Reward-Biased Maximum Likelihood Estimation for Neural Contextual Bandits: A Distributional Learning Perspective

Yu-Heng Hung, Ping-Chun Hsieh

Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
hungyh.cs08@nycu.edu.tw, pinghsieh@nycu.edu.tw

Abstract

Reward-biased maximum likelihood estimation (RBMLE) is a classic principle in the adaptive control literature for tackling explore-exploit trade-offs. This paper studies the neural contextual bandit problem from a distributional perspective and proposes NeuralRBMLE, which leverages the likelihood of surrogate parametric distributions to learn the unknown reward distributions and thereafter adapts the RBMLE principle to achieve efficient exploration by properly adding a reward-bias term. NeuralRBMLE leverages the representation power of neural networks and directly encodes exploratory behavior in the parameter space, without constructing confidence intervals of the estimated rewards. We propose two variants of NeuralRBMLE algorithms: The first variant directly obtains the RBMLE estimator by gradient ascent, and the second variant simplifies RBMLE to a simple index policy through an approximation. We show that both algorithms achieve order-optimality. Through extensive experiments, we demonstrate that the NeuralRBMLE algorithms achieve comparable or better empirical regrets than the state-of-the-art methods on real-world datasets with non-linear reward functions.

1 Introduction

Efficient exploration has been a fundamental challenge in sequential decision-making in unknown environments. As a classic principle originally proposed in the stochastic adaptive control literature for solving unknown Markov decision processes (MDPs), *Reward-Biased Maximum Likelihood Estimation (RBMLE)* learns an optimal policy by alternating between estimating the unknown model parameters in an "exploratory" manner and applying the optimal control law based on the estimated parameters (Kumar and Becker 1982; Borkar 1990; Campi and Kumar 1998; Prandini and Campi 2000). Specifically, to resolve the inherent issue of insufficient exploration of maximum likelihood estimation (MLE), RBMLE enforces exploration by incorporating into the likelihood function a bias term in favor of those model parameters that correspond to higher long-term average rewards. This generic exploration scheme has been shown to asymptotically attain the optimal long-term average reward (Kumar and Becker 1982).

Recently, the RBMLE principle has been adapted to optimize the regrets in stochastic bandit problems, including the classic non-contextual multi-armed bandit problems (Liu et al. 2020), the contextual bandit problems with generalized linear reward functions (Hung et al. 2021) and model-based reinforcement learning for finite MDPs (Mete et al. 2021). Moreover, RBMLE has been shown to achieve order-optimal finite-time regret bounds and competitive empirical regret performance in the above settings. Despite the recent progress, the existing RBMLE bandit algorithms, as well as their regret guarantees, rely heavily on the structural assumptions, such as the absence of contextual information in (Liu et al. 2020) and linear realizability in (Hung et al. 2021), and hence are not readily applicable to various real-world bandit applications with more complex reward structures, such as recommender systems and clinical trials.

Motivated by the competitive performance of the RBMLE principle in the bandit problems mentioned above, this paper takes one step further to study RBMLE in contextual bandits with general reward functions. To unleash the full potential of RBMLE in contextual bandits, we propose NeuralRBMLE, which leverages the representation power of neural networks to learn the unknown reward distributions. Compared to the existing regression-based neural bandit algorithms (Zhou, Li, and Gu 2020; Zhang et al. 2021), NeuralRBMLE presents its novel technical insights into addressing its own salient technical challenges:

- *A new perspective for neural contextual bandits* – The main idea of NeuralRBMLE is to leverage the likelihood of the parametric distributions to learn the unknown reward distributions and thereafter achieve efficient exploration through a reward-bias term. With that said, NeuralRBMLE takes a distributional viewpoint toward understanding the power of neural networks for contextual bandits, and this viewpoint is fundamentally different from the existing regression-based neural bandit methods (Zhou, Li, and Gu 2020; Zhang et al. 2021), which focus on learning the mean rewards. Notably, our attempt of NeuralRBMLE resonates with the recent progress in *distributional methods* in deep RL (Bellemare, Dabney, and Munos 2017), which leverages the representation power of neural networks to learn the value distributions.
- *Model misspecification of reward distributions* – NeuralRBMLE is a distributional approach that leverages the machinery of MLE. The standard use of MLE requires the knowledge of the likelihood function of the stochas-

tic rewards, whose distributions are commonly unknown in practical bandit problems. Thus, given that we make no parametric assumption about the underlying true reward distributions, NeuralRBMLE needs to rely on a *surrogate* likelihood function that may not match the true likelihood. This naturally leads to a *model misspecification* issue, which could have a significant effect on the regret. Despite this, we rigorously show that NeuralRBMLE enjoys favorable regret bounds under the surrogate likelihood of the exponential families.

- *Identifying and resolving the compatibility dilemma of RBMLE with neural tangent kernel (NTK)* – To establish the regret bounds of NeuralRBMLE, we leverage the technique of NTK (Jacot, Gabriel, and Hongler 2018), which underpins that the output of a wide neural network is approximately linear in its parameters. Notably, to enable NTK approximation for RBMLE, we identify and tackle one fundamental *chicken-and-egg dilemma*: The effect of reward bias on the learned neural network parameters can only be quantified in the NTK regime, but the NTK regime is valid only if the perturbation effect of reward bias is already characterized. We address this dilemma by presenting a novel induction argument over time. This is one major novelty of the regret analysis for NeuralRBMLE, compared to its linear bandit counterpart (Hung et al. 2021).
- *Three-way tradeoff due to reward bias* – In the RBMLE principle, the reward bias determines the exploration-exploitation tradeoff (larger reward bias indicates more exploration) and needs to be chosen carefully to achieve low regret. Moreover, as described above, the reward bias needs to be properly configured to enable NTK approximation. Hence, the choice of reward bias reflects an inherent *three-way trade-off* in NeuralRBMLE: exploration, exploitation, and ability of NTK approximation. This three-way trade-off is one unique challenge of NeuralRBMLE, compared to the existing neural bandit methods (Zhou, Li, and Gu 2020; Zhang et al. 2021).
- *The effect of inexact maximizers of RBMLE on regret* – Different from RBMLE for linear bandits (Hung et al. 2021), which relies on the exact maximizer of RBMLE, NeuralRBMLE uses gradient ascent to approximate the true RBMLE estimator during training, and the resulting approximation error should be considered and precisely characterized in the regret analysis.

Moreover, all of the above issues are tightly coupled with each other, and this renders the regret analysis of NeuralRBMLE even more challenging. Despite the above challenges, this paper addresses the above issues rigorously by extending the RBMLE principle to the neural reward function approximation and proposing the first RBMLE bandit algorithm with regret guarantees for the contextual bandit problems without the linear realizability assumption.

We highlight the main contributions as follows:

- We propose NeuralRBMLE, which provides a new distributional perspective for neural contextual bandit problem. We first present a prototypic NeuralRBMLE algorithm that enjoys an index form by incorporating a surrogate likelihood function and a proper reward-bias term. We then propose two practical approaches, namely NeuralRBMLE-GA and NeuralRBMLE-PC, to substantiate the prototypic NeuralRBMLE algorithm.
- We formally establish the regret bounds for the two practical NeuralRBMLE algorithms in the NTK regime. Through regret analysis, we provide an affirmative answer to the compatibility of RBMLE with NTK. Moreover, we fully resolve the model misspecification issue and validate the flexibility in using any exponential family distribution as the surrogate likelihood function for NeuralRBMLE-GA, thereby opening up a whole new family of neural bandit algorithms. This serves as an additional degree of algorithmic freedom in achieving low empirical regret. For NeuralRBMLE-PC, we characterize the interplay between the reward-bias term, the regret, and the condition of the parameters of the neural network.
- We evaluate NeuralRBMLE and other benchmark algorithms through extensive simulations on various benchmark real-world datasets. The simulation results show that NeuralRBMLE achieves comparable or better empirical regret performance than the benchmark methods. RBMLE also exhibits better robustness with a smaller standard deviation of final regret compared to other benchmark methods. Moreover, unlike NeuralUCB and NeuralTS, NeuralRBMLE-GA does not require computing the inverse of a high-dimensional matrix, which is computationally expensive.

Notations. Throughout this paper, for any positive integer K, we use $[K]$ as a shorthand for the set $\{1, \cdots, K\}$. We use $\|\cdot\|_2$ to denote the L_2-norm of a vector. We use boldface fonts for vectors and matrices throughout the paper. Moreover, we use $\mathbf{0}$ and $\mathbf{I}$ to denote the zero matrices and the identity matrices, respectively.

2 Problem Formulation

In this section, we formally describe the neural contextual bandit problem considered in this paper.

2.1 Contextual Bandits with General Rewards

We consider the stochastic K-armed contextual bandit problem, where the total number of rounds T is known[1]. At each decision time $t \in [T]$, the K context vectors $\{\mathbf{x}_{t,a} \in \mathbb{R}^d \mid a \in [K]\}$, which capture the feature information about the arms, are revealed to the learner. Without loss of generality, we assume that $\|\mathbf{x}_{t,a}\|_2 \leq 1$, for all $t \in [T]$ and for all $a \in [K]$. Given the contexts, the learner selects an arm $a_t \in [K]$ and obtains the corresponding random reward r_{t,a_t}. For ease of notation, we define (i) $\mathbf{x}_t := \mathbf{x}_{t,a_t}$, (ii) $r_t := r_{t,a_t}$, (iii) $\mathcal{F}_t := (\mathbf{x}_1, a_1, r_1, \ldots, \mathbf{x}_t)$ as the observation history up to the beginning of time t, (iv) $a_t^* := \text{argmax}_{a \in [K]} E[r_{t,a}|\mathcal{F}_t]$, and (v) $r_t^* := r_{t,a_t^*}$. The goal of

[1]The assumption of a known horizon is mild as one could apply the standard *doubling trick* to convert a horizon-dependent algorithm to an anytime one (Lattimore and Szepesvári 2020).

the learner is to minimize the *pseudo regret* as

$$\mathcal{R}(T) := \mathbb{E}\Big[\sum_{t=1}^{T}(r_t^* - r_t)\Big] \tag{1}$$

In the neural contextual bandit problem, the random reward at each time t takes the form of $r_t = h(\mathbf{x}_t) + \epsilon_t$, where $h : \mathbb{R}^d \to [0,1]$ is an unknown reward function and ϵ_t is a ν-sub-Gaussian noise conditionally independent of all the other rewards in the past given the context $\mathbf{x}_t$ and satisfying $\mathbb{E}[\epsilon_t|\,\mathbf{x}_t] = 0$, and the reward function $h(\cdot)$ is approximated by a neural network through training. Compared to the generalized linear bandit model, here we assume no special structure (e.g., linearity or convexity) on the reward function, except that the reward signal is bounded in $[0,1]$.

2.2 Neural Function Approximation for Rewards

In this paper, we leverage a neural network to learn the reward distributions. Let $L \geq 2$ be the depth of this neural network, $\sigma(\cdot) = \max\{\cdot, 0\}$ be the Rectified Linear Unit (ReLU) activation function, and m_l be the width of the l-th hidden layer, for $l \in [L-1]$. We also let $m_0 = d$ and $m_L = 1$. Let $\mathbf{W}_l \in \mathbb{R}^{m_l \times m_{l-1}}$ denote the weight matrix of the l-th layer of the neural network, for $l \in [L]$. For ease of exposition, we focus on the case where $m_l = m$, for all $l \in [L-1]$. For ease of notation, we define $\boldsymbol{\theta} := [\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_L)^\top] \in \mathbb{R}^p$, where $p = m + md + m^2(L-1)$ denotes the total number of parameters of the neural network. Let $f(\mathbf{x};\boldsymbol{\theta})$ denote the output of the neural network with parameters $\boldsymbol{\theta}$ and input $\mathbf{x}$, i.e.,

$$f(\mathbf{x};\boldsymbol{\theta}) := \sqrt{m} \cdot \mathbf{W}_L\, \sigma(\mathbf{W}_{L-1}\, \sigma(\mathbf{W}_{L-2} \ldots \sigma(\mathbf{W}_1\, \mathbf{x}))). \tag{2}$$

$f(\mathbf{x};\boldsymbol{\theta})$ serves as the parameter of the learned reward distributions. Let $\boldsymbol{\theta}_0$ be the initial model parameters selected by the following random initialization steps[2]: (i) For $l \in [L-1]$, let $\mathbf{W}_l$ take the form of $\mathbf{W}_l = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{pmatrix}$, where each entry of $\mathbf{W}$ is drawn independently from $\mathcal{N}(0, 2/m)$. (ii) For the output layer, let $\mathbf{W}_L$ take the form of $\mathbf{W}_L = (\mathbf{w}^\top, -\mathbf{w}^\top)$, where each entry of $\mathbf{w}$ is drawn independently from $\mathcal{N}(0, 1/m)$.

3 An Overview of the RBMLE Principle

In this section, we review the generic RBMLE principle in the context of adaptive control for maximizing the long-term average reward of an unknown dynamic system. Consider a discrete-time MDP with a state space $\mathcal{S}$, an action space $\mathcal{A}$, and unknown transition dynamics as well as a reward function that are both dependent on the unknown true parameter $\boldsymbol{\theta}_*$ belonging to some known set Θ. For ease of notation, for any $\boldsymbol{\theta} \in \Theta$[3], we denote the transition probabilities under $\boldsymbol{\theta}$ by $p(s_t, s_{t+1}, a_t; \boldsymbol{\theta}) := \text{Prob}(s_{t+1}|s_t, a_t)$, where p is a probability function parameterized by $\boldsymbol{\theta}$, $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and the action taken at time t. Let $J(\pi;\boldsymbol{\theta})$ be the long-term average reward under policy $\pi : \mathcal{S} \to \mathcal{A}$. We let $J^*(\boldsymbol{\theta}) := \max_\pi J(\pi;\boldsymbol{\theta})$ denote the optimal long-term average reward for any $\boldsymbol{\theta} \in \Theta$ and use $\pi^* := \text{argmax}_\pi\, J(\pi;\boldsymbol{\theta}_*)$ to denote an optimal policy for $\boldsymbol{\theta}_*$.

- **Closed-loop identification issue:** Originally proposed by (Mandl 1974), the *certainty equivalent* (CE) method addresses the optimal control of an unknown dynamic system by first finding the MLE of the true parameter and then following an optimal policy with respect to the MLE. Specifically, the MLE of the true parameter $\boldsymbol{\theta}_*$ at each time t can be derived as

$$\boldsymbol{\theta}_t^{\text{MLE}} := \underset{\boldsymbol{\theta}\in\Theta}{\text{argmax}} \prod_{i=1}^{t-1} p(s_i, s_{i+1}, a_i; \boldsymbol{\theta}). \tag{3}$$

  Let $\pi_t^{\text{MLE}} := \text{argmax}_\pi\, J(\pi, \boldsymbol{\theta}_t^{\text{MLE}})$ denote an optimal policy for the system with parameter $\boldsymbol{\theta}_t^{\text{MLE}}$. Then, it was shown in (Kumar and Becker 1982) that under the sequence of policies $\{\pi_t^{\text{MLE}}\}$, the sequence of maximum likelihood estimates $\{\boldsymbol{\theta}_t^{\text{MLE}}\}$ converges to some estimate $\boldsymbol{\theta}_\infty^{\text{MLE}}$ in the limit such that for all pairs of $s, s' \in S$,

$$p(s, s', \pi_\infty^{\text{MLE}}(s); \boldsymbol{\theta}_\infty^{\text{MLE}}) = p(s, s', \pi_\infty^{\text{MLE}}(s); \boldsymbol{\theta}_*), \tag{4}$$

  where $\pi_\infty^{\text{MLE}} := \text{argmax}_\pi\, J(\pi, \boldsymbol{\theta}_\infty^{\text{MLE}})$ is an optimal policy for $\boldsymbol{\theta}_\infty^{\text{MLE}}$. Notably, (4) is typically known as the "closed-loop identification" property, which indicates that under the policy π_∞^{MLE}, the transition probabilities can be correctly identified only in a "closed-loop" manner. As a result, under the CE approach, it is *not* guaranteed that all the transition probabilities are correctly estimated, and therefore the policy π_∞^{MLE} is not necessarily optimal for the true parameter $\boldsymbol{\theta}_*$.
- **The inherent bias resulting from MLE:** The above key insight about the CE approach can be made more explicit (Kumar and Becker 1982) by

$$J(\pi_\infty^{\text{MLE}}; \boldsymbol{\theta}_\infty^{\text{MLE}}) = J(\pi_\infty^{\text{MLE}}; \boldsymbol{\theta}_*) \leq J(\pi^*; \boldsymbol{\theta}_*), \tag{5}$$

  where the first equality in (5) follows from (4). As $J(\pi_\infty^{\text{MLE}}; \boldsymbol{\theta}_\infty^{\text{MLE}}) \equiv J^*(\boldsymbol{\theta}_\infty^{\text{MLE}})$ and $J(\pi^*; \boldsymbol{\theta}_*) \equiv J^*(\boldsymbol{\theta}_*)$, (5) indicates that the estimates under CE suffer from an inherent *bias* that favors the parameters with *smaller* optimal long-term average rewards than $\boldsymbol{\theta}_*$.
- **Adding a reward-bias term for correcting the inherent bias of MLE.** To counteract this bias, (Kumar and Becker 1982) proposed the RBMLE approach, which directly multiplies the likelihood by an additional reward-bias term $J^*(\boldsymbol{\theta})^{\alpha(t)}$ with $\alpha(t) > 0, \alpha(t) \to \infty, \alpha(t) = o(t)$, with the aim of encouraging exploration over those parameters $\boldsymbol{\theta}$ with a potentially larger optimal long-term average reward. That is, the parameter estimate under RBMLE is

$$\boldsymbol{\theta}_t^{\text{RBMLE}} := \underset{\boldsymbol{\theta}\in\Theta}{\text{argmax}} \Big\{ J^*(\boldsymbol{\theta})^{\alpha(t)} \prod_{i=1}^{t-1} p(s_i, s_{i+1}, a_i; \boldsymbol{\theta}) \Big\}. \tag{6}$$

  Accordingly, the policy induced by RBMLE at each t is $\pi_t^{\text{RBMLE}} := \text{argmax}_\pi\, J(\pi; \boldsymbol{\theta}_t^{\text{RBMLE}})$. It has been shown

[2] The initialization steps are the same as those in (Cao and Gu 2019; Zhou, Li, and Gu 2020).

[3] To make the connection between RBMLE and neural bandits explicit, in Section 3 we slightly abuse the notation $\boldsymbol{\theta}$ to denote the parameters of the dynamical system.

in (Kumar and Becker 1982) that RBMLE successfully corrects the inherent bias and converges to the optimal policy π^* through the following steps: (i) Since $\alpha(t) = o(t)$, the effect of the reward-bias term becomes negligible compared to the likelihood term for large t. Hence, the sublinearity of $\alpha(t)$ leads to diminishing exploration and thereby preserves the convergence property similar to that of MLE. As a result, both the limits $\boldsymbol{\theta}_\infty^{\text{RBMLE}} := \lim_{t\to\infty} \boldsymbol{\theta}_t^{\text{RBMLE}}$ and $\pi_\infty^{\text{RBMLE}} := \lim_{t\to\infty} \pi_t^{\text{RBMLE}}$ exist, and the result similar to (5) still holds under RBMLE, i.e.,

$$J(\pi_\infty^{\text{RBMLE}}; \boldsymbol{\theta}_\infty^{\text{RBMLE}}) \leq J(\pi^*; \boldsymbol{\theta}_*). \tag{7}$$

(ii) Given that $\alpha(t) \to \infty$, the reward-bias term $J^*(\boldsymbol{\theta})^{\alpha(t)}$, which favors those parameters with higher rewards, remains large enough to undo the inherent bias of MLE. As a result, RBMLE achieves

$$J(\pi_\infty^{\text{RBMLE}}, \boldsymbol{\theta}_\infty^{\text{RBMLE}}) \geq J(\pi^*, \boldsymbol{\theta}^*), \tag{8}$$

as proved in (Kumar and Becker 1982, Lemma 4). By (7)-(8), we know that the delicate choice of $\alpha(t)$ ensures $\pi_\infty^{\text{RBMLE}}$ is an optimal policy for the true parameter $\boldsymbol{\theta}_*$.

Note that the above optimality result implies that RBMLE achieves a sublinear regret, but without any further characterization of the regret bound and the effect of the bias term $\alpha(t)$. In this paper, we adapt the RBMLE principle to neural contextual bandits and design two practical bandit algorithms with regret guarantees.

4 RBMLE for Neural Contextual Bandits

In this section, we present how to adapt the generic RBMLE principle described in Section 3 to the neural bandit problem and propose the NeuralRBMLE algorithms.

4.1 A Prototypic NeuralRBMLE Algorithm

By leveraging the RBMLE principle in (6), we propose to adapt the parameter estimation procedure for MDPs in (6) to neural bandits through the following modifications:

- **Likelihood functions via surrogate distributions:** For each time $t \in [T]$, let $\ell^\dagger(\mathcal{F}_t; \boldsymbol{\theta})$ denote the log-likelihood of the observation history $\mathcal{F}_t$ under a neural network parameter $\boldsymbol{\theta}$. Notably, different from the likelihood of state transitions in the original RBMLE in (6), here $\ell^\dagger(\mathcal{F}_t; \boldsymbol{\theta})$ is meant to capture the statistical behavior of the received rewards given the contexts. However, one main challenge is that the underlying true reward distributions may not have a simple parametric form and are unknown to the learner. To address this challenge, we use the log-likelihood of *canonical exponential family distributions* as a surrogate for the true log-likelihood. Specifically, the surrogate log-likelihood[4] is chosen as $\log p(r_s | \mathbf{x}_s; \boldsymbol{\theta}) = r_s f(\mathbf{x}_s; \boldsymbol{\theta}) - b(f(\mathbf{x}_s; \boldsymbol{\theta}))$, where $b(\cdot) : \mathbb{R} \to \mathbb{R}$ is a known strongly convex and smooth function with $L_b \leq b''(z) \leq U_b$, for all $z \in \mathbb{R}$. Note that the above $\log p(r_s | \mathbf{x}_s; \boldsymbol{\theta})$ is used only for arm selection under NeuralRBMLE, and we do not impose any distributional assumption on the rewards other than sub-Gaussianity. Hence, $\ell^\dagger(\mathcal{F}_t; \boldsymbol{\theta})$ can be written as

$$\ell^\dagger(\mathcal{F}_t; \boldsymbol{\theta}) := \sum_{s=1}^{t-1} \big(r_s f(\mathbf{x}_s; \boldsymbol{\theta}) - b(f(\mathbf{x}_s; \boldsymbol{\theta})) \big). \tag{9}$$

  For example, one candidate choice for (9) is using Gaussian likelihood with $b(z) = z^2/2$.
- **Reward-bias term:** We consider one natural choice $(\max_{a\in[K]} f(\mathbf{x}_{t,a}; \boldsymbol{\theta}))^{\alpha(t)}$, which provides a bias in favor of the parameters $\boldsymbol{\theta}$ with larger estimated rewards.
- **Regularization:** As the RBMLE procedure requires a maximization step, we also incorporate into the reward-biased likelihood a quadratic regularization term $\frac{m\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2$ with $\lambda > 0$, as typically done in training neural networks. For ease of notation, we define

$$\ell_\lambda^\dagger(\mathcal{F}_t; \boldsymbol{\theta}) := \ell^\dagger(\mathcal{F}_t; \boldsymbol{\theta}) - \frac{m\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2. \tag{10}$$

Based on the RBMLE principle in (6) and the design in (9)-(10), at each time t, the learner under NeuralRBMLE selects an arm that maximizes $f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_t^\dagger)$, where

$$\boldsymbol{\theta}_t^\dagger := \operatorname*{argmax}_{\boldsymbol{\theta}} \big\{ \ell_\lambda^\dagger(\mathcal{F}_t; \boldsymbol{\theta}) + \alpha(t) \max_{a\in[K]} f(\mathbf{x}_{t,a}; \boldsymbol{\theta}) \big\}. \tag{11}$$

Inspired by (Hung et al. 2021), we can further show that NeuralRBMLE can be simplified to an *index strategy* by interchanging the two max operations in (11). Now we are ready to present the prototypic NeuralRBMLE algorithm as follows: At each time t,

1. Define the *arm-specific* RBMLE estimators

$$\boldsymbol{\theta}_{t,a}^\dagger := \operatorname*{argmax}_{\boldsymbol{\theta}} \big\{ \ell_\lambda^\dagger(\mathcal{F}_t; \boldsymbol{\theta}) + \alpha(t) f(\mathbf{x}_{t,a}; \boldsymbol{\theta}) \big\}. \tag{12}$$

2. Accordingly, for each arm, we construct an index as

$$\mathcal{I}_{t,a}^\dagger := \ell_\lambda^\dagger(\mathcal{F}_t; \boldsymbol{\theta}_{t,a}^\dagger) + \alpha(t) f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t,a}^\dagger). \tag{13}$$

Then, it can be shown that the policy induced by the NeuralRBMLE in (11) is equivalent to an index strategy which selects an arm with the largest $\mathcal{I}_{t,a}^\dagger$ at each time t. The proof is similar to that in (Hung et al. 2021) and provided in Appendix E.1 for completeness.

Remark 1. Note that in (10)-(13) we use exponential family distributions as the surrogate likelihood functions for RBMLE. When the reward distributions are unknown, one could simply resort to some commonly-used distributions, such as the Gaussian likelihood function. On the other hand, when additional structures of the reward distributions are known, this design also enables the flexibility of better matching the true likelihood and the surrogate likelihood. For example, in the context of logistic bandits, the rewards are known to be binary, and hence one can apply the Bernoulli likelihood to obtain the corresponding NeuralRBMLE estimator.

[4] For brevity, we ignore the normalization function of the canonical exponential families since this term depends only on r_t and is independent of $\boldsymbol{\theta}$.

Algorithm 1 NeuralRBMLE-GA

Input: $\alpha(t)$, $\zeta(t)$, λ, f, $\boldsymbol{\theta}_0$, η, J.
Initialization: $\{\widetilde{\boldsymbol{\theta}}_{0,i}^{\dagger})\}_{i=1}^{K} \leftarrow \boldsymbol{\theta}_0$.
for $t = 1, 2, \cdots$ **do**
  Observe all the contexts $\{\mathbf{x}_{t,a}\}_{1 \le a \le K}$.
  for $a = 1, \cdots, K$ **do**
    Set $\widetilde{\boldsymbol{\theta}}_{t,a}^{\dagger}$ to be the output of J-step gradient ascent with step size η for maximizing $\ell_{\lambda}^{\dagger}(\mathcal{F}_t; \boldsymbol{\theta}) + \alpha(t) f(\mathbf{x}_{t,a}; \boldsymbol{\theta})$.
  end for
  Choose $a_t = \operatorname{argmax}_a \left\{\ell_{\lambda}^{\dagger}(\mathcal{F}_t; \widetilde{\boldsymbol{\theta}}_{t,a}^{\dagger}) + \alpha(t)\zeta(t) f(\mathbf{x}_{t,a}; \widetilde{\boldsymbol{\theta}}_{t,a}^{\dagger})\right\}$ and obtain reward r_t.
end for

Algorithm 2 NeuralRBMLE-PC

Input: $\alpha(t)$, λ, f, $\boldsymbol{\theta}_0$, η, J.
Initialization: $\mathbf{Z}_0 \leftarrow \lambda \mathbf{I}$, $\widehat{\boldsymbol{\theta}}_0 \leftarrow \boldsymbol{\theta}_0$.
for $t = 1, 2, \cdots$ **do**
  Observe all the contexts $\{\mathbf{x}_{t,a}\}_{1 \le a \le K}$.
  for $a = 1, \cdots, K$ **do**
    $\bar{\boldsymbol{\theta}}_{t,a} \leftarrow \widehat{\boldsymbol{\theta}}_t + \frac{\alpha(t)}{m} \cdot \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \widehat{\boldsymbol{\theta}}_t)$.
  end for
  Choose $a_t = \operatorname{argmax}_a \{f(\mathbf{x}_{t,a}; \bar{\boldsymbol{\theta}}_{t,a})\}$ and obtain reward r_t.
  Set $\widehat{\boldsymbol{\theta}}_t$ as the output of J-step gradient ascent with step size η for maximizing $\ell_{\lambda}^{\dagger}(\mathcal{F}_t; \boldsymbol{\theta})$.
  $\mathbf{Z}_t \leftarrow \mathbf{Z}_{t-1} + \mathbf{g}(\mathbf{x}_t; \widehat{\boldsymbol{\theta}}_t)\mathbf{g}(\mathbf{x}_t; \widehat{\boldsymbol{\theta}}_t)^{\top}/m$.
end for

Remark 2. The surrogate likelihood function in (9) follows the similar philosophy as that for the generalized linear bandits in (Hung et al. 2021). Despite the similarity, one fundamental difference is that the objective function of NeuralRBMLE is no longer concave in $\boldsymbol{\theta}$. In spite of this, in Section 5 we show that the practical algorithms derived from NeuralRBMLE still enjoy favorable regret bounds with the help of the neural tangent kernel.

4.2 Practical NeuralRBMLE Algorithms

One major challenge in implementing the prototypic NeuralRBMLE algorithm in (12)-(13) is that the exact maximizer $\boldsymbol{\theta}_{t,a}^{\dagger}$ in (12) can be difficult to obtain since the maximization problem of (12), which involves the neural function approximator $f(\mathbf{x}; \boldsymbol{\theta})$, is non-convex in $\boldsymbol{\theta}$. In this section, we proceed to present two practical implementations of NeuralRBMLE algorithm.

- **NeuralRBMLE by gradient ascent (NeuralRBMLE-GA):** To solve the optimization problem in (12), one natural approach is to apply gradient ascent to obtain an approximator $\widetilde{\boldsymbol{\theta}}_{t,a}^{\dagger}$ for $\boldsymbol{\theta}_{t,a}^{\dagger}$, for each arm. The pseudo code of NeuralRBMLE-GA is provided in Algorithm 1[5]. Given the recent progress on the neural tangent kernel of neural networks (Jacot, Gabriel, and Hongler 2018; Cao and Gu 2019), the estimator $\widetilde{\boldsymbol{\theta}}_{t,a}^{\dagger}$ serves as a good approximation for $\boldsymbol{\theta}_{t,a}^{\dagger}$ despite the non-concave objective function in (12). The detail description is in the regret analysis.
- **NeuralRBMLE via reward-bias-guided parameter correction (NeuralRBMLE-PC):** Note that by (12), finding each $\boldsymbol{\theta}_{t,a}^{\dagger}$ originally involves solving an optimization problem for each arm. To arrive at a more computationally efficient algorithm, we propose a surrogate index for the original index policy in (13) with Gaussian likelihood. We observe that the main difference among the estimators $\boldsymbol{\theta}_{t,a}^{\dagger}$ of different arms lies in the reward-bias term $\alpha(t) f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t,a}^{\dagger})$, as shown in (13). Based on this observation, we propose to first (i) find a *base estimator* $\widehat{\boldsymbol{\theta}}_t$ without any reward bias and then (ii) approximately obtain the arm-specific RBMLE estimators by involving the neural tangent kernel of neural networks. Define

$$\widehat{\boldsymbol{\theta}}_t^{\dagger} := \operatorname*{argmax}_{\boldsymbol{\theta}} \left\{\ell_{\lambda}^{\dagger}(\mathcal{F}_t; \boldsymbol{\theta})\right\}, \tag{14}$$

Notably, $\widehat{\boldsymbol{\theta}}_t^{\dagger}$ can be viewed as the least squares estimate given $\mathcal{F}_t$ for the neural network $f(\cdot; \boldsymbol{\theta})$. We apply J-step gradient ascent with step size η to solve (14), and denote $\widehat{\boldsymbol{\theta}}_t$ as the output of gradient ascent. Let $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) := \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})$ be the gradient of $f(\mathbf{x}; \boldsymbol{\theta})$. Next, we define $\mathbf{Z}_t := \lambda \mathbf{I} + \frac{1}{m} \sum_{\tau=1}^{t} \mathbf{g}(\mathbf{x}_\tau; \widehat{\boldsymbol{\theta}}_\tau)\, \mathbf{g}(\mathbf{x}_\tau; \widehat{\boldsymbol{\theta}}_\tau)^{\top}$ and construct an approximate estimator for $\boldsymbol{\theta}_{t,a}^{\dagger}$ as

$$\bar{\boldsymbol{\theta}}_{t,a} := \widehat{\boldsymbol{\theta}}_t + \frac{\alpha(t)}{m} \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \widehat{\boldsymbol{\theta}}_t), \tag{15}$$

where $\frac{\alpha(t)}{m} \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \widehat{\boldsymbol{\theta}}_t)$ reflects the effect of the reward-bias term on the neural network parameter. Then, we propose a surrogate index $\bar{\mathcal{I}}_{t,a}$ for the index $\mathcal{I}_{t,a}^{\dagger}$ as $\bar{\mathcal{I}}_{t,a} := f(\mathbf{x}_{t,a}; \bar{\boldsymbol{\theta}}_{t,a})$, for all $a \in [K]$ and for all $t \in [T]$. The derivation for the surrogate index $\bar{\mathcal{I}}_{t,a}$ is provided in Appendix E.2. The pseudo code of the NeuralRBMLE-PC of this surrogate index is provided in Algorithm 2. The main advantage of NeuralRBMLE-PC is that at each time step t, the learner only needs to solve one optimization problem for the base estimator $\widehat{\boldsymbol{\theta}}_t$ and then follow the guidance of $\mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \widehat{\boldsymbol{\theta}}_t)$ that are readily available, instead of solving multiple optimization problems.

5 Regret Analysis of NeuralRBMLE

In this section, we present the regret analysis of the NeuralRBMLE algorithms. We denote $\mathbf{H}$ as the NTK matrix and $\widetilde{d}$ as the effective dimension of $\mathbf{H}$. The detailed definitions are in Appendix A.

[5]Compared to the index in (13), one slight modification in Line 8 of Algorithm 1 is the additional factor $\zeta(t)$, which is presumed to be a positive-valued and strictly increasing function (e.g., $\zeta(t)$ can be chosen as $1 + \log t$). This modification was first considered by (Hung et al. 2021). As shown in our regret analysis in Appendix C.1, the technical reason behind $\zeta(t)$ is only to enable the trick of completing the square, and $\zeta(t)$ does not affect the regret bound.

Assumption 1. *$\boldsymbol{H} \succeq \lambda \boldsymbol{I}$, and for all $a \in [K], t \in [T]$, $\|\boldsymbol{x}_{t,a}\|_2 = 1$ and $[\boldsymbol{x}]_j = [\boldsymbol{x}]_{j+d/2}$.*

Based on this assumption, we can ensure that $\mathbf{H}$ is a positive-definite matrix, and this can be satisfied if no two contexts are parallel. The second part ensures that $f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_0) = 0$, which is for the convenience of analysis. Theorem 1 shows the regret bounds of NeuralRBMLE.

Theorem 1. *Under NeuralRBMLE-GA in Algorithm 1, there exist positive constants $\{C_{GA,i}\}_{i=1}^4$ such that for any $\delta \in (0,1)$, if we have $\alpha(t) = \Theta(\sqrt{t})$, $\eta \leq C_{GA,1}(m\lambda + TmL)^{-1}$, $J \geq C_{GA,2}\frac{TL}{\lambda}$, and*

$$m \geq C_{GA,3} \max\Bigg\{ T^{16}\lambda^{-7}L^{24}(\log m)^3, \\ T^6K^6L^6\lambda^{-\frac{1}{2}}\left(\log\left(TKL^2/\delta\right)\right)^{\frac{3}{2}} \Bigg\}, \quad (16)$$

then with probability at least $1-\delta$, the regret satisfies $\mathcal{R}(T) \leq C_{GA,4}\sqrt{T}\cdot\widetilde{d}\log(1+TK/\lambda)$.

The detailed proof of Theorem 1 is provided in Appendix C. We highlight the technical novelty of the analysis for NeuralRBMLE-GA as follows. As described in Section 1, the challenges of establishing the regret bound in Theorem 1 are mainly three-fold: (a) Compatibility dilemma of RBMLE with NTK; (b) Model misspecification; (c) Three-way trade-off due to reward bias. We address the above issues as follows:

- We disentangle the dilemma in issue (a) by presenting a novel induction argument and thereby carefully quantifying the distance between the learned policy parameters and the initial parameter (cf. Lemmas 10-11 along with the supporting Lemmas 7-9).
- We tackle the issue (b) by providing bounds regarding the log-likelihood of the exponential family distributions in analyzing the arm-specific index (cf. Lemmas 12-13).
- Finally, based on the above results, we address the issue (c) by choosing a proper $\alpha(t) = \Theta(\sqrt{t})$ that achieves low regret and enables NTK-based analysis simultaneously.

Theorem 2. *Under NeuralRBMLE-PC in Algorithm 2, there exist positive constant $C_{PC,1}$, $C_{PC,2}$ and $C_{PC,3}$ such that for any $\delta \in (0,1)$, if we have $\alpha(t) = \Theta(\sqrt{t})$, $\eta \leq C_{PC,1}(m\lambda + TmL)^{-1}$, and*

$$m \geq C_{PC,2} \max\Bigg\{ T^{21}\lambda^{-7}L^{24}(\log m)^3, \\ mT^6K^6L^6\lambda^{-\frac{1}{2}}\left(\log\left(TKL^2/\delta\right)\right)^{\frac{3}{2}} \Bigg\}, \quad (17)$$

the with probability at least $1-\delta$, the regret satisfies $\mathcal{R}(T) \leq C_{PC,3}\sqrt{T}\cdot\widetilde{d}\log(1+TK/\lambda)$.

The complete proof is provided in Appendix D. We highlight the challenge and technical novelty of the analysis for NeuralRBMLE-PC as follows: Notably, the main challenge in establishing the regret bound in Theorem 2 lies in that the bias term affects not only the *explore-exploit trade-off* but also the *approximation capability* of the NTK-based analysis. Such a three-way trade-off due to the reward-bias term serves as one salient feature of NeuralRBMLE-PC, compared to RBMLE for linear bandits (Hung et al. 2021) and other existing neural bandit algorithms (Zhou, Li, and Gu 2020; Zhang et al. 2021). Despite the above challenges, we are still able to (i) characterize the interplay between regret and the reward-bias $\alpha(t)$ (cf. (206)-(215) in Appendix D.2), (ii) specify the distance between $\boldsymbol{\theta}_0$ and the learned policy parameter $\bar{\boldsymbol{\theta}}_{t,a}^{\dagger}$ induced by the bias term (cf. Lemma 18), and (iii) carefully handle each regret component that involves $\|\mathbf{g}(\mathbf{x}_t^*; \widehat{\boldsymbol{\theta}}_t)\|_2$ in the regret bound by the technique of completing the square (cf. (217)-(227) in Appendix D.2).

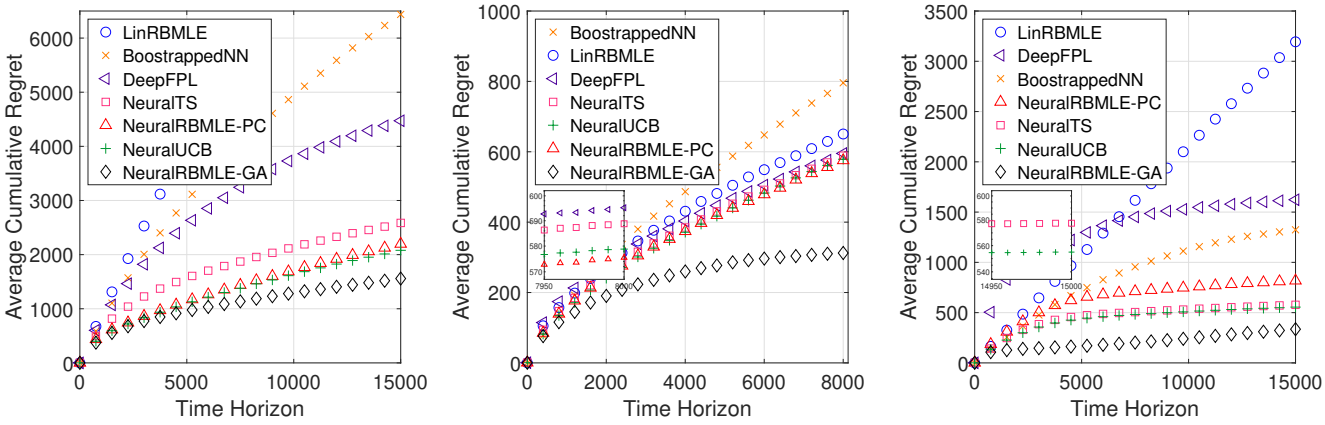
6 Numerical Experiments

We evaluate the performance of NeuralRBMLE against the popular benchmark methods through experiments on various real-world datasets, including Adult, Covertype, Magic Telescope, Mushroom, Shuttle (Asuncion et al. 2003), and MNIST (LeCun, Cortes, and Burges 2010). The detailed configuration of the experiments is provided in Appendix F. Unless stated otherwise, the results reported in this section are the average over 10 random seeds.

Effectiveness of NeuralRBMLE. Figure 1, Table 2, Figure 2, and Figure 5 (cf. Appendix H) show the empirical regret performance of each algorithm. We observe that NeuralRBMLE-GA outperforms most of the other benchmark methods under the six real-world datasets, and NeuralRBMLE-PC achieves empirical regrets competitive to the other state-of-the-art neural bandit algorithms. Table 3 in Appendix H shows the standard deviation of the regrets over random seeds. Notably, we observe that the performance of NeuralRBMLE and LinRBMLE appears more consistent across different random seeds than other benchmarks. This suggests that the reward-biased methods empirically enjoy better robustness across different sample paths.

NeuralRBMLE-GA with different likelihood functions. As described in Sections 4.2 and 5, NeuralRBMLE-GA opens up a whole new family of algorithms with provable regret bounds and can readily support any parametric likelihood. We provide the experiment of NeuralRBMLE-GA with the three different likelihood functions (i) Gaussian: $-\frac{1}{2}(f(\mathbf{x}_s;\boldsymbol{\theta}) - r_s)^2$, (ii) Bernoulli: $r_s f(\mathbf{x}_s;\boldsymbol{\theta}) - \log\left(1 + e^{f(\mathbf{x}_s;\boldsymbol{\theta})}\right)$ and (iii) Categorical distribution with two atoms located at 0 and 1. Notice that for NeuralRBMLE-GA with categorical distribution, we let the output of the neural network $f(\cdot) \in \mathbb{R}^2$ to be the weight of each atom. Figure 6 shows the empirical regrets of NeuralRBMLE-GA under different likelihood functions over 5 random seeds. We can observe that these variants of NeuralRBMLE-GA have empirical regret competitive to other neural bandit algorithms like NeuralUCB, NeuralTS and DeepFPL, and NeuralRBMLE-GA with categorical distribution appears the best in empirical regret. This manifests the connection between NeuralRBMLE and distributional RL as well as corroborates the flexibility of NeuralRBMLE-GA in selecting the surrogate likelihood.

Computational efficiency of NeuralRBMLE-GA.



(a) MNIST (b) Mushroom (c) Shuttle

Figure 1: Cumulative regret averaged over 10 random seeds with $T = 1.5 \times 10^4$.

	NeuralUCB	NeuralTS	GA	DeepFPL
$t_{\mathbf{Z}}$	0.976s	0.975s	N/A	N/A
t_{step}	1.117s	1.116s	**0.15s**	**0.045s**

Table 1: Computation time per step of NeuralRBMLE-GA, NeuralUCB, NeuralTS and DeepFPL. We use t_{step} to denote the total computation time of selecting action and training, and $t_{\mathbf{Z}}$ to denote the computation time of computing the inverse of $\mathbf{Z}$.

NeuralRBMLE-GA and DeepFPL(NPR, (Jia et al. 2022)) achieves the same $\tilde{\mathcal{O}}(\sqrt{T})$ regret as NeuralUCB and NeuralTS, but *without requiring the inverse of* $\mathbf{Z} \in \mathbb{R}^{p\times p}$, where $p = m + md + m^2(L-1)$. The computation of $\mathbf{Z}^{-1}$ can take a substantial amount of time even if the context dimension d is only moderately large. To further verify this, we measure the average per-step computation time and the time spent on computing $\mathbf{Z}^{-1}$ of the three algorithms on the Covertype dataset (with $d = 378$, $m = 100$, and one hidden layer) and show the computational advantage of NeuralRBMLE-GA in Table 1.

Ablation study for m. The large width m of hidden layers is a sufficient condition for regret bounds in NTK regime, and similar conditions also appear in NeuralUCB and NeuralTS. Given that the conditions of m in (16)-(17) are sufficient but not necessary. The ablation study of m is provided in Figure 3. We show in practice NeuralRBMLE can achieve sublinear empirical regret with a relatively small m.

Ablation study for $\alpha(t)$. We provide an ablation study for $\alpha(t)$ by evaluating the empirical regret on MNIST for NeuralRBMLE-GA with $\alpha(t) = t^n$, where $n \in \{0, 0.25, 0.5, 1, 2\}$. The results in Figure 4 corroborate the choice $\alpha(t) = \Theta(\sqrt{t})$ provided by the theoretical result.

The difference in behavior between NeuralRBMLE and other regression-based neural bandit methods. Inspired by (Kassraie and Krause 2022), we leverage the classifier for MNIST dataset provided by Pytorch, and select a collection of particularly ambiguous digits that are classified incorrectly by a well-trained neural network model and can be viewed as hard examples. We compare the estimation error of the models learned by different neural bandit algorithms over these hard examples. The result is provided in Table 4 in Appendix H. We can observe that NeuralRBMLE-GA has a lower MSE than other neural bandit methods.

7 Related Work

To relax the linear realizability assumption, contextual bandits have been studied from the perspectives of using known general kernels (Valko et al. 2013) and the surrogate model provided by Gaussian processes (Chowdhury and Gopalan 2017). Moreover, one recent attempt is to leverage the representation power of deep neural networks to learn the unknown bounded reward functions (Gu et al. 2021; Kveton et al. 2020; Riquelme, Tucker, and Snoek 2018; Zahavy and Mannor 2019; Zhou, Li, and Gu 2020; Zhang et al. 2021; Zhu et al. 2021; Xu et al. 2020; Jia et al. 2022). Among the above, the prior works most related to this paper are NeuralUCB (Zhou, Li, and Gu 2020) and NeuralTS (Zhang et al. 2021), which incorporate the construction of confidence sets of UCB and the posterior sampling technique of TS into the training of neural networks, respectively. Another two recent preprints propose Neural-LinUCB (Xu et al. 2020) and NPR (Jia et al. 2022), which incorporate shallow exploration in the sense of LinUCB and the technique of reward perturbation into neural bandits (similar to DeepFPL), respectively. Different from the above neural bandit algorithms, NeuralRBMLE achieves efficient exploration in a fundamentally different manner by following the guidance of the reward-bias term in the parameter space, instead of using posterior sampling, reward perturbation, or confidence bounds derived from concentration inequalities.

8 Conclusion

This paper presents NeuralRBMLE, the first bandit algorithm that extends the classic RBMLE principle in adaptive control to contextual bandits with general reward functions. Through regret analysis and extensive simulations on real-world datasets, we show that NeuralRBMLE achieves competitive regret performance compared to the state-of-the-art neural bandit algorithms.

Acknowledgements

This material is based upon work partially supported by the National Science and Technology Council (NSTC), Taiwan under Contract No. 110-2628-E-A49-014 and Contract No. 111-2628-E-A49-019, and based upon work partially supported by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24: 2312–2320.

Abeille, M.; Lazaric, A.; et al. 2017. Linear Thompson sampling revisited. *Electronic Journal of Statistics*, 11(2): 5165–5197.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 242–252. PMLR.

Asuncion, A.; Newman, D.; Bache, K.; and Lichman, M. 2003. UCI Machine Learning Repository. *Meta*, 2003.

Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov): 397–422.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 449–458. PMLR.

Borkar, V. 1990. The Kumar-Becker-Lin scheme revisited. *Journal of Optimization Theory and Applications*, 66(2): 289–309.

Campi, M. C.; and Kumar, P. 1998. Adaptive linear quadratic Gaussian control: the cost-biased approach revisited. *SIAM Journal on Control and Optimization*, 36(6): 1890–1907.

Cao, Y.; and Gu, Q. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, 10836–10846.

Chowdhury, S. R.; and Gopalan, A. 2017. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 844–853.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 208–214.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*.

Dumitrascu, B.; Feng, K.; and Engelhardt, B. 2018. PG-TS: Improved Thompson sampling for logistic contextual bandits. In *Advances in neural information processing systems*, 4624–4633.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, 586–594.

Gu, Q.; Karbasi, A.; Khosravi, K.; Mirrokni, V.; and Zhou, D. 2021. Batched Neural Bandits. *arXiv preprint arXiv:2102.13028*.

Hung, Y.-H.; Hsieh, P.-C.; Liu, X.; and Kumar, P. 2021. Reward-Biased Maximum Likelihood Estimation for Linear Stochastic Bandits. In *AAAI Conference on Artificial Intelligence*, volume 35, 7874–7882.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*.

Jia, Y.; Zhang, W.; Zhou, D.; Gu, Q.; and Wang, H. 2022. Learning Neural Contextual Bandits Through Perturbed Rewards. In *International Conference on Learning Representations*.

Jun, K.-S.; Bhargava, A.; Nowak, R.; and Willett, R. 2017. Scalable generalized linear bandits: online computation and hashing. In *Advances in Neural Information Processing Systems*, 98–108.

Kassraie, P.; and Krause, A. 2022. Neural contextual bandits without regret. In *International Conference on Artificial Intelligence and Statistics*, 240–278. PMLR.

Kirschner, J.; and Krause, A. 2018. Information directed sampling and bandits with heteroscedastic noise. In *Conference On Learning Theory*, 358–384.

Kumar, P.; and Becker, A. 1982. A new family of optimal adaptive controllers for Markov chains. *IEEE Transactions on Automatic Control*, 27(1): 137–146.

Kveton, B.; Zaheer, M.; Szepesvari, C.; Li, L.; Ghavamzadeh, M.; and Boutilier, C. 2020. Randomized exploration in generalized linear bandits. In *International Conference on Artificial Intelligence and Statistics*, 2066–2076.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. Mnist handwritten digit database. AT&T Labs.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, 2071–2080.

Liu, X.; Hsieh, P.-C.; Hung, Y. H.; Bhattacharya, A.; and Kumar, P. 2020. Exploration Through Reward Biasing: Reward-Biased Maximum Likelihood Estimation for Stochastic Multi-Armed Bandits. In *International Conference on Machine Learning*, 6248–6258. PMLR.

Mandl, P. 1974. Estimation and control in Markov chains. *Advances in Applied Probability*, 40–60.

Mete, A.; Singh, R.; Liu, X.; and Kumar, P. 2021. Reward biased maximum likelihood estimation for reinforcement learning. In *Learning for Dynamics and Control*, 815–827.

Nesterov, Y. 2013. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1): 125–161.

Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep exploration via bootstrapped DQN. *arXiv preprint arXiv:1602.04621*.

Prandini, M.; and Campi, M. C. 2000. Adaptive LQG Control of Input-Output Systems—A Cost-biased Approach. *SIAM Journal on Control and Optimization*, 39(5): 1499–1519.

Riquelme, C.; Tucker, G.; and Snoek, J. 2018. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *International Conference on Learning Representations*.

Rusmevichientong, P.; and Tsitsiklis, J. N. 2010. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2): 395–411.

Russo, D.; and Van Roy, B. 2016. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(1): 2442–2471.

Russo, D.; and Van Roy, B. 2018. Learning to optimize via information-directed sampling. *Operations Research*, 66(1): 230–252.

Valko, M.; Korda, N.; Munos, R.; Flaounas, I.; and Cristianini, N. 2013. Finite-Time Analysis of Kernelised Contextual Bandits. In *Uncertainty in Artificial Intelligence*.

Vershynin, R. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Xu, P.; Wen, Z.; Zhao, H.; and Gu, Q. 2020. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*.

Zahavy, T.; and Mannor, S. 2019. Deep neural linear bandits: Overcoming catastrophic forgetting through likelihood matching. *arXiv preprint arXiv:1901.08612*.

Zhang, W.; Zhou, D.; Li, L.; and Gu, Q. 2021. Neural Thompson Sampling. In *International Conference on Learning Representations*.

Zhou, D.; Li, L.; and Gu, Q. 2020. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, 11492–11502.

Zhu, Y.; Zhou, D.; Jiang, R.; Gu, Q.; Willett, R.; and Nowak, R. 2021. Pure Exploration in Kernel and Neural Bandits. *Advances in Neural Information Processing Systems*, 34.