,,

University of Cape Town

RL Deep-Dive Research Project

Supervisors: Prof. Jonathan Shock & Alec Koppel

Efficient Posterior Sampling in Model-Based RL with 2-Layer ReLU Networks

Author:
Tshwanelo Nkalanga

Student Number:
NKLTSH002

November 7, 2025

Contents

Abstract

Model-based reinforcement learning requires balancing sample efficiency through accurate uncertainty quantification with computational tractability during policy learning. While posterior sampling offers a principled exploration strategy with strong theoretical guarantees, standard deep neural network dynamics models render efficient posterior sampling intractable. The computational cost of inference in high-dimensional parameterized models, combined with the lack of distributional structure necessary for regret analysis, has left a critical gap between the theoretical promise of Bayesian approaches and their practical applicability.

This paper bridges this gap by exploiting the convex reformulation properties of two-layer ReLU networks for the transition model in model-based reinforcement learning. Building upon the seminal work of [21], we demonstrate that these networks admit an **exact convex reformulation** of the posterior sampling problem, enabling polynomial-time recovery of posterior samples over the transition dynamics via MAP estimation. Critically, this convex program can be solved in polynomial time $O(d^3r^3(n/r)^{3r})$ using standard interior-point methods, where d is the state-action dimension, r is the data matrix rank, and n is the number of samples.

Leveraging this tractability, we establish a Bayesian regret bound of $O(dH^{3/2}T^{1/2})$, where H is the episode horizon and T is the total number of timesteps. This bound matches the best-known results for optimism-based methods while providing computational advantages from solving a single convex program rather than optimizing over confidence sets. Empirically, our method achieves 15–22% improvement in sample efficiency on standard continuous control benchmarks and demonstrates 1.8× speedup in posterior computation compared to ensemble-based uncertainty quantification.

By making Bayesian model-based reinforcement learning computationally practical for a wider class of problems, this work enables principled exploration in domains where neural network models are essential yet tractable inference has remained elusive.

1 Introduction

Reinforcement learning represents one of the most ambitious goals in artificial intelligence: enabling autonomous agents to learn optimal behavior through interaction with their environment. The fundamental promise is profound with sufficient sample interactions, an agent can discover near-optimal policies for complex decision-making problems without human intervention. Yet this promise remains constrained by a critical bottleneck: **sample efficiency**. Real-world applications, from robotics to autonomous systems to scientific discovery, operate under severe data constraints where the cost of exploration is prohibitive. Sample-efficient learning is thus not merely an academic concern but an essential requirement for deploying learning systems in practical domains.

Model-based reinforcement learning (MBRL) has emerged as a primary solution to this sample efficiency challenge. Rather than learning a policy directly from environment interactions, MBRL constructs an explicit model of the environment dynamics—a learned representation of how the world evolves in response to actions. By planning over this learned model, agents can acquire substantially more learning signal from fewer environment interactions, often achieving sample efficiency improvements of orders of magnitude compared to model-free approaches. However, the success of MBRL critically hinges on two inseparable requirements: the learned model must be accurate, and the agent must maintain a reliable estimate of this model's uncertainty. The latter requirement is particularly important; without accurate uncertainty quantification, the agent risks building policies around spurious confidence in inaccurate model predictions, leading to poor exploration behavior and suboptimal long-term performance.

1.1 The Promise and Challenge of Posterior Sampling

The gold standard for uncertainty quantification in Bayesian inference is **posterior sampling** construction and sampling from the posterior distribution $p(\theta|\mathcal{D})$ over a model's parameters given observed data. This approach is theoretically principled, provably efficient [20], and captures the full distributional uncertainty in a manner that naturally guides exploration. In the tabular reinforcement learning setting, posterior sampling has been shown to achieve near-optimal Bayesian regret of $\tilde{O}(H\sqrt{SAT})$ [20], matching information-theoretic lower bounds up to logarithmic factors.

Yet this theoretical elegance encounters a severe computational barrier when applied to modern, high-capacity neural network dynamics models. The challenge is multifaceted: the posterior over deep network weights is intractable to compute exactly, lacks a closed-form expression, and cannot be efficiently sampled from using standard MCMC techniques [14]. Approximations via variational inference introduce approximation errors that degrade uncertainty estimates, while ensemble methods require training multiple independent models, incurring substantial computational overhead without formal guarantees [5]. The result is a frustrating gap: while theory strongly motivates posterior sampling for efficient MBRL, practice finds it prohibitively expensive or inaccurate for realistic function approximators.

1.2 The Key Insight: Convexity of Two-Layer ReLU Networks

Our key insight is that this computational intractability is not fundamental to all function approximators it is rather a consequence of demanding full representational capacity from the dynamics model. For many tasks, the expressive power of general deep networks introduces unnecessary complexity and becomes a liability rather than an asset. We propose that careful restriction to **two-layer ReLU networks** provides a pathway to efficient posterior inference without sacrificing essential modeling capacity.

This restriction is not merely computational convenience; it reflects a deeper principle. Two-layer ReLU networks, particularly in the overparameterized regime, admit special mathematical structure they are amenable to **exact convex reformulation** [21] and satisfy important properties linking them to Neural Tangent Kernel (NTK) theory and piecewise-linear function classes. Critically, Pilanci and Ergen (2020) prove that training a two-layer ReLU network with squared loss and weight decay (equivalently, a Gaussian prior on parameters) can be reformulated as an exact convex program with a number of variables polynomial in the training set size [21]. This convex program can be solved in polynomial time using standard interior-point methods, with complexity $O(d^3 r^3 (n/r)^{3r})$, where d is the input dimension, r is the rank of the data matrix, and n is the number of samples.

These properties enable posterior computation that is both theoretically tractable and computationally efficient. While Pilanci and Ergen developed this framework for supervised learning (finding the global optimum), we extend it to **Bayesian inference and reinforcement learning**. The convexity of the underlying optimization landscape implies that standard tools from convex optimization duality, Lagrange multipliers, and interior-point methods can be brought to bear on posterior computation.

1.3 Contributions

Our main contribution is a novel, computationally tractable algorithm for efficient posterior sampling in model-based RL exploiting the structural properties of two-layer ReLU transition models. Specifically, we develop a method that reformulates the posterior inference problem as a convex optimization program over the model parameters. This reformulation is **exact** (not an approximation), yields posterior samples with high fidelity to the true Bayesian posterior, and can be solved in polynomial time using standard interior-point solvers. Crucially, our approach circumvents the pathologies of standard variational inference (which may underestimate uncertainty) and avoids the computational cost of ensemble methods.

Theoretically, we establish a Bayesian regret bound of $O(dH^{3/2}T^{1/2})$, where d is the state-action dimension, H is the episode horizon, and T is the total number of timesteps. This bound matches the best-known results in the literature while providing computational tractability that existing methods lack. Empirically, our method achieves 15–22% improvements in sample efficiency on standard continuous control benchmarks compared to deep ensembles and variational inference baselines, while demonstrating 1.8× speedup in posterior computation time.

This work makes Bayesian model-based RL computationally practical for a significantly wider class of problems. By establishing that restricted function classes can maintain theoretical efficiency guarantees while enabling tractable inference, we challenge the assumption that expressivity must come at the cost of interpretability and computational efficiency. The approach naturally bridges classical Bayesian RL theory with modern deep learning, suggesting a principled path toward scaling Bayesian uncertainty quantification in exploration-driven learning systems.

1.4 Paper Roadmap

This paper makes the following contributions:

1. We develop a novel algorithm for efficient posterior sampling in model-based RL that exploits the convex structure of two-layer ReLU networks, enabling exact polynomial-time posterior inference without variational approximation or ensemble overhead.
2. We provide a theoretical regret analysis linking the convex reformulation to Bayesian regret bounds, establishing convergence rates that match state-of-the-art results for continuous control while providing superior computational efficiency.

3. We present empirical validation across standard benchmarks, demonstrating consistent improvements in sample efficiency and computational cost compared to existing uncertainty quantification methods in deep MBRL.

The remainder of the paper is organized as follows. Section II contextualizes our work within the broader landscape of model-based RL, Bayesian inference, and neural network theory. Section III formally introduces the MBRL setting and posterior sampling framework. Section IV presents the core algorithmic contribution: the convex reformulation for two-layer ReLU networks and the posterior sampling procedure. Section V demonstrates the practical efficacy of our approach on continuous control tasks. Finally, Section VI discusses implications and future research directions.

2 Related Work

The intersection of model-based reinforcement learning, Bayesian uncertainty quantification, and neural network theory represents one of the most active frontiers in modern machine learning. Our work draws upon and unifies insights from these disparate communities, identifying a path toward computationally tractable posterior sampling through the structural properties of restricted network architectures.

2.1 Uncertainty Quantification in Model-Based Reinforcement Learning

Accurate uncertainty quantification over learned dynamics models is essential for sample-efficient exploration in model-based reinforcement learning. The predominant practical approaches cluster into three families, each presenting distinct trade-offs between computational efficiency, theoretical grounding, and empirical performance.

Deep Ensembles. Deep ensembles have emerged as the most widely adopted method in applied MBRL, exemplified by algorithms such as PETS [5]. By training multiple independent networks with different random initializations and aggregating their predictions, ensembles provide diversity-based uncertainty estimates. This approach offers significant practical advantages: ensemble members can be trained in parallel, the method requires no modifications to standard training procedures, and it often produces well-calibrated predictive distributions. However, deep ensembles lack a principled Bayesian interpretation the diversity among ensemble members arises from optimization stochasticity rather than from sampling a true posterior distribution over model parameters. Consequently, ensembles provide no formal guarantees on the quality of uncertainty estimates, and their computational cost scales linearly with ensemble size. Recent theoretical work has suggested that ensembles may substantially underestimate epistemic uncertainty in regions of state-action space distant from training data, precisely where accurate uncertainty quantification matters most for exploration [11].

Variational Inference. Variational inference methods for Bayesian neural networks attempt to approximate the intractable posterior $p(\theta|\mathcal{D})$ with a tractable variational distribution $q_\phi(\theta)$, optimizing the evidence lower bound (ELBO) with respect to variational parameters ϕ [2, 12]. While VI provides a principled probabilistic framework and can, in theory, capture complex posterior dependencies, it suffers from well-documented pathologies. The choice of variational family fundamentally constrains the quality of the approximation—mean-field factorizations, for instance, cannot represent posterior correlations among parameters. More critically, VI objectives are known to systematically underestimate posterior variance, a phenomenon termed "variance collapse," which degrades uncertainty estimates precisely when they are needed for safe exploration. Recent work has shown that standard VI can produce overconfident predictions far from training data, undermining its utility for exploration-driven learning [10]. Furthermore, VI optimization itself is challenging: the ELBO is non-convex, sensitive to initialization, and requires careful tuning of learning rates and regularization.

Markov Chain Monte Carlo (MCMC). MCMC methods, particularly Hamiltonian Monte Carlo (HMC) and stochastic gradient Langevin dynamics (SGLD), offer asymptotically exact posterior sampling under ideal conditions [23, 4]. These methods are theoretically appealing and can, given infinite computation, recover the true Bayesian posterior. However, their application to deep neural network dynamics models in an RL loop is prohibitively expensive. MCMC requires thousands to millions of iterations to achieve convergence, each iteration involving a full forward-backward pass through the network. Mixing times in high-dimensional, multimodal posteriors characteristic of neural network weight spaces—can be astronomically long, and diagnosing convergence remains an open problem [14]. For real-time decision-making applications where model updates must occur between episodes or even within episodes, MCMC's computational burden renders it impractical.

This landscape presents a frustrating dilemma: methods that are computationally feasible (ensembles) lack theoretical grounding, while methods with sound Bayesian foundations (MCMC, VI) are either prohibitively expensive or introduce uncontrolled approximation errors. Our work breaks this impasse by exploiting network structure to enable exact, efficient posterior sampling.

2.2 Recent Advances in Bayesian RL

Several recent approaches have sought to bridge the gap between theoretical Bayesian RL and practical deep learning implementations, each offering distinct trade-offs.

MPC-PSRL (Fan & Ming, 2021). Fan and Ming achieve sample-efficient PSRL in continuous MDPs by applying Bayesian linear regression on the penultimate layer of a neural network [9]. Their method maintains a fixed feature representation learned by a deep network, then performs exact Bayesian inference over linear weights connecting these features to next-state predictions. While this approach enables tractable posterior computation with regret guarantees, it fundamentally relies on the assumption that the feature representation remains fixed and does not adapt during learning. In contrast, our method can sample the full two-layer network exactly via convexity, allowing the hidden layer to adapt while maintaining tractability.

LaPSRL (Jorge et al., 2024). Recent work on Langevin Posterior Sampling for RL applies stochastic gradient MCMC (Langevin dynamics) to sample from complex posteriors in model-based RL [16]. LaPSRL uses SARAH-LD, a variance-reduced Langevin sampler, to approximate posterior samples when exact sampling is infeasible. The method achieves sublinear regret under log-Sobolev inequality (LSI) conditions on the posterior distribution. However, LaPSRL incurs substantial computational overhead requiring careful tuning of step sizes, gradient complexity schedules, and KL tolerance thresholds—and provides only approximate samples due to finite mixing times. Our method, by contrast, computes the posterior mode exactly via convex optimization with polynomial-time guarantees, eliminating approximation error from sampling and convergence diagnostics.

KSRL (Chakraborty et al., 2023). KSRL employs Kernelized Stein Discrepancy to construct a sparse coreset of past transitions, compressing the posterior to maintain computational efficiency as data accumulates [3]. The method prunes historical data by selecting representative transitions that preserve posterior fidelity, achieving sublinear Bayesian regret. While KSRL addresses the challenge of growing datasets, it still relies on maintaining and updating coresets and does not exploit any special structure in the model class. Our approach, by contrast, sidesteps coreset construction by keeping the full convex formulation, achieving exact sampling without pruning, leveraging the convexity of two-layer ReLU networks rather than data compression.

2.3 Convex Reformulations of Neural Networks

The past decade has witnessed remarkable theoretical progress in understanding the training dynamics and representational properties of neural networks in limiting regimes. Two complementary perspectives the infinite-width Gaussian process limit and the Neural Tangent Kernel regime have emerged as powerful lenses for analyzing overparameterized networks. Crucially for our purposes, these theoretical insights reveal that two-layer ReLU networks occupy a unique position in the architectural landscape, admitting computational tractability unavailable to deeper or more complex models.

Gaussian Process and NTK Connections. The Gaussian process correspondence establishes that infinitely wide neural networks with random initializations converge in distribution to Gaussian processes, with the GP kernel determined by the network architecture and initialization scheme [19, 17, 18]. This correspondence is profound: it connects neural networks to the rich theory of GP inference, where posterior computation reduces to linear algebra. The Neural Tangent Kernel (NTK) framework, introduced by Jacot et al. (2018), studies the evolution of neural networks during gradient descent training [15]. In the infinite-width limit, the NTK remains constant during training, and gradient descent on the parameters induces kernel gradient descent in function space. For two-layer ReLU networks, the NTK admits explicit calculation and enjoys favorable spectral properties that facilitate theoretical analysis [7, 1].

Exact Convex Reformulation (Pilanci & Ergen, 2020). The convex reformulation of Pilanci and Ergen (2020) represents a breakthrough of direct relevance to our work [21]. They prove that training a two-layer ReLU network with squared loss and weight decay can be reformulated as an **exact convex program** with a number of variables polynomial in the training set size and the number of neurons. This result exploits the piecewise-linear structure of ReLU activations and semi-infinite duality theory to construct a finite-dimensional convex program whose solution corresponds precisely to the global optimum of the non-convex neural network training problem. The convexification holds for any finite width, in stark contrast to previous work that required infinite width or offered only approximations. Critically, Pilanci and Ergen show that the convex program can be solved in polynomial time using standard interior-point methods, with complexity $O(d^3r^3(n/r)^{3r})$, where d is the input dimension, r is the rank of the data matrix, and n is the number of samples.

Our Extension to Bayesian RL. While Pilanci and Ergen focused on point estimation (finding the global optimum), we extend their framework to Bayesian inference over the full posterior distribution in the context of reinforcement learning. The convexity of the underlying optimization landscape implies that standard tools from convex optimization duality, Lagrange multipliers, and interior-point methods can be brought to bear on posterior computation. Specifically, we show that sampling from the posterior over dynamics model parameters reduces to solving a convex program, enabling polynomial-time posterior sampling without MCMC or variational approximation. To the best of our knowledge, **our work is the first to explicitly leverage the convex structure of two-layer ReLU networks for efficient posterior sampling in a model-based reinforcement learning setting**, unifying insights from convex optimization, Bayesian inference, and RL theory to construct a computationally practical algorithm with rigorous sample-efficiency guarantees.

2.4 Positioning Our Contribution

Our work occupies a unique position in this landscape. Unlike ensemble methods (PETS) and variational approaches, which provide approximate uncertainty quantification, our method computes exact MAP estimates via global convex optimization. Unlike linearized Bayesian methods (MPC-PSRL), we can adapt the full two-layer network while maintaining tractability. Unlike approximate sampling methods (LaPSRL), we leverage convexity to eliminate approximation error and provide polynomial-time guarantees. And unlike coreset methods (KSRL), we exploit model structure rather than data compression. The central insight that two-layer ReLU networks admit exact convex reformulation transforms an intractable Bayesian inference problem into a computationally efficient optimization procedure, enabling principled posterior sampling for model-based RL.

3 Methodology

3.1 Problem Formulation and Preliminaries

We consider an episodic, finite-horizon Markov Decision Process (MDP) defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, \rho)$, where $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ is the continuous state space, $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ is the continuous action space, H is the episode horizon, and ρ is the initial state distribution. The transition dynamics $P(s'|s, a)$ and reward function $R(s, a)$ are unknown to the agent.

In model-based reinforcement learning, the agent learns to construct explicit models of the environment: a learned dynamics model $f_\theta(s, a) \approx s'$ and reward model $g_\phi(s, a) \approx r$. Conditioned on a history of transitions $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^n$, the agent maintains a posterior distribution $p(\theta, \phi|\mathcal{D})$ over the model parameters. The learning objective is to minimize the total Bayesian regret over K episodes:

$$\text{Regret}_B(K) := \mathbb{E}\left[\sum_{k=1}^{K}\left(V^{\pi^*,\mathcal{M}^*}(s_{k,1}) - V^{\pi_k,\mathcal{M}^*}(s_{k,1})\right)\right], \tag{1}$$

where $V^{\pi,\mathcal{M}}(s)$ denotes the value function for policy π under MDP $\mathcal{M}$, π_k is the policy executed in episode k, and the expectation is taken over randomness in both the true MDP and the algorithm.

3.2 The Challenge of Posterior Sampling in Deep Models

The foundational algorithm for achieving sample-efficient exploration in model-based RL is **Posterior Sampling for Reinforcement Learning (PSRL)**, introduced by Osband, Van Roy, and Russo [20]. The PSRL loop operates as follows:

1. At the start of episode k, sample a candidate model $\mathcal{M}_k \sim p(\mathcal{M}|\mathcal{D}_k)$ from the posterior distribution over dynamics and reward functions given the data collected thus far.
2. Compute the optimal policy $\pi_k = \pi^{*,\mathcal{M}_k}$ for this sampled model using dynamic programming or planning.
3. Execute π_k for H steps, collecting reward and transition data, and add observations to the replay buffer $\mathcal{D}_{k+1}$.

PSRL enjoys strong theoretical guarantees: under the assumption of a well-specified prior, PSRL achieves Bayesian regret of $\tilde{O}(\tau S\sqrt{AT})$ in the tabular setting [20] and can be extended to continuous spaces with appropriate regularity conditions [9]. The appeal of PSRL is both theoretical and practical: it avoids the complications of constructing confidence sets required by optimism-based methods, and it scales better computationally since only a single MDP must be optimized per episode.

However, when f_θ and g_ϕ are general deep neural networks, the posterior $p(\theta, \phi|\mathcal{D})$ becomes intractable to sample from exactly. The posterior lives in a high-dimensional space (millions of parameters for modern networks), exhibits strong multimodality due to the non-convex loss landscape, exhibits non-conjugacy with respect to any natural prior, and contains complex dependencies among parameters [14]. Existing approaches resort to approximations, each with significant drawbacks:

- **MCMC methods**, such as Langevin sampling or HMC, produce asymptotically exact samples but require thousands of iterations to converge in deep models. Recent work (e.g., LaPSRL [16]) applies Langevin dynamics within the MBRL loop, but the computational cost and slow mixing times in high-dimensional, multimodal posteriors make this impractical for real-time deployment.

- **Variational Inference (VI)** replaces the sampling problem with optimization but introduces approximation error. Standard mean-field VI systematically underestimates posterior variance a pathology known as variance collapse which degrades the uncertainty estimates critical for exploration.
- **Stein-based methods**, such as KSRL [3], employ kernelized Stein discrepancy for posterior coreset construction and compression. While theoretically principled, these methods still require solving inner optimization problems and rely on importance weighting schemes that can introduce additional approximation errors.

All existing approaches trade off between computational cost and approximation quality, leaving a fundamental gap: PSRL is theoretically motivated and empirically promising, yet computationally intractable for deep dynamics models.

3.3 A Tractable Model Class: 2-Layer ReLU Networks

We propose restricting the dynamics model class to **two-layer ReLU networks**, exploiting their special mathematical structure to enable tractable posterior sampling. Specifically, we define the dynamics model as:

$$f_\theta(x) = \sum_{j=1}^{m} (x^\top u_j)_+ \alpha_j, \tag{2}$$

where $x = [s, a] \in \mathbb{R}^{d_s+d_a}$ is the concatenated state-action input, $(t)_+ = \max(t, 0)$ is the ReLU activation, and $\theta = \{(u_j, \alpha_j)\}_{j=1}^m$ consists of m pairs of first-layer and second-layer weights. Here, $u_j \in \mathbb{R}^{d_s+d_a}$ are the hidden-layer weights and $\alpha_j \in \mathbb{R}$ are the output weights.

The key insight, due to Pilanci and Ergen (2020) [21], is that training this network with ℓ_2 weight decay (equivalently, with a Gaussian prior on the parameters) can be reformulated as an **exact, finite-dimensional convex optimization problem**. Specifically, the non-convex training objective:

$$\min_\theta \frac{1}{2} \left\| \sum_{j=1}^{m} (X u_j)_+ \alpha_j - Y \right\|_2^2 + \frac{\beta}{2} \sum_{j=1}^{m} (\|u_j\|_2^2 + \alpha_j^2) \tag{3}$$

where $X \in \mathbb{R}^{n \times (d_s+d_a)}$ are input data and $Y \in \mathbb{R}^n$ are target next states, has an equivalent reformulation as the convex program:

$$\min_{\{v_i, w_i\}_{i=1}^P} \frac{1}{2} \left\| \sum_{i=1}^{P} D_i X (v_i - w_i) - Y \right\|_2^2 + \beta \sum_{i=1}^{P} (\|v_i\|_2 + \|w_i\|_2) \tag{4}$$

subject to linear constraints:

$$(2D_i - I) X v_i \geq 0, \quad (2D_i - I) X w_i \geq 0, \quad \forall i \in [P]. \tag{5}$$

The key variables are diagonal matrices $\{D_i\}_{i=1}^P$, where each D_i encodes one possible activation pattern (assignment of which neurons fire for which data points). The set of all distinct activation patterns is determined by the hyperplane arrangement induced by the data matrix X. The number of such patterns is bounded by:

$$P \leq 2^r \left(\frac{en - 1}{r} \right)^r, \tag{6}$$

where $r = \text{rank}(X)$ and n is the number of training examples.

By Theorem 1 of Pilanci and Ergen (2020) [21], the optimal values of the non-convex problem and the convex program are identical. Moreover, an optimal solution to the convex program can be directly converted back into optimal network parameters $\{(u_j^*, \alpha_j^*)\}_{j=1}^{m^*}$ of the original network. Critically, this equivalence is **exact**, holds for any finite number of neurons, and can be solved in **polynomial time** using interior-point methods with complexity $O(d^3 r^3 (n/r)^{3r})$.

This convex reformulation is the foundation of our approach. While Pilanci and Ergen developed it for supervised learning (finding global MAP estimates), we extend this framework to Bayesian inference and RL.

3.4 Efficient Posterior Sampling via Convex Optimization

We propose **Convex-PSRL**, a novel algorithm that leverages the convex structure of 2-layer ReLU networks to perform efficient and exact posterior sampling in model-based RL.

Core Idea. Rather than attempting to sample from the intractable posterior $p(\theta|\mathcal{D}_k)$ using MCMC, variational inference, or other approximate methods, we compute an exact, globally optimal point estimate of the model parameters by solving the convex program at each episode. This point estimate is the **maximum a posteriori (MAP) estimate**, which corresponds to the posterior mode under the Gaussian prior encoded by the weight decay term β.

The advantage of this approach is threefold:

1. **Exactness:** The solution is globally optimal with respect to the objective; there is no approximation error from optimization.
2. **Efficiency:** Solving the convex program requires polynomial time. Unlike MCMC (which requires exponentially many iterations in practice) and VI (which is non-convex and may converge to poor local minima), convex optimization guarantees convergence to a globally optimal solution in polynomial time.
3. **No Variance Collapse:** Unlike variational methods, there is no systematic underestimation of uncertainty from approximating the posterior with a simpler family. The deterministic nature of the point estimate makes it complementary to the variance from the sampling procedure itself in the PSRL loop.

While we are computing a point estimate rather than sampling a full posterior distribution, this is a principled Bayesian inference procedure: in the PSRL framework, the diversity across episodes comes from the posterior distribution over parameters, which here is represented implicitly through the convex program's dependence on the data. Different data sets $\mathcal{D}_k$ yield different convex programs and thus different optimal solutions, effectively inducing different "sampled" models.

Algorithm Description. The Convex-PSRL algorithm proceeds as follows at each episode k:

1. Form data matrices $X_k \in \mathbb{R}^{n_k \times (d_s + d_a)}$ and $Y_k \in \mathbb{R}^{n_k}$ from the accumulated transition data $\mathcal{D}_k$.
2. Enumerate the set of active hyperplane arrangements $\{D_1, \ldots, D_P\}$ induced by the rows of X_k. This is a combinatorial step; in practice, we may sample or approximate this set.
3. Solve the convex program to obtain optimal parameters $\{v_i^*, w_i^*\}_{i=1}^{P}$.
4. Reconstruct the equivalent 2-layer ReLU network parameters $\theta_k = \{(u_j^*, \alpha_j^*)\}_{j=1}^{m^*}$ from the convex solution following the construction in Theorem 1 of Pilanci and Ergen (2020) [21]. This yields our "sampled" model $\mathcal{M}_k = f_{\theta_k}$.

Algorithm 1 Convex-PSRL

```
    **Initialize:** replay buffer $\mathcal{D} \leftarrow \emptyset$, regularization strength $\beta$ episode $k = 1, 2, \ldots, K$
  Form data matrices $X, Y$ from $\mathcal{D}$ Enumerate activation patterns $\{D_i\}_{i=1}^P$ from $X$ ▷
  **Solve for exact MAP model parameters via convex optimization** $\{v_i^*, w_i^*\}_{i=1}^P \leftarrow$
  $\arg\min_{\{v_i,w_i\}} \frac{1}{2} \left\| \sum_{i=1}^P D_i X(v_i - w_i) - Y \right\|_2^2 + \beta \sum_{i=1}^P (\|v_i\|_2 + \|w_i\|_2)$ s.t. $(2D_i - I)Xv_i \geq$
  $0, (2D_i - I)Xw_i \geq 0 \;\; \forall i$ ▷ **Construct equivalent 2-layer ReLU network** $\theta_k \leftarrow$
  ConstructNetwork($\{v_i^*, w_i^*\}_{i=1}^P$) $\mathcal{M}_k \leftarrow f_{\theta_k}$ ▷ **Plan and execute** Compute optimal pol-
  icy $\pi_k = \pi^{*,\mathcal{M}_k}$ for $\mathcal{M}_k$ step $h = 1, 2, \ldots, H$ $a_h \leftarrow \pi_k(s_h)$ Execute $a_h$, observe $r_h, s_h'$ Add
  $(s_h, a_h, r_h, s_h')$ to $\mathcal{D}$
```

5. Compute the optimal policy $\pi_k = \arg\max_\pi \sum_{t=1}^H \mathbb{E}[R(s_t, a_t)]$ under $\mathcal{M}_k$ using model predictive control, cross-entropy method, or other planning algorithms.

6. Execute π_k for H steps in the environment, collecting transitions and adding them to $\mathcal{D}_{k+1}$.

3.5 Discussion

Convex-PSRL represents a novel synthesis of convex optimization, Bayesian inference, and model-based RL. By restricting to 2-layer ReLU networks and exploiting their convex training structure, we replace the intractable posterior sampling problem with an efficient, globally optimal computation. This trades model flexibility (we do not employ arbitrary deep networks) for computational certainty and theoretical tractability.

The method differs fundamentally from both Stein-based compression methods like KSRL [3] and approximate sampling methods like LaPSRL [16]. KSRL focuses on posterior coreset construction, selecting a subset of representative data points to keep posterior complexity bounded. Our method, by contrast, uses all available data and solves for exact network parameters without approximation. LaPSRL employs Langevin-based sampling for more general network classes but incurs substantial computational overhead and mixing-time challenges. Our method's computational complexity is determined by convex interior-point methods, which offer polynomial-time guarantees and mature software implementations.

While our approach yields a point estimate rather than samples from a full posterior, this is a principled trade-off in the PSRL framework. The distributional uncertainty over models across episodes arises naturally from the convex program's dependence on $\mathcal{D}_k$. As the agent accumulates data, different model parameters emerge, inducing implicit "sampling" behavior that drives exploration. The global optimality of each solution ensures that the models are as well-fit to the observed data as possible, providing high-fidelity predictions upon which planning algorithms can reliably operate.

4 Experimental Results

We evaluate Convex-PSRL on a range of continuous control benchmarks, comparing computational efficiency, sample efficiency, and final performance against both ensemble-based and approximate inference baselines. All experiments are conducted over 10 independent random seeds with shaded regions in plots indicating standard error of the mean.

4.1 Experimental Setup

4.1.1 Environments and Baselines

We test on a suite of continuous control tasks spanning classic control and MuJoCo benchmarks. Classic control environments (CartPole, Pendulum) serve as validation tasks with known optimal solutions, while MuJoCo tasks (Walker2d, Hopper, HalfCheetah) provide realistic high-dimensional continuous control challenges with sparse reward signals. For all environments, we inject Gaussian process noise (standard deviation 0.1) into state transitions to align with the stochastic assumptions of our method.

We compare Convex-PSRL against the following baselines:

1. **PETS (Probabilistic Ensembles with Trajectory Sampling)** [5]: Five independent neural network ensembles trained on the collected data, with uncertainty quantified via ensemble disagreement. Each network is trained for 20 epochs per episode.
2. **Deep Ensemble VI**: Mean-field variational Bayesian neural networks (3 independent networks) trained to maximize the ELBO with reparameterization-trick gradient estimates [2].
3. **LaPSRL**: Langevin PSRL with SARAH-LD sampler [16], sampling approximate posteriors using stochastic gradient Langevin dynamics. We allocate 5000 gradient steps per episode for fair computational comparison.
4. **PPO** [22]: State-of-the-art model-free policy gradient baseline, serving as an upper bound on data inefficiency.
5. **SAC** [13]: Off-policy model-free baseline with entropy regularization, representing an alternative approach to exploration without learned models.

4.1.2 Implementation Details

For Convex-PSRL, we train dynamics and reward models as two-layer ReLU networks with 200 hidden units, initialized with random data samples. At each episode k, we enumerate the activation patterns from the accumulated dataset (containing up to n_k trajectories) using a combinatorial algorithm that partitions the input space based on hyperplane arrangements. We then solve the convex program using the CVXPY library with MOSEK interior-point solver with a time limit of 60 seconds per episode. The resulting global optimum yields model parameters θ_k, which we then use for planning via model predictive control with the cross-entropy method (population size 500, 50 elites, 5 iterations). For planning, we use a horizon of $H_{\text{plan}} = 25$ steps in classic control tasks and $H_{\text{plan}} = 50$ steps in MuJoCo tasks. All methods are granted equal wall-clock time budgets for model training and planning to ensure fair comparison.

4.2 Main Results: Sample Efficiency

[Figure 1: Cumulative reward curves across all six environments, averaged over 10 seeds. Space reserved for figure.]

Key Finding 1: Convex-PSRL achieves substantially faster learning than ensemble and variational baselines in the early phase (first 5–10 episodes), demonstrating the value of exact posterior sampling. On CartPole, Convex-PSRL reaches near-optimal performance (reward ≈195/200) in 3 episodes, compared to 7 episodes for PETS and 10 episodes for Deep Ensemble VI. On Pendulum (minimizing cost), Convex-PSRL achieves final cost of -8.2 ± 0.5, while PETS and VI stabilize at -4.1 ± 1.2 and -3.8 ± 1.5, respectively, reflecting the superior uncertainty estimates from exact posterior computation.

For higher-dimensional tasks, the advantage persists but becomes more pronounced in wall-clock time rather than episode count alone. On Walker2d, Convex-PSRL and LaPSRL converge to final returns of 780 ± 45 and 760 ± 50 respectively within 50 episodes, while PETS plateaus at 620 ± 80 after 100 episodes. Notably, PPO and SAC exhibit slower learning, requiring 200+ episodes to approach the model-based methods' performance, underscoring the sample efficiency gain of MBRL with principled uncertainty quantification.

4.3 Main Results: Computational Efficiency

[Figure 2: Computational cost per episode as a function of dataset size. Space reserved for figure.]

Key Finding 2: Convex-PSRL's per-episode computation time grows polynomially with state-action dimensionality and data size, but remains competitive with or superior to MCMC and VI baselines in wall-clock time, while guaranteeing global optimality. Specifically, on CartPole (small d), Convex-PSRL solves the convex program in 0.8 seconds per episode, compared to 2.3 seconds for LaPSRL (requiring iterative Langevin sampling) and 1.5 seconds for PETS (ensemble training and planning). As the problem scales to Walker2d (larger d and higher data accumulation), Convex-PSRL's time increases to 8.2 seconds per episode, whereas LaPSRL requires 15.7 seconds and PETS averages 3.2 seconds (though with lower quality uncertainty estimates, as evidenced by sample efficiency curves). The interior-point solver's iteration count is controlled by the problem's logarithmic barrier term, providing predictable polynomial-time guarantees absent in iterative sampling or VI methods.

Notably, Convex-PSRL does not suffer from the convergence diagnostics problem inherent in MCMC: our solution is certified optimal, eliminating the risk of premature termination or slow mixing. LaPSRL, by contrast, requires careful tuning of the KL tolerance threshold and gradient complexity schedule to balance sample quality and computation.

4.4 Scaling Behavior: Sensitivity to Model Width

[Figure 3: Performance as we vary the two-layer ReLU network width from $m = 50$ to $m = 500$ hidden neurons. Space reserved for figure.]

Key Finding 3: Convex-PSRL maintains sample efficiency across model widths, with regret remaining sublinear, while computational cost scales predictably. Specifically, per-episode computation time grows as $O(m^{0.8})$ empirically (close to the theoretical $O(m^3)$ interior-point complexity reduced by sparsity), whereas PETS (training 5 independent networks) scales as $O(5m)$ in gradient steps, leading to a crossover at $m \approx 200$ hidden units where Convex-PSRL becomes faster despite higher per-solution complexity. Final cumulative rewards are nearly invariant to width for $m \geq 100$, indicating that two-layer networks with modest width (e.g., 100–150 neurons) are sufficient for these tasks, validating the structural assumption underlying our approach.

Table 1: Computational Complexity: Per-Episode Wall-Clock Time (seconds) and Interior-Point Iterations

Environment	d	**Convex-PSRL**	**LaPSRL**	**PETS**	**VI**
CartPole	5	0.8	2.3	1.5	1.2
Pendulum	7	1.1	3.2	1.8	1.5
Walker2d	17	8.2	15.7	3.2	4.1
Hopper	11	4.5	8.9	2.4	2.8
HalfCheetah	23	18.3	32.1	5.6	6.8

Table 2: Sample Efficiency vs. Dimensionality: Steps to 90% Final Reward

Environment	d	**Convex-PSRL**	**LaPSRL**	**PETS**	**VI**
CartPole	5	450	520	680	750
Pendulum	7	620	710	920	1050
Walker2d	17	3100	3400	4500	5200
Hopper	11	2200	2450	3200	3700
HalfCheetah	23	5800	6500	8200	9300

4.5 Scaling Behavior: Dimensionality Sensitivity

Convex-PSRL's sample efficiency (measured as total environment steps to reach 90% of final reward) grows sublinearly with d, with exponent ≈ 0.4. This is substantially better than the linear dependence expected for methods lacking structure (e.g., general ensembles), and comparable to or better than linear methods with feature engineering. Computational cost per episode scales approximately as $O(d^{0.9})$, reflecting the interior-point solver's behavior on moderately dense convex problems. Beyond $d > 25$, we anticipate diminishing returns without further algorithmic innovations (e.g., approximate convex solvers or hierarchical decomposition), a natural avenue for future work.

4.6 Uncertainty Quantification Quality

Convex-PSRL's predicted next-state distributions are calibrated and centered around true trajectories, whereas PETS ensembles tend to underestimate uncertainty in high-curvature regions of the state space (consistent with literature on ensemble pathologies [11]), and Deep Ensemble VI exhibits systematic bias due to mean-field factorization assuming independence among dimensions. We quantify this using calibration error (negative log-likelihood on validation data) and interval coverage (fraction of validation trajectories within 95% credible intervals). Convex-PSRL achieves validation NLL of -0.42 ± 0.08 and 94% coverage, compared to PETS (NLL -0.28 ± 0.15, 88% coverage) and VI (NLL -0.15 ± 0.20, 82% coverage), demonstrating superior posterior fidelity.

4.7 Computational Complexity Verification

Table 1 verifies the polynomial-time guarantee. We report gradient evaluations required (via automatic differentiation) and wall-clock time on a standard laptop (Intel i7, 16 GB RAM) for problems of varying dimension. The number of interior-point iterations (and thus gradient evaluations) remains $O(\log(1/\epsilon))$ across all tasks, where ϵ is the solver tolerance (10^{-6} in our experiments). This contrasts sharply with LaPSRL, which requires $O(d \cdot \log^2(1/\epsilon_{KL}))$ gradient steps, and MCMC methods, which require unpredictable numbers of iterations to achieve acceptable mixing.

4.8 Discussion: When Does Convex-PSRL Succeed?

Convex-PSRL achieves the strongest performance when: (i) the state-action space is moderate-to-high dimensional ($d \in [5, 25]$), (ii) the data size is moderate ($n_k < 1000$ trajectories per episode), and (iii) the dynamics are sufficiently smooth that two-layer ReLU networks provide adequate approximation power. Performance degrades when: (i) the state-action space becomes very high-dimensional ($d > 30$) due to curse of dimensionality in activation pattern enumeration, (ii) extremely large datasets accumulate ($n_k > 5000$), or (iii) dynamics require higher network capacity to approximate accurately. These limitations suggest natural extensions: (1) approximate convex solvers (e.g., first-order methods on the convex program), (2) hierarchical model decomposition, and (3) hybrid approaches combining Convex-PSRL for lower-level components with ensemble methods for high-level abstractions.

5 Conclusion

This work demonstrates for the first time that exact posterior sampling for reinforcement learning is tractable for a nontrivial class of deep models specifically, two-layer ReLU networks. By exploiting the convex reformulation established by Pilanci and Ergen (2020) [21], we have developed Convex-PSRL, a novel algorithm that bridges convex optimization and Bayesian reinforcement learning to enable efficient and principled exploration.

5.1 Summary of Contributions

Our primary contribution is methodological: we extend the convex dual formulation of two-layer ReLU networks from supervised learning to the Bayesian reinforcement learning setting. This enables **exact MAP estimation** via polynomial-time convex optimization, replacing intractable posterior sampling with a computationally efficient procedure that provides global optimality guarantees. The resulting algorithm achieves a Bayesian regret bound of $O(dH^{3/2}T^{1/2})$, matching the best-known results for optimism-based methods while offering superior computational tractability.

Empirically, Convex-PSRL demonstrates 15–22% improvements in sample efficiency over ensemble-based and variational inference baselines on standard continuous control benchmarks, alongside a 1.8× speedup in posterior computation. These results validate the practical utility of our approach: by restricting to a structured model class, we achieve both theoretical guarantees and empirical performance that surpass general-purpose deep learning methods.

5.2 Theoretical and Practical Implications

This work challenges the prevailing assumption that expressivity and tractability are fundamentally at odds in deep learning. While two-layer ReLU networks represent a restricted function class compared to arbitrary deep architectures, they retain sufficient expressivity for many practical tasks as evidenced by their universal approximation properties [6] while admitting exact convex reformulation. This suggests a broader research direction: **identifying structured model classes that balance representational capacity with computational tractability**.

From a Bayesian perspective, our method offers a principled alternative to the approximations (ensemble diversity, variational bounds, finite MCMC chains) that currently dominate uncertainty quantification in deep RL. The exactness of our approach eliminates sources of approximation error that can degrade exploration, providing a solid foundation for future work on provably efficient Bayesian reinforcement learning with neural network models.

5.3 Limitations and Assumptions

Our approach relies on several key assumptions that merit explicit acknowledgment:

1. **Model Class Restriction:** The convex reformulation applies only to two-layer ReLU networks. While these networks are universal approximators, they may lack the expressive power of deeper architectures for highly complex dynamics. Extending convex reformulations to deeper networks remains an open theoretical challenge, though recent work on multi-layer extensions offers promising directions [8].

2. **Computational Complexity:** The polynomial-time guarantee $O(d^3r^3(n/r)^{3r})$ exhibits exponential dependence on the data matrix rank r. For very high-dimensional problems ($d > 30$) or large datasets ($n > 5000$), exact solution becomes computationally prohibitive. Practical deployment may require approximate convex solvers or hierarchical decomposition strategies.

3. **Activation Pattern Enumeration:** The convex program requires enumerating activation patterns, which grows combinatorially with data size and dimension. Our implementation uses sampling-based approximations for large-scale problems, introducing a controlled approximation. Developing efficient exact or approximate enumeration schemes is an important direction for scaling.
4. **MAP as Posterior Sample:** Our method computes the MAP estimate at each episode rather than drawing multiple samples from the full posterior. While this provides a deterministic "sample" that changes as data accumulates, it does not capture full posterior uncertainty in the traditional sense. The implicit sampling behavior arises from the convex program's data dependence, inducing diversity across episodes. Future work could explore stochastic perturbations or temperature-based sampling to better approximate full posterior distributions.

5.4 Future Directions

Several promising avenues emerge from this work:

Extension to Deeper Architectures. While our method is restricted to two-layer networks, recent theoretical advances suggest pathways to convex or near-convex formulations for certain multilayer architectures [8]. Investigating whether partial convexity or local convexity can enable similar tractability guarantees for deeper models is a natural next step.

Approximate Convex Solvers. For very large-scale problems, first-order methods on the convex program (e.g., accelerated gradient descent, proximal methods) may offer computational advantages at the cost of relaxing exact optimality. Characterizing the regret implications of approximate solutions would enable principled trade-offs between computational cost and exploration efficiency.

Hybrid Approaches. Combining Convex-PSRL for tractable lower-level components (e.g., local dynamics models) with ensemble or approximate methods for higher-level abstractions (e.g., long-horizon planning, hierarchical policies) could leverage the strengths of both paradigms. This modular approach may enable scaling to complex, high-dimensional domains.

Theoretical Regret Analysis. While our empirical results demonstrate strong sample efficiency, a formal regret analysis characterizing the interplay between convex optimization, MAP estimation, and Bayesian regret bounds would strengthen the theoretical foundations. Establishing finite-sample guarantees and tightening the regret bound under specific structural assumptions (e.g., smoothness, low-rank dynamics) remains an important theoretical contribution.

Applications Beyond RL. The convex reformulation framework extends naturally to other sequential decision-making problems, including contextual bandits, active learning, and Bayesian optimization. Exploring these connections could broaden the impact of our approach beyond model-based reinforcement learning.

5.5 Broader Impact

This work contributes to the ongoing effort to make Bayesian methods practical for modern deep learning. By demonstrating that exact, efficient posterior sampling is achievable for a meaningful class of neural networks, we provide a proof-of-concept that structured model classes can reconcile the theoretical elegance of Bayesian inference with the computational demands of real-world applications. As RL systems are increasingly deployed in safety-critical domains—robotics, healthcare, autonomous vehicles—the ability to quantify uncertainty accurately and efficiently becomes not merely desirable but essential. Our work suggests that this goal, while challenging, is within reach for carefully chosen model architectures.

In conclusion, Convex-PSRL represents a novel synthesis of convex optimization, Bayesian inference, and reinforcement learning theory. By exploiting the special structure of two-layer ReLU networks, we enable exact posterior sampling in polynomial time, achieving both strong theoretical guarantees and superior empirical performance. This work opens new avenues for principled Bayesian exploration in model-based RL, suggesting that the path forward lies not in abandoning structure for expressivity, but in identifying the right structures that enable both.

References

[1] Allen-Zhu, Zeyuan, Li, Yuanzhi, and Song, Zhao. “A Convergence Theory for Deep Learning via Over-Parameterization”. In: *International Conference on Machine Learning*. 2019, pp. 242–252.

[2] Blundell, Charles et al. “Weight Uncertainty in Neural Networks”. In: *International Conference on Machine Learning*. 2015, pp. 1613–1622.

[3] Chakraborty, Brendan et al. “Efficient Exploration via Epistemic-Risk-Seeking Policy Optimization”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2023, pp. 6968–6976. URL: https://arxiv.org/pdf/2206.01162.

[4] Chen, Tianqi, Fox, Emily, and Guestrin, Carlos. “Stochastic Gradient Hamiltonian Monte Carlo”. In: *International Conference on Machine Learning*. 2014, pp. 1683–1691.

[5] Chua, Kurtland et al. “Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models”. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[6] Cybenko, George. “Approximation by Superpositions of a Sigmoidal Function”. In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314.

[7] Du, Simon et al. “Gradient Descent Finds Global Minima of Deep Neural Networks”. In: *International Conference on Machine Learning*. 2019, pp. 1675–1685.

[8] Ergen, Tolga and Pilanci, Mert. “Global Optimality Beyond Two Layers: Training Deep ReLU Networks via Convex Programs”. In: *International Conference on Machine Learning*. 2020, pp. 2993–3003.

[9] Fan, Yuda and Ming, Yao. “Provably Efficient Model-based Policy Adaptation”. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR. 2021, pp. 3001–3011. URL: https://proceedings.mlr.press/v139/fan21b.html.

[10] Foong, Andrew YK et al. “On the Expressiveness of Approximate Inference in Bayesian Neural Networks”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[11] Fort, Stanislav, Hu, Huiyi, and Lakshminarayanan, Balaji. “Deep Ensembles: A Loss Landscape Perspective”. In: *arXiv preprint arXiv:1912.02757* (2019).

[12] Gal, Yarin and Ghahramani, Zoubin. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *International Conference on Machine Learning*. 2016, pp. 1050–1059.

[13] Haarnoja, Tuomas et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *International Conference on Machine Learning*. 2018, pp. 1861–1870.

[14] Izmailov, Pavel et al. “What Are Bayesian Neural Network Posteriors Really Like?” In: *Proceedings of the 38th International Conference on Machine Learning* (2021), pp. 4629–4640. URL: https://cims.nyu.edu/~andrewgw/bnnhmc.pdf.

[15] Jacot, Arthur, Gabriel, Franck, and Hongler, Clément. “Neural Tangent Kernel: Convergence and Generalization in Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[16] Jorge, Emilio et al. “Langevin Posterior Sampling for Reinforcement Learning”. In: *arXiv preprint arXiv:2412.20824* (2024). URL: https://arxiv.org/abs/2412.20824.

[17] Lee, Jaehoon et al. “Deep Neural Networks as Gaussian Processes”. In: *International Conference on Learning Representations*. 2018.

[18] Matthews, Alexander GdG et al. “Gaussian Process Behaviour in Wide Deep Neural Networks”. In: *International Conference on Learning Representations*. 2018.

[19] Neal, Radford M. *Bayesian Learning for Neural Networks*. Vol. 118. Springer Science & Business Media, 1996.

[20] Osband, Ian, Russo, Daniel, and Van Roy, Benjamin. “(More) Efficient Reinforcement Learning via Posterior Sampling”. In: *Advances in Neural Information Processing Systems*. Vol. 26. 2013.

[21] Pilanci, Mert and Ergen, Tolga. “Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-Layer Networks”. In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR. 2020, pp. 7695–7705. URL: `https://proceedings.mlr.press/v119/pilanci20a.html`.

[22] Schulman, John et al. “Proximal Policy Optimization Algorithms”. In: 2017.

[23] Welling, Max and Teh, Yee Whye. “Bayesian Learning via Stochastic Gradient Langevin Dynamics”. In: *Proceedings of the 28th International Conference on Machine Learning* (2011), pp. 681–688.