language-model fine-tuning (Korbak et al., 2022; Go et al., 2023). Pedramfar & Ravanbakhsh (2026) study simple and cumulative sampling regret for finite arms; Zhao et al. (2025) establish logarithmic regret for KL-regularized online RL under their assumptions. At fixed positive temperature, sampling from an estimated Gibbs distribution can itself provide sufficient exploration, but guarantees can deteriorate as temperature decreases. Ji et al. (2026) give near-optimal cumulative-regret guarantees across temperature regimes for finite-armed KL-regularized bandits. We study GP sampling over a structured domain: GP-PM, GP-UCS, and GP-TS^2 address cumulative regret, while OneStep targets the next learned distribution and permits exploratory queries that need not follow the target.