Scalable Thompson Sampling via Optimal Transport

Ruiyi Zhang[1] **Zheng Wen**[2] **Changyou Chen**[3] **Chen Fang**[2] **Tong Yu**[4] **Lawrence Carin**[1]
[1]Duke University [2]Adobe Research [3]University at Buffalo [4]Carnegie Mellon University
ryzhang@cs.duke.edu, zwen@adobe.com

Abstract

Thompson sampling (TS) is a class of algorithms for sequential decision making, in which a posterior distribution is maintained over a reward model. However, calculating exact posterior distributions is intractable for all but the simplest models. Development of computationally-efficiently approximate methods for the posterior distribution is consequently a crucial problem for scalable TS with complex models, such as neural networks. In this paper, we use distribution optimization techniques to approximate the posterior distribution, solved via Wasserstein gradient flows. Based on the framework, a principled particle-optimization algorithm is developed for TS to approximate the posterior efficiently. Our approach is scalable and does not make explicit distribution assumptions on posterior approximations. Extensive experiments on both synthetic and real large-scale data demonstrate the superior performance of the proposed methods.

1 Introduction

In many online sequential decision-making problems, such as contextual bandits [Bubeck et al., 2012] and reinforcement learning [Sutton and Barto, 1998], an agent learns to take a sequence of actions to maximize its expected cumulative reward, while repeatedly interacting with an unknown environment. Moreover, since in such problems the agent's actions affect both its rewards and its observations, it faces the well-known exploration-exploitation tradeoff. Consequently, the exploration strategy is crucial for a learning algorithm: under-exploration will typically yield a sub-optimal strategy, while over-exploration tends to incur a significant exploration cost.

Various exploration strategies have been proposed, including ϵ-greedy (EG), Boltzmann exploration [Sutton, 1990, Cesa-Bianchi et al., 2017], upper-confidence-bound (UCB) [Agrawal, 1995, Auer, 2002] type exploration, and Thompson sampling (TS). Among them, TS [Thompson, 1933, Russo et al., 2018], which is also known as posterior sampling or probability matching, is a widely used exploration strategy with good practical performance [Li et al., 2010, Chapelle and Li, 2011] and theoretical guarantees [Russo and Van Roy, 2016, Agrawal and Goyal, 2012, Russo et al., 2018]. For contextual bandits, the vanilla version of TS maintains a posterior distribution over all the plausible models. During each round of interacting with the environment, it (*i*) first samples a model from the current posterior; (*ii*) then chooses an action that is optimal under the sampled model, and receives a reward; and finally it (*iii*) updates the posterior based on its action and reward in this round of interaction.

Vanilla TS is computationally efficient when a closed-form posterior is available, such as for Bernoulli or Gaussian rewards. For cases without a closed-form posterior, computational variants of TS have also been developed. Some of them are based on approximate posteriors, such as Gaussian TS for linear bandits with non-Gaussian rewards [Agrawal and Goyal, 2013], or TS with Laplace approximation for contextual bandits with a logistic regression model [Chapelle and Li, 2011]. However, most such TS algorithms cannot be extended in a computationally efficient manner to cases with complex generalization models, such as neural networks.

Posterior approximations have also been widely studied in the field of approximate inference. Variational inference is a common method for approximating the posterior, but the explicit-form assumption usually leads to underestimation of uncertainty. This idea leads to TS via variational inference, which has been explored in [Urteaga and Wiggins, 2018]. Alternatively, posterior

*Work performed at Adobe Research. Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

sampling has been achieved by particle-optimization-based sampling methods, which show significant advantages without assumptions on the form of the posterior distribution.

In this paper, adopting ideas from the Wasserstein-gradient-flow literature, we propose a general particle-based distribution optimization framework for TS. Our framework improves the recently proposed particle-based framework for TS [Lu and Van Roy, 2017]. Specifically, our framework employs a set of particles interacting with each other to approximate the posterior distribution. The proposed method is called particle-interactive Thompson sampling, or π-TS, distinct from [Lu and Van Roy, 2017] which treats particles independently. In our proposed framework, the posterior can be efficiently updated with new observations via gradient-descent-based methods, thus drawing samples from the approximate posterior distribution is fast.

Specifically, we optimize the posterior distribution in TS based on the Wasserstein-gradient-flow framework. In this setting, Bayesian sampling in TS becomes a *convex* optimization problem on the space of *probability measures*, thus the optimality of the learned distribution could be guaranteed. For tractability, the posterior distribution in TS is approximated by a set of particles (a.k.a. samples). We test our framework on a number of applications, first in simulated scenarios and then on large-scale real-world datasets. In all cases, our proposed particle-interactive Thompson sampling significantly outperforms other baselines.

2 Large-scale Contextual Bandits

2.1 Contextual Bandits

We consider a contextual bandit characterized by triple $(\mathcal{X}, \mathcal{A}, P)$, where $\mathcal{X} \subseteq \Re^d$ is a context (state) space with dimension d, $\mathcal{A} = \{1, 2, \ldots, K\}$ is a finite action space, and P encodes the reward distributions at all the context-action pairs. Specifically, for all context-action pairs $(\mathbf{x}, \mathbf{a}) \in \mathcal{X} \times \mathcal{A}$, if an *agent* chooses action $\mathbf{a}$ for context $\mathbf{x}$, then it receives a stochastic reward r that is drawn conditionally independent from $P_{\mathbf{x},\mathbf{a}}$. We also use $\bar{r}(\mathbf{x}, \mathbf{a})$ to denote the mean of the reward distribution $P_{\mathbf{x},\mathbf{a}}$.

In the contextual bandit setting, the agent is assumed to know $\mathcal{X}$ and $\mathcal{A}$, but not P. The agent repeatedly interacts with the contextual bandit for T rounds. At each round $t = 1, \ldots, T$, the agent first observes a context $\mathbf{x}_t \in \mathcal{X}$, which is independently chosen by the *environment*. The agent then adaptively chooses an action $\mathbf{a}_t \in \mathcal{A}$, based on the current context $\mathbf{x}_t$ and the agent's past observations. Finally, the agent observes and receives a reward r_t, which is drawn conditionally independently from the reward distribution $P_{\mathbf{x}_t,\mathbf{a}_t}$. By definition, $\mathbb{E}\left[r_t | \mathbf{x}_t, \mathbf{a}_t\right] = \bar{r}(\mathbf{x}_t, \mathbf{a}_t)$.

The agent's objective is to learn to maximize its expected cumulative reward $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$ in the first T rounds, or equivalently, to minimize its expected cumulative regret in the first T rounds:

$$R(T) = \sum_{t=1}^{T} \mathbb{E}\left[\max_{\mathbf{a} \in \mathcal{A}} [\bar{r}(\mathbf{x}_t, \mathbf{a})] - r_t\right]. \tag{1}$$

2.2 Reward Generalization

Many practical online decision-making problems that fit in the framework of a contextual bandit have intractably large scale. Specifically, in such problems, at least one of $\mathcal{X}$ or $\mathcal{A}$ has unmanageably large cardinality (if it is discrete) or dimension (if it is continuous). One standard approach for developing scalable learning algorithms for such large-scale problems is to exploit generalization models.

Specifically, in this paper we assume that the agent has access to a generalization model $m(\mathbf{x}, \mathbf{a}; \boldsymbol{\theta})$ for the mean reward function $\bar{r}(\mathbf{x}, \mathbf{a})$, where $\boldsymbol{\theta}$ is a vector encoding the model parameters. We assume that the generalization model m is "appropriate" in the sense that there exists a specific model parameter vector $\boldsymbol{\theta}^*$ s.t.

$$m(\mathbf{x}, \mathbf{a}; \boldsymbol{\theta}^*) \approx \bar{r}(\mathbf{x}, \mathbf{a}) \quad \forall (\mathbf{x}, \mathbf{a}) \in \mathcal{X} \times \mathcal{A}.$$

We say m is a *perfect* generalization model if exact equality holds in the above equation. Note that m is a function of the context-action pair $(\mathbf{x}, \mathbf{a})$ and the parameter vector $\boldsymbol{\theta}$. We further assume that the agent knows the function m, but not $\boldsymbol{\theta}^*$. Consequently, by exploiting this generalization model, it is sufficient to learn $\boldsymbol{\theta}^*$ to choose near-optimal actions.

One example of the above-mentioned generalization model is a neural network. In this case m represents the topology and activation functions of the neural network, $(\mathbf{x}, \mathbf{a})$ is the input to the neural network, and $\boldsymbol{\theta}$ encodes all the edge weights. Notice that simpler generalization models, such as linear regression models and logistic regression models, can be viewed as special cases of neural networks.

2.3 Thompson Sampling

Thompson sampling (TS) [Thompson, 1933] is a widely used class of algorithms for bandits, contextual bandits, and reinforcement learning. For contextual bandits with reward generalization considered in Section 2.1 and 2.2, assuming the reward generalization is perfect, the vanilla version of TS maintains a posterior distribution p_t over $\boldsymbol{\theta}^*$, which is initialized as a prior distribution p_0. At each time t, TS first samples a parameter vector $\hat{\boldsymbol{\theta}}_t$ from the current posterior p_{t-1}; it then chooses action $\mathbf{a}_t \in \text{argmax}_{\mathbf{a}}\, m(\mathbf{x}_t, \mathbf{a}; \hat{\boldsymbol{\theta}}_t)$ and receives reward r_t; finally, it updates the posterior based on Bayes rule. The pseudocode of the vanilla TS algorithm is provided in Algorithm 1.

Algorithm 1 Thompson Sampling (Vanilla Version)

```
Require: prior distribution $p_0$
1: for $t = 1, 2, \ldots, T$ do
2:     Observe context $\mathbf{x}_t$
3:     Draw $\hat{\boldsymbol{\theta}}_t$ from the posterior $p_{t-1}$
4:     Select $\mathbf{a}_t \in \arg\max_{\mathbf{a}} m(\mathbf{x}_t, \mathbf{a}; \hat{\boldsymbol{\theta}}_t)$
5:     Observe and receive reward $r_t$
6:     Update posterior $p_t(\boldsymbol{\theta}) \leftarrow p_{t-1}(\boldsymbol{\theta}|(\mathbf{x}_t, \mathbf{a}_t, r_t))$.
7: end for
```

Though various regret bounds have been developed for the vanilla TS, it has two major limitations. First, it typically requires the reward generalization model m to be perfect; otherwise, it is very complicated to define the posterior. Second, for many generalization models such as a neural network, it is computationally infeasible to update and sample from the exact posterior. In this paper, we develop a novel and scalable version of TS based on optimal transport (Wasserstein gradient flows), which overcomes these limitations.

3 Wasserstein Gradient Flows

We first review Wasserstein gradient flows (WGF), which corresponds to gradient descent on the space of probability measures. For ease of understanding, we first motivate from gradient flows on the Euclidean space.

Gradient flows on the Euclidean space For a smooth function[*] $F : \mathbb{R}^d \to \mathbb{R}$, and a starting point $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, the gradient flow of $F(\boldsymbol{\theta})$ is defined as the solution of the differential equation: $\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}\tau} = -\nabla F(\boldsymbol{\theta}(\tau))$, for time $\tau > 0$ and initial condition $\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$. This is a standard Cauchy problem [Rulla, 1996], endowed with a unique solution if ∇F is Lipschitz continuous. When F is non-differentiable, the gradient is replaced with its subgradient, which gives a similar definition, omitted here for simplicity.

Gradient flows on the probability measure space WGF is a generalization of gradient flows on Euclidean space by lifting the differential equation above onto the space of probability measures, denoted $\mathcal{P}(\Omega)$ with $\Omega \subset \mathbb{R}^d$. Formally, we first endow a Riemannian geometry [Carmo, 1992] on $\mathcal{P}(\Omega)$. The geometry is characterized by the length between two elements (two distributions), defined by the second-order Wasserstein distance:

$$W_2^2(\mu, \nu) \triangleq \inf_{\gamma} \left\{ \int_{\Omega \times \Omega} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2 \mathrm{d}\gamma(\boldsymbol{\theta}, \boldsymbol{\theta}') : \gamma \in \Gamma(\mu, \nu) \right\} ,$$

where $\Gamma(\mu, \nu)$ is the set of joint distributions over $(\boldsymbol{\theta}, \boldsymbol{\theta}')$ such that the two marginals equal μ and ν, respectively. The Wasserstein distance defines an optimal-transport problem, where one wants to transform μ to ν with minimum cost [Villani, 2008]. Thus the term $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2$ represents the cost to transport $\boldsymbol{\theta}$ in μ to $\boldsymbol{\theta}'$ in ν, and can be replaced by a general metric $c(\boldsymbol{\theta}, \boldsymbol{\theta}')$ in a metric space. If μ is absolutely continuous w.r.t. the Lebesgue measure, there is a unique optimal transport plan from μ to ν, *i.e.*, a mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ pushing μ onto ν satisfying $T_{\#}\mu = \nu$. Here $T_{\#}\mu$ denotes the pushforward measure [Villani, 2008] of μ. The Wasserstein distance thus can be equivalently reformulated as

$$W_2^2(\mu, \nu) \triangleq \inf_{T} \left\{ \int_{\Omega} \|\boldsymbol{\theta} - T(\boldsymbol{\theta})\|_2^2 \mathrm{d}\mu(\boldsymbol{\theta}) \right\} , \qquad (2)$$

Consider $\mathcal{P}(\Omega)$ with a Riemannian geometry endowed by the second-order Wasserstein metric. Let $\{\mu_\tau\}_{\tau \in [0,1]}$ be an absolutely continuous curve in $\mathcal{P}(\Omega)$ with distance between μ_τ and $\mu_{\tau+h}$ measured by $W_2^2(\mu_\tau, \mu_{\tau+h})$. We overload the definition of T to denote the underlying transformation from μ_τ to $\mu_{\tau+h}$ as $\boldsymbol{\theta}_{\tau+h} = T_h(\boldsymbol{\theta}_\tau)$. Motivated by the Euclidean-space case, if we define $\mathbf{v}_\tau(\boldsymbol{\theta}) \triangleq \lim_{h \to 0} \frac{T_h(\boldsymbol{\theta}_\tau) - \boldsymbol{\theta}_\tau}{h}$ as the *velocity of the particle*, a gradient flow can be defined on $\mathcal{P}(\Omega)$ correspondingly in Lemma 1 [Ambrosio et al., 2005].

Lemma 1 *Let $\{\mu_\tau\}_{\tau \in [0,1]}$ be an absolutely-continuous curve in $\mathcal{P}(\Omega)$ with finite second-order moments. Then for a.e. $\tau \in [0, 1]$, the above vector field $\mathbf{v}_\tau$ defines a gradient flow on $\mathcal{P}(\Omega)$ as $\partial_\tau \mu_\tau + \nabla_{\boldsymbol{\theta}} \cdot (\mathbf{v}_\tau \mu_\tau) = 0$, where $\nabla_{\boldsymbol{\theta}} \cdot \mathbf{a} \triangleq \nabla_{\boldsymbol{\theta}}^\top \mathbf{a}$ for a vector $\mathbf{a}$.*

Function F in Section 3 is lifted to be a functional in the space of probability measures, mapping a probability measure μ to a real value, *i.e.*, $F : \mathcal{P}(\Omega) \to \mathbb{R}$. F is the energy functional of a gradient flow on $\mathcal{P}(\Omega)$. Consequently, it can be shown that $\mathbf{v}_\tau$ in Lemma 1 has the form $\mathbf{v}_\tau = -\nabla_{\mathbf{x}} \frac{\delta F}{\delta \mu_\tau}(\mu_\tau)$ [Ambrosio et al., 2005], where $\frac{\delta F}{\delta \mu_\tau}$ is called the *first variation* of F at μ_τ [Do**u**gan and Nochetto]. Based on this, gradient flows on $\mathcal{P}(\Omega)$ can be written in a form of partial differential equation (PDE) as

$$\partial_\tau \mu_\tau = -\nabla_{\boldsymbol{\theta}} \cdot (\mathbf{v}_\tau \mu_\tau) = \nabla_{\boldsymbol{\theta}} \cdot \left(\mu_\tau \nabla_{\boldsymbol{\theta}} (\frac{\delta F}{\delta \mu_\tau}(\mu_\tau)) \right) . \qquad (3)$$

Intuitively, an energy functional F characterizes the landscape structure of the corresponding manifold, and the gradient flow (3) defines a solution path on this manifold. Usually, by choosing appropriate F, the landscape is convex, *e.g.*, the Itó-diffusion case [Chen et al., 2018]. This provides a theoretical guarantee of optimal convergence of a gradient flow.

[*]We will focus on the convex case, since this is the case for many gradient flows on the space of probability measures, as detailed subsequently.

Remark 1 *WGF provides an alternative means of performing Bayesian sampling: If the WGF* (3) *is designed such that the stationary distribution equals our target distribution, sampling from the target distribution is equivalent to solving the WGF* (3)*.*

4 Thompson Sampling via Optimal Transport

This section describes our proposed Particle-Interactive Thompson sampling (π-TS) framework. We first interpret Thompson sampling as a WGF problem, then propose a specific energy function to design a WGF, and finally propose particle-approximation methods to solve the π-TS problem.

4.1 Thompson Sampling via Wasserstein gradient flows

In Thompson sampling, we describe model uncertainty by imposing distributions on parameters. The optimal parameter distribution (posterior distribution) thus can be solved via a WGF defined on the parameter distribution. Specifically, given a prior distribution $p_0(\boldsymbol{\theta})$ on the model parameters $\boldsymbol{\theta}$ and the set of past observations $\mathcal{D}_t \triangleq \{(\mathbf{x}_i, \mathbf{a}_i, r_i)\}_{i=1}^{t-1}$, Thompson sampling [Thompson, 1933] maintains a posterior distribution of $\boldsymbol{\theta}$ as $p_{\boldsymbol{\theta}} \triangleq p(\boldsymbol{\theta}|\,\mathcal{D}_{t-1}) \propto e^{U(\boldsymbol{\theta})}$, where the potential energy is defined as

$$\begin{aligned} U(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p_0(\boldsymbol{\theta}) \qquad (4) \\ &= \sum_{i=1}^{t} \left(\log p(r_i|\,\mathbf{x}_i, \mathbf{a}_i, \boldsymbol{\theta}) + \frac{1}{t} \log p_0(\boldsymbol{\theta}) \right), \end{aligned}$$

where, and $\mu(\cdot)$ is a neural network. To apply WGFs for posterior approximation in Thompson sampling, a variational (posterior) distribution for $\boldsymbol{\theta}$, denoted as $\mu(\boldsymbol{\theta})$, is learned by solving an appropriate gradient-flow problem. To make the stationary distribution of the WGF consistent with the target posterior distribution, we define an energy functional characterizing the similarity between the current variational distribution and the true distribution $p_{\boldsymbol{\theta}}$ induced by the rewards as:

$$\begin{aligned} F(\mu) &\triangleq \underbrace{-\int U(\boldsymbol{\theta})\mu(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}_{E_1} + \underbrace{\int \mu(\boldsymbol{\theta}) \log \mu(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}_{E_2} \qquad (5) \\ &= \mathsf{KL}\left(\mu \| p_{\boldsymbol{\theta}}\right) \ . \end{aligned}$$

Note E_2 is the energy functional of a pure Brownian motion (*e.g.*, $U(\boldsymbol{\theta}) = 0$ in (5)). According to (3), the first variation of functional E_1 and E_2 can be calculated as:

$$\frac{\delta E_1}{\delta \mu} = -U, \qquad \frac{\delta E_2}{\delta \mu} = \log \mu + 1 \ . \qquad (6)$$

Substituting (6) into (3) yields the specific PDE form of the WGF for Thompson sampling. The energy functional $F(\mu)$ defines a landscape determined by the rewards, whose minimum is obtained at $\mu = p_{\boldsymbol{\theta}}$.

Proposition 2 *For the gradient flow with energy functional defined in* (5)*, μ converges to $p_{\boldsymbol{\theta}}$ in the infinite-time limit.*

4.2 Particle Approximation for Thompson Sampling

To solve the above WGF problem (3), following methods such as those in [Chen et al., 2018], we proposed to use particles, approximating μ with M particles $\{\boldsymbol{\theta}^i\}_{i=1}^M$ as

$$\mu^{(h)} \approx \frac{1}{M} \sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^i} . \qquad (7)$$

where $\delta_{\boldsymbol{\theta}_k}$ is a delta function with a spike at $\boldsymbol{\theta}_k$. Consequently, solving for the optimal μ is equivalent to updating the particles. We investigate two numerical methods to solve (3), with the discrete-gradient-flow (DGF) method and the blob method.

Discrete gradient flow method Discrete gradient flows (DGFs) approximate (3) by discretizing the continuous curve μ_t into a piece-wise linear curve, leading to an iterative optimization problem to solve the intermediate points denoted as $\{\mu_k^h\}_k$, where k denotes the discrete points, and h is referred to as the stepsize parameter. The iterative optimization problem is also known as the Jordan-Kinderleher-Otto (JKO) scheme [Jordan et al., 1998], where for iteration k, $\mu_{k+1}^{(h)}$ is obtained by solving the following optimization problem:

$$\mu_{k+1}^{(h)} = \arg\min_{\mu} \mathsf{KL}\left(\mu \| p_{\boldsymbol{\theta}}\right) + \frac{W_2^2(\mu, \mu_k^{(h)})}{2h} \ . \qquad (8)$$

With particles approximating the μ in (7), the evolution of distributions described by (3) can be approximated with gradient ascent on particles. Specifically, the two terms in (8) can be decomposed as:

$$\begin{aligned} F_1 &\triangleq -\mathbb{E}_\mu[\log p(\boldsymbol{\theta}|\mathcal{D})] + \lambda_1 \mathbb{E}_\mu[\log \mu] \\ F_2 &\triangleq \lambda_2 \mathbb{E}_\mu[\log \mu] + \frac{1}{2h} W_2^2(\mu, \mu_k^{(h)}), \qquad (9) \end{aligned}$$

where $\lambda_1 + \lambda_2 = 1$. According to Liu and Wang [2016], the gradient of the first term can be easily approximated as:

$$\frac{\partial F_1}{\partial \boldsymbol{\theta}_k^i} = \sum_{j=1}^{M} \left[-\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \nabla_{\boldsymbol{\theta}_k^i} U(\boldsymbol{\theta}_k^i) + \nabla_{\boldsymbol{\theta}_k^j} \kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \right],$$

where κ is the kernel function, which typically is the radial basis function (RBF) kernel defined as $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp(-\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2/h)$. For the second term F_2, we can solve the entropy-regularized Wasserstein distance by introducing Lagrangian multipliers as:

$$\begin{aligned}\frac{\partial F_2}{\partial \boldsymbol{\theta}_k^i} &\approx -\frac{\sum_j u_i v_j c_{ij} e^{-\frac{c_{ij}}{\lambda_2}}}{\partial \boldsymbol{\theta}_k^i} \qquad (10)\\ &= \sum_j 2 u_i v_j \left(\frac{c_{ij}}{\lambda_2} - 1\right) \exp^{-\frac{c_{ij}}{\lambda_2}} (\boldsymbol{\theta}_k^i - \boldsymbol{\theta}_{k-1}^j) \ .\end{aligned}$$

where $c_{ij} \triangleq \|\boldsymbol{\theta}^i - \boldsymbol{\theta}^j\|_2^2$. Theoretically, we need to adaptively update Lagrangian multipliers $\{u_i, v_j\}$ to ensure the constraints in (10). In practice, however, we use a fixed scaling factor γ to approximate $u_i v_j$ for the sake of simplicity. The entropy-regularized Wasserstein term F_2 works as a complex force between particles in two ways: *i)* When $\frac{c_{ij}}{\lambda} > 1$, $\boldsymbol{\theta}_k^i$ is pulled close to previous particles $\{\boldsymbol{\theta}_{k-1}^j\}$, with force proportional to $(\frac{c_{ij}}{\lambda} - 1) e^{-c_{ij}/\lambda}$; *ii)* when $\boldsymbol{\theta}^i$ is close enough to a previous particle $\boldsymbol{\theta}_k^j$, *i.e.*, $\frac{c_{ij}}{\lambda} < 1$, $\boldsymbol{\theta}_k^i$ is pushed away, preventing it from collapsing to $\boldsymbol{\theta}_k^j$. Formally, in the k-th iteration, the particles are updated with:

$$\begin{aligned}\boldsymbol{\theta}_{k+1}^i = \boldsymbol{\theta}_k^i &+ \frac{h}{M} \sum_{j=1}^M \left[-\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \nabla_{\boldsymbol{\theta}_k^i} U(\boldsymbol{\theta}_k^i) + \nabla_{\boldsymbol{\theta}_k^j} \kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i)\right] \\ &+ \frac{h}{M} \sum_{j=1}^M \left(\frac{c_{ij}}{\lambda_2} - 1\right) \exp^{-\frac{c_{ij}}{\lambda_2}} (\boldsymbol{\theta}_k^i - \boldsymbol{\theta}_{k-1}^j), \qquad (11)\end{aligned}$$

Blob method Discrete gradient flows apply particle approximation on the discretization form (8). In blob methods, particle approximation is applied directly on the original PDE of the WGF (3). According to [Carrillo et al., 2017], if we define $\mathbf{v}_t(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \frac{\delta(F_1 + F_2)}{\delta \mu}$, (8) directly reduces to the following differential equation system for the particles $\{\boldsymbol{\theta}^i\}$:

$$\mathrm{d}\boldsymbol{\theta}^i = v_t(\boldsymbol{\theta}^i)\mathrm{d}t \ . \qquad (12)$$

Here $\mathbf{v}_t$ can be interpreted as the velocity of a particle in the gradient flow. Consequently, the blob method corresponds to solving (12) numerically with a time-spacing h following the velocity $\mathbf{v}_t$. Under a $\mathcal{H}$-Wasserstein distance metric defined by [Liu, 2017], the velocity $\mathbf{v}_t$ can be calculated as:

$$\mathbf{v}_t(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}' \sim \tilde{\mu}} \left[-\kappa(\boldsymbol{\theta}', \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}'} U(\boldsymbol{\theta}') + \nabla_{\boldsymbol{\theta}'} \kappa(\boldsymbol{\theta}', \boldsymbol{\theta})\right] \ , \qquad (13)$$

where $\tilde{\mu}$ is the empirical particle distribution, *i.e.*, $\tilde{\mu}(\cdot) = \frac{1}{M} \sum_{i=1}^M \delta_{\boldsymbol{\theta}^i}(\cdot)$. By inspecting the representation of $\mathbf{v}_t$ in (13), the last term acts as a repulsive force. Formally, in the k-th iteration, the particles are updated with:

$$\boldsymbol{\theta}_{k+1}^i = \boldsymbol{\theta}_k^i + \frac{h}{M} \sum_{j=1}^M \left[-\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \nabla_{\boldsymbol{\theta}_k^j} U(\boldsymbol{\theta}_k^j) + \nabla_{\boldsymbol{\theta}_k^j} \kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i)\right] , \qquad (14)$$

4.3 Particle-Interactive Thompson Sampling

By applying the above methods to solve the WGF for Thompson sampling, we arrive at the Particle-Interactive Thompson Sampling (π-TS) framework. The pseudocode of π-TS is described in Algorithm 2. In π-TS, the initial particles are drawn from the model prior $p_0(\boldsymbol{\theta})$, and then are updated iteratively via either discrete gradient flow or the blob method to approximate the posterior distributions.

Algorithm 2 Particle-Interactive Thompson Sampling

```
Require: $\mathcal{D}_0 = \emptyset$; initialize particles $\Theta_0 = \{\boldsymbol{\theta}_0^i\}_{i=1}^M$;
1: for $t = 1, 2, \ldots, T$ do
2:     Observe context $\mathbf{x}_t$
3:     Draw $\hat{\boldsymbol{\theta}}_t$ uniformly from $\Theta_t$
4:     Select $a_t \in \arg\max_{\mathbf{a}} m(\mathbf{x}_t, \mathbf{a}; \hat{\boldsymbol{\theta}}_t)$
5:     Observe and receive reward $r_t$
6:     $D_{t+1} = D_t \cup (\mathbf{x}_t, \mathbf{a}_t, r_t)$
7:     Update $\Theta_{t+1}$, according to (11) or (14)
8: end for
```

In theory, use of more particles leads to better performance. However, maintaining a large number of particles is computationally expensive. A balance between performance and computational cost is achieved by choosing a reasonable number of particles. This is investigated in Section 5.2. In contrast with vanilla Thompson sampling, one approximate posterior sample is randomly selected from the particle set Θ_t in each iteration of π-TS to make decisions at time t. Intuitively, the continuous-time flow of posterior distributions governed by (3) describes the evolution of posterior distributions. By discretizing this continuous-time flow, we can iteratively update the posterior distribution. With particle approximation, the distribution evolution is described by evolving particles, which can be updated efficiently by stochastic gradient descent.

5 Experiments

We consider the performance of our proposed π-TS framework in both static scenarios and contextual-bandit problems. Specifically, in Section 5.1 we consider a static scenario, and verify the ability of our framework to sample from multi-mode distributions. We then consider in Section 5.2 contextual bandits with a linear or sparse linear reward function. Finally, we evaluate π-TS on standard real-world benchmarks in Section 5.3. Our implementation is in TensorFlow. All computations were run on a single Tesla P100 GPU, and all results are averaged over 50 realizations. Code for our experiments is available from `https://www.github.com/zhangry868/pi_Thompson_Sampling`.

5.1 Multi-mode Distributions

To provide insight into the representation power of the proposed methods, we demonstrate experimental results in *static* scenarios for multi-mode distribution sampling. We implement the WGF-based methods, *i.e.*, DGF and Blob methods, for sampling from four complex 2D-distributions with unnormalized densities $p(z) \propto \exp[U(z)]$. The detailed functional forms of $U(z)$ are provided in Section A.1. We use 2000 particles to approximate the target distributions, with final results shown in Figure 1. It can be seen that WGF-based algorithms successfully find all the modes and fit the distributions well, demonstrating the advantages of using WGF-based methods for uncertainty estimation.

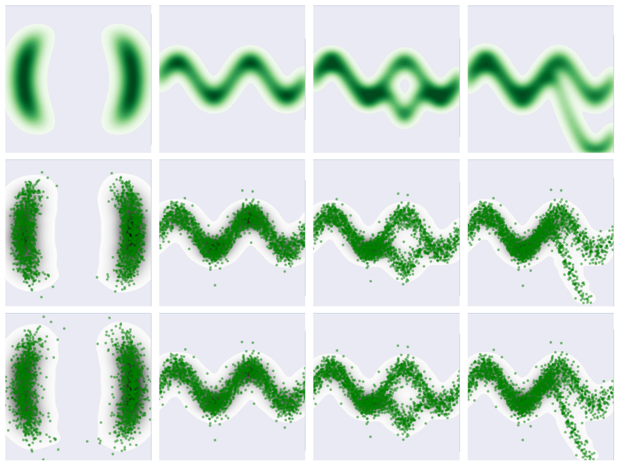

Figure 1: Illustration of different algorithms on toy distributions with 2000 particles. Each column is a distribution case. 1st row: Ground Truth; 2nd row: DGF; 3rd row: Blob.

5.2 Linear and Sparse Linear Contextual Bandits

We consider a contextual bandit scenario where uncertainty estimation is driven by sequential decision making. This is more challenging because in this case the observations $\mathcal{D}$ are no longer i.i.d., leading to larger accumulative error as time proceeds. Poor uncertainty estimation, such as mode-seeking or covering only one of the modes, will lead to approximation error with high regret. This usually happens when the approximated policy keeps selecting a sub-optimal action. We test the proposed method in two reward settings: one with linear rewards and the other with sparse linear rewards [Riquelme et al., 2018]. We compare our method with VI-TS [Urteaga and Wiggins, 2018], neural linear [Riquelme et al., 2018] and Linear-TS [Agrawal and Goyal, 2013] and normalize the cumulative regrets relative to that of the uniform action selection.

Linear Rewards We consider a contextual bandit with $k = 8$ arms and $d = 20$ dimensional context. For a given context $X \sim \mathcal{N}(\mu, \Sigma)$, the reward obtained by pulling arm i follows a linear model $r_{i,X} = X^T\beta_i + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_i^2)$, where $\sigma_i^2 = 0.01i$ The posterior distribution over $\beta_i \in \mathrm{R}^d$ can be computed exactly using the standard Bayesian linear regression formula, denoted as Linear-TS. We set the prior of $\beta \sim \mathcal{N}(0, \lambda\, \mathrm{I}_d)$, with $\lambda = 0.1$. The results in terms of both regret and normalized regret are plotted in Figure 2. The proposed methods, π-TS-Blob and π-TS-DGF, perform almost as well as Linear-TS, the exact model, whereas other methods such as Neural-Linear and VI-TS receive much larger regret. The gap is primarily caused by the approximation error between the exact posterior and approximate posterior. VI-TS shows a higher regret variance. Furthermore, π-TS-Blob and π-TS-DGF are found to perform similarly.

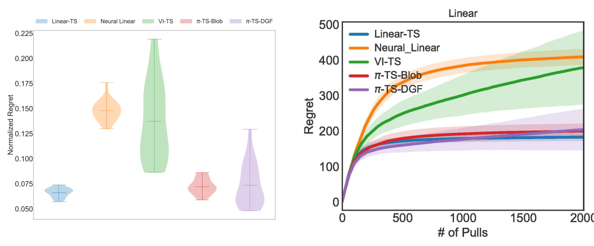

Figure 2: Normalized regret comparison among five methods on linear cases.

Effects of Numbers of Particles

We investigate the influence of the number of particles used on the performance. We consider $M = 1, 5, 20, 50$ particles, where the case of a single particle corresponds to the greedy setting. We use the same model as the above experiments. We here use larger noise $\sigma_i^2 = 0.1$, and pull each arm two times at the initial stage. Figure 3 shows accumulated regret as a function of the number of particles. As expected, the best performance is achieved with the largest number of particles. The performance keeps improving with an increasing number of particles, but the gain becomes insignificant considering the increased computational costs.

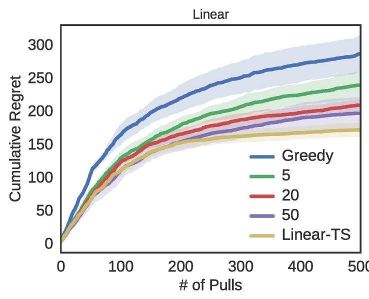

Figure 3: Impact of particle number.

Sparse Linear Rewards In this case, the weight vector $\beta_i^s \in \mathrm{R}^d$ is sparse. Specifically, β_i^s is more sparse than the standard β used above. The reward obtained by pulling arm i follows a sparse linear model is $r_{i,X} = X^T\beta_i^s + \epsilon$, where $\sigma_i^2 = 0.01i$. The results are plotted in Figure 4; much less regret is achieved by

π-TS-DGF and π-TS-Blob, which are comparable to Linear-TS.

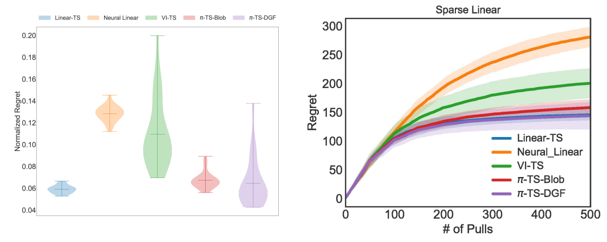

Figure 4: Normalized regret comparison among five methods on sparse linear cases.

5.3 Deep Contextual Bandits

Following the settings of [Riquelme et al., 2018], we evaluate the algorithms on a range of bandit problems created from real-world data: the Statlog, Covertype, Adult, Census, Financial, and Mushroom datasets. These datasets exhibit a broad range of properties: from small to large data sizes, from one dominating action to more homogeneous optimality, and from stochastic to deterministic rewards. Details on the datasets are provided in Table 1. We normalize the cumulative regrets relative to that of the uniform action selection, and plot the box-plot of the final normalized regrets in Figure 6. In Figure 5 we provide the mean (dark curves) and standard derivation (light areas) of regret, along with number of pulls, over 50 realizations.

Table 1: Description of datasets

Dataset	Contexts	Actions
Mushroom	22	2
Statlog	16	7
Covertype	54	7
Financial	21	8
Census	389	9
Adult	94	14

In summary, π-TS outperforms other methods; the performance of Linear-TS is not as good due to its poor representation. Especially with more data observed, it becomes more and more difficult to approximate the exact posterior with the Linear-TS. With features extracted by a neural network, the neural linear approach improves the performance and generally outperforms Linear-TS. Nevertheless, there are some cases where valid features cannot be well extracted by neural networks, leading to poor performance of Neural Linear. Furthermore, VI-TS [Urteaga and Wiggins, 2018] consistently performs poorly, with very high variance. The main cause might be that the underestimated uncertainty leads to poor exploration. Our proposed π-TS outperforms other methods, since it can provide better uncertainty estimation than VI-TS, and endows more

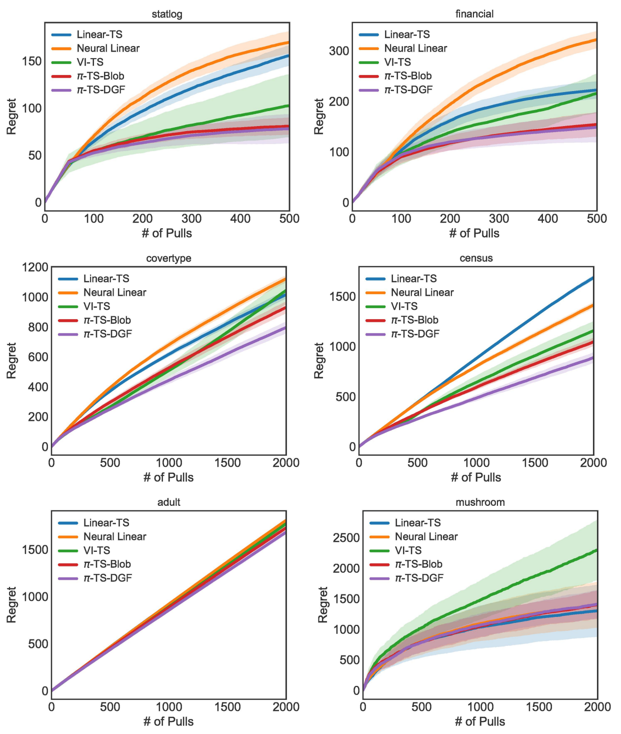

Figure 5: Normalized regret comparison on real-world datasets.

representation power than Linear-TS. Specifically, the performance of π-TS for relatively large datasets is much better than that of other methods.

5.4 Online Response Selection of Dialogue

We consider the response-selection task in the dialog problem, under the contextual bandit framework. We use the Ubuntu Dialogue Corpus (UDC) [Lowe et al., 2015] dataset, a multi-turn-based dialog corpus constructed from Ubuntu chat logs. The dataset contains dialog context-response pairs from real chat logs and each data sample contains three components: the dialog context, ten candidate responses, and their labels (match or non-match). We perform standard pre-processing by replacing named entities with corresponding tags. We use Recall@k for evaluation (k=1, 2, 5). Recall@k measures whether the ranking is considered correct if the matched response is among the top k selections of the 10 candidate responses.

We use a bidirectional LSTM [Schuster and Paliwal, 1997] to encode dialog context and response, where the LSTM state size and output size are both set to 256, and the word-embeddings size is 150. In this setting, the candidate responses are regarded as actions. We use the RMSProp optimizer in the contextual-bandit based response selection model training with a learning rate of 0.001. The bidirectional LSTM encoder is pretrained using data from the UDC training set as an encoder-decoder model. Online bandit learning and

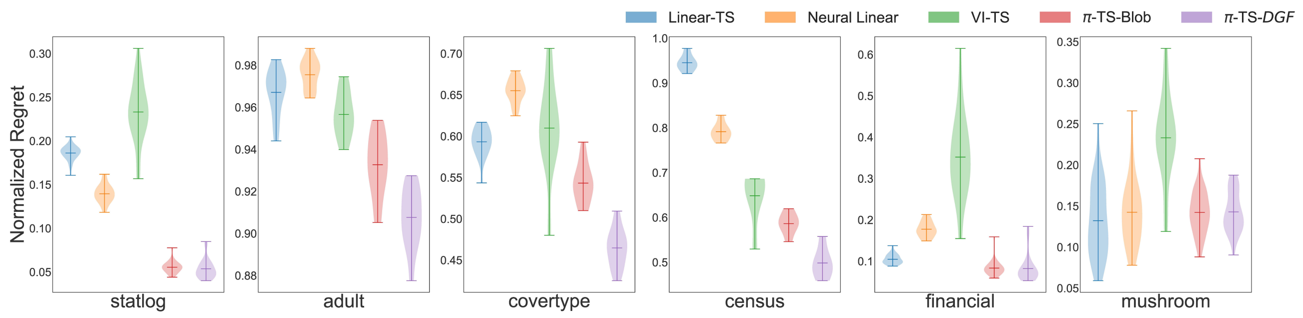

Figure 6: Normalized cumulative regret on real-world datasets.

evaluation is performed using data sampled from the test set.

In each dialog interaction, a context is randomly selected from the UDC test set and provided to the agent. Given the context and candidate responses, the agent returns its best predicted response to the user. The user then provides the agent with feedback to help improve the performance of the agent. If a good response is selected for the contexts, the agent will receive a reward of 1; otherwise, a reward of 0 will be given.

Similar to the setting of [Liu et al., 2017], 2000 context-response pairs are held in the test set for evaluation Liu and Wang [2016], and the rest are used in the online learning by the agent. The results are shown in Table 2.

Table 2: Recall@k evaluation results for the online response selection with k=1, 2 and 5.

	1 in 10 R@1	1 in 10 R@2	1 in 10 R@5
VI-TS	11.5%	21.5%	52.5%
Neural Linear	17.5%	29.5%	63.5%
π-TS-Blob	23.5%	33.5%	66.0%
π-TS-DGF	**26.0%**	**36.5%**	**70.5%**

It is observed that π-TS outperforms other methods; Linear-TS is not applicable in this dataset, and thus it is excluded from the baselines. Moreover, the performance of π-TS-Blob is slightly worse than π-TS-DGF.

6 Related Work

It is difficult in general to calculate exact posteriors in Thompson sampling. Thus it is necessary to efficiently approximate a posterior distribution to make TS scalable for complex models. [Blundell et al., 2015] first used standard variational inference to approximate the posterior of neural networks, *i.e.*, Bayesian neural networks, which were then incorporated into Thompson sampling. Further, [Osband et al., 2016] proposed to use different heads for a deep Q-network to approximate posterior with bootstrap. Inspired by [Osband et al., 2016], Lu and Van Roy [2017] proposed ensemble sampling, which uses a set of particles to approximate a posterior distribution. These particles are updated independently with stochastic gradient descent, without a convergence guarantee, in terms of posterior-approximation convergence. Similarly, weighted bootstrap [Vaswani et al., 2018] uses random weights performed on the likelihood to mimic the bootstrap, which is connected to TS. [Riquelme et al., 2018] built a benchmark to evaluate deep Bayesian bandits, and especially recommended the neural linear method, which uses a deep neural network to extract features and perform linear Thompson sampling based on these features. Similar to neural linear, [Azizzadenesheli et al., 2018] replaced the final layer of a deep neural network with Bayesian logistic regression for deep Q-networks, which greatly boosted the performance on Atari benchmarks. Zhang et al. [2018b] firstly investigate particle-based Thompson sampling in contextual bandits settings. Zhang et al. [2018a] places policy optimization into the space of probability measures, and interpret it as Wasserstein gradient flows. In this work, we provide a distribution optimization perspective to understand the posterior approximation, and propose efficient algorithms to approximate posterior distributions in Thompson sampling. This work can be regarded as the counterpart of [Zhang et al., 2018b] for value-based methods.

7 Conclusion

We have proposed a scalable Thompson sampling framework, π-TS, for posterior estimation in Thompson sampling. We approximate the posterior distribution without an explicit-form variational distribution assumption, which leverages more powerful uncertainty estimation. Importantly, our methods can be applied on large-scale problems with complex models, such as neural networks. Specifically, π-TS approximates a distribution by defining gradient flows on the space of probability measures, and uses particles for approximation. Extensive experiments are conducted, demonstrating the effectiveness and efficiency of our proposed π-TS framework. Interesting future work includes designing more practically efficient variants of π-TS, and developing theory to study general regret bounds of the algorithms, as was done in [Lu and Van Roy, 2017].

Acknowledgments This research was supported in part by DARPA, DOE, NIH, NSF and ONR.

References

Rajeev Agrawal. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 1995.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, 2013.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures.* Lectures in Mathematics ETH Zürich, 2005.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 2002.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. *arXiv:1802.04412*, 2018.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *ICML*, 2015.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 2012.

Manfredo Perdigão do Carmo. *Riemannian geometry.* Birkhäuser, 1992.

J. A. Carrillo, K. Craig, and F. S. Patacchini. A blob method for diffusion. (arXiv:1709.09195), 2017.

Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. Boltzmann exploration done right. In *NIPS*, 2017.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *NIPS*, 2011.

Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. In *UAI*, 2018.

Günay Do*u*gan and Ricardo H Nochetto. First variation of the general curvature-dependent surface energy. *ESAIM: Mathematical Modelling and Numerical Analysis.*

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIMA*, 1998.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010.

Bing Liu, Tong Yu, Ian Lane, and Ole J Mengshoel. Customized nonlinear bandits for online response selection in neural conversation models. In *AAAI*, 2017.

Qiang Liu. Stein variational gradient descent as gradient flow. In *NIPS*, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NIPS*, 2016.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL*, 2015.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In *NIPS*, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *NIPS*, 2016.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *ICLR*, 2018.

Jim Rulla. Error analysis for implicit approximations to solutions to cauchy problems. *SINUM*, 1996.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *JMLR*, 2016.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 2018.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *TSP*, 1997.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings*. Elsevier, 1990.

Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning.* MIT press Cambridge, 1998.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

Iñigo Urteaga and Chris Wiggins. Variational inference for the multi-armed contextual bandit. In *AISTATS*, 2018.

Sharan Vaswani, Branislav Kveton, Zheng Wen, Anup Rao, Mark Schmidt, and Yasin Abbasi-Yadkori. New insights into bootstrapping for bandits. *arXiv:1805.09793*, 2018.

C. Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. 2018a.

Ruiyi Zhang, Chunyuan Li, Changyou Chen, and Lawrence Carin. Learning structural weight uncertainty for sequential decision-making. 2018b.