ϵ-Neural Thompson Sampling of Deep Brain Stimulation for Parkinson Disease Treatment

Hao-Lun Hsu
Computer Science
Duke University
Durham, NC, USA
hao-lun.hsu@duke.edu

Qitong Gao
Electrical and Computer Engineering
Duke University
Durham, NC, USA
qitong.gao@duke.edu

Miroslav Pajic
Electrical and Computer Engineering
Duke University
Durham, NC, USA
miroslav.pajic@duke.edu

***Abstract*—Deep Brain Stimulation (DBS) stands as an effective intervention for alleviating the motor symptoms of Parkinson's disease (PD). Traditional commercial DBS devices are only able to deliver fixed-frequency periodic pulses to the basal ganglia (BG) regions of the brain, *i.e.*, continuous DBS (cDBS). However, they in general suffer from energy inefficiency and side effects, such as speech impairment. Recent research has focused on adaptive DBS (aDBS) to resolve the limitations of cDBS. Specifically, reinforcement learning (RL) based approaches have been developed to adapt the frequencies of the stimuli in order to achieve both energy efficiency and treatment efficacy. However, RL approaches in general require significant amount of training data and computational resources, making it intractable to integrate RL policies into real-time embedded systems as needed in aDBS. In contrast, contextual multi-armed bandits (CMAB) in general lead to better sample efficiency compared to RL. In this study, we propose a CMAB solution for aDBS. Specifically, we define the context as the signals capturing irregular neuronal firing activities in the BG regions (*i.e.*, beta-band power spectral density), while each 'arm' signifies the (discretized) pulse frequency of the stimulation. Moreover, an ϵ-exploring strategy is introduced on top of the classic Thompson sampling method, leading to an algorithm called ϵ-Neural Thompson sampling (ϵ-NeuralTS), such that the learned CMAB policy can better balance exploration and exploitation of the BG environment. The ϵ-NeuralTS algorithm is evaluated using a computation BG model that captures the neuronal activities in PD patients' brains. The results show that our method outperforms both existing cDBS methods, as well as the baselines that do not use the ϵ-exploring as introduced by our method (*i.e.,* the vanilla Thompson sampling method).**

***Index Terms*—Deep Brain Stimulation, Contextual Multi-armed Bandit, Thompson Sampling**

I. Introduction

Millions of individuals in the U.S. suffer from Parkinson's disease (PD), a neurodegenerative disorder causing motor symptoms such as tremors, muscle stiffness, and bradykinesia [1]. Deep brain stimulation (DBS) has become widely used to treat motor symptoms by delivering electric pulses to the basal ganglia (BG) region of the brain through implantable devices [2]–[5] illustrated in Figure 1. DBS system consists of two main components: the electrode and the pulse generator. The electrode is a thin and insulated wire implanted in the brain with its tip positioned within the BG region. The pulse generator is usually placed under the skin near the collarbone or implanted closer to chest or abdomen. Two components are connected with an insulated wire passing under the skin of the head, neck, and shoulder so that the electrical impulses can be sent from the pulse generator, up along the extension wire and the electrode, and into the brain for treatment.

DBS can significantly improve patients' daily life by alleviating PD symptoms; however, existing commercial DBS devices can only provide stimuli with pre-determined and fixed parameters (*e.g.,* pulse frequency and amplitude) – *i.e.*, continuous DBS (cDBS). To facilitate desirable therapeutic outcomes, the process of determining the parameters is often time-consuming because the parameters are usually determined by trial-and-error over multiple clinical visits [6]. In addition, stimulation with constant high frequency and amplitude significantly shortens the battery life of the implantable device and can result in serious side effects [7]. Therefore, there has been a notable surge in research focused on advancing the automation of parameter selection for DBS, especially feedback-based stimulation controllers.

Existing research has primarily focused on developing adaptive DBS (aDBS) techniques, which automatically adjust stimulation parameters using various electrophysiological biomarkers as feedback signals [7]–[14]; specifically, local field potentials (LFPs) from the BG, internal electroencephalography (iEEG), and data from wearable devices such as electromyography and accelerometers with predefined thresholds established by physicians based on trial data. While the aforementioned aDBS approaches show promise in reducing energy consumption and mitigating stimulation-related side effects [11]–[15], the configuration of aDBS devices to optimize the balance between stimulation efficacy and battery efficiency remains a labor-intensive task. To reduce these substantial efforts, a distributed closed-loop neuromodulation architecture designed for the automated tuning of Proportional Integral (PI) controllers in DBS, leveraging Bayesian optimization is further introduced in [16].

Recent research has explored the application of reinforcement learning (RL) to devise closed-loop controllers for aDBS in the context of PD. In particular, the studies conducted by [17] introduce an approach where EEG and LFP signals are employed to define the state space within the RL

This work is sponsored in part by the NSF CNS-1837499 award and the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562, as well as by NIH UH3 NS103468.

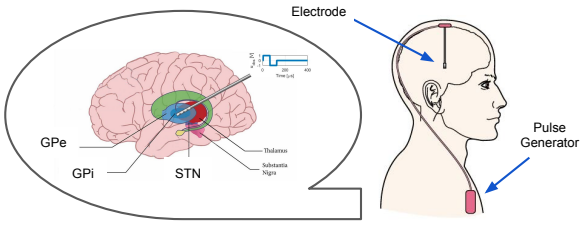


Fig. 1. Deep brain stimulation: the implantable pulse generator is placed in the patient's chest; electrodes that can record local field potentials (LFPs) and deliver stimulation are positioned in the basal ganglia (BG) to stimulate the subthalamic nucleus or the internal segment of the globus pallidus (GPi).

framework. They employ fitted Q-iteration to synthesize RL control policies. These policies are geared toward selecting stimulation frequencies that aim to enhance energy efficiency. The work presented in [18], and extended in [12], [13] utilizes deep actor-critic RL to craft personalized stimulation patterns tailored to each patient, which improves both energy efficiency and stimulation efficacy.

Compared with RL methods, contextual multi-armed bandit (CMAB) algorithms are more sample-efficient, which can better facilitate real-world DBS applications as data collection with human participants can be costly [19], [20]. Moreover, lower computational resources are required for training and evaluating CMAB policies in general, facilitating better compatibility with the latest generation of embedded DBS systems, which do not provide the functionalities and bandwidth needed by executing full RL policies in real-time [21].

In this work, we introduce a CMAB approach to adapt the stimulation frequency of DBS, in response to the contexts defined as the beta-band (13-35 HZ) power spectral density (P_β) [22] of the LFP signals collected from the BG [23], [24]. Moreover, the bandit arms represent the (discretized) stimulation frequency from the range of 0 Hz (*i.e., turn off DBS*) to 180 Hz (*i.e., cDBS*). Specifically, we propose an algorithm called ϵ-Neural Thompson Sampling (ϵ-NeuralTS), which blends deep neural networks with Thompson Sampling (TS) [25]. It optimizes neural network approximators over a Bayesian objective, to estimate the posterior return distribution, normally parametrized as Gaussian, capturing the expected return of each arm with confidence, *e.g.*, conditional (co-)variances to quantify the level of uncertainty. In addition, the arms are selected greedily by directly maximizing the expected return (*i.e.*, the expectation of the posterior distribution) with probability $1-\epsilon$, or to sample from the posterior with probability ϵ. Here, ϵ is hyper-parameter that helps balance exploration and exploitation.

In what follows, a computational Basal Ganglia Model (BGM) [18] is used as the testbed for training and evaluation of the CMAB policies, where P_β and Error Index (EI) [26] are considered as the Quality-of-Control (QoC) metrics. We conduct comprehensive hyper-parameter tuning over the ϵ value and the reward function of CMAB method followed by the comparison with existing approaches. The results show that our method outperforms both existing cDBS methods and

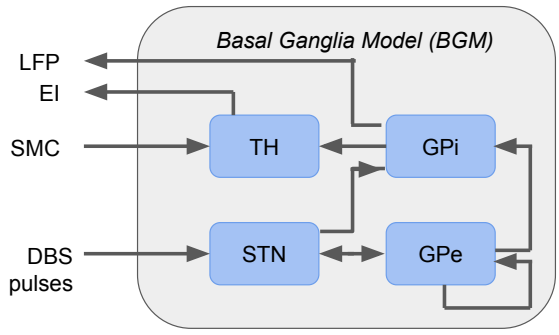


Fig. 2. An illustration of the computational brain model. The DBS stimulation is deployed to the subthalamic nucleus (STN), propagating to the other sub-regions. Error index (EI) is computed with the activations passing from sensorimotor cortex (SMC) to thalamus (TH).

vanilla TS methods, in terms of the two metrics above as well as energy/time efficiency and robustness.

The main contributions of this work are:

1) We re-formulate the aDBS problem into CMAB, where the interactions between the BG and the CMAB policy pertain to an environment with pre-defined feature contexts, action space, and reward functions.
2) We propose a novel ϵ-NeuralTS algorithm that is suitable for deployment over the latest generation of embedded DBS systems [21]. Moreover, it can trade off exploration and exploitation during training, leading to improved sample efficiency. As a result, it lays out the foundation of next-generation aDBS frameworks.
3) We successfully demonstrate that our method outperforms several baselines, including both existing CMAB baselines and cDBS from the perspective of task performance and real-world scenarios.

The paper is organized as follows. Section II introduces a Basal Ganglia Model (BGM) of the brain, the QoC metrics used to evaluate DBS control performance and the background of existing algorithms from MAB to CMAB. Our problem statement is formulated with the adaption of CMAB to DBS in Section III. We introduce our proposed ϵ-NeuralTS in Section IV. The experiments and analyses of the results are elaborated in Section V. This work is concluded with possible future extensions in Section VI.

II. Preliminaries

In this section, we start by describing the BGM with a formal definition of the QoC metrics the will be utilized for evaluating DBS control performance. We refer readers to [23], [24] for an in-depth review of the model. We then introduce the background of Multi-armed Bandit (MAB) and extend it to contextual bandit settings.

A. Computational BGM

The Basal Ganglia (BG) is a prominent cerebral region composed of three principal sub-regions, namely, the subthalamic nucleus (STN), globus pallidus pars externa (GPe), and globus pallidus pars interna (GPi). To comprehensively

capture and quantify the manifestations of Parkinson's disease (PD), it is imperative to include not only these sub-regions but also the thalamic region (TH) and the sensory-motor cortex (SMC) inputs within the PD-specific brain model, as illustrated in Figure 2.

Supposing that there exists n neurons in each sub-region, the state from the computational BGM at each time t can be succinctly represented as a vector denoting electrical potential, as delineated below:

$$\boldsymbol{v}^q(t) = [\nu_1^q, ..., \nu_n^q]; \tag{1}$$

here, ν_j^q denotes the value of the j^{th} neuron with the corresponding sub-region $q \in \{STN, GPe, GPi, TH\}$. The initial states of these neurons are considered model parameters that are stochastically determined in our experimental setup. Neurons are interconnected through chemical synapses, forming BGM structure illustrated in Figure 2. The neural activation of each neuron at time t is captured by binary events $a_j^q \in \{0, 1\}$, which occur when the neuron's electrical potential ν_j^q exceeds a predefined threshold h_j^q, defined as

$$a_j^q(t) = \mathbb{I}\Big([v_j^q(t) > h_j^q] \wedge [\exists \delta, \forall \epsilon \in (0, \delta), v_j^q(t - \epsilon) < h_j^q] \Big). \tag{2}$$

We now formally define the two QoC metrics (*i.e.,* P_β and EI) in order to evaluate the efficacy of DBS.

1) Error Index (EI): EI is defined as the portion of erroneous TH neuron activations in response to SMC inputs[1] SMC_τ at $t = \tau$. Specifically, SMC_τ can modulate TH neuron potentials and is expected to activate all TH neurons exactly once within a time window of $25ms$ in healthy brains. In contrast, in the context of Parkinson's disease (PD), no such response or activation should be present within the $25ms$ window immediately following the reception of an SMC input. Formally, EI is defined as

$$EI(t) = \frac{\sum_{i=1}^{n} \sum_{t=t_0}^{t} a_i^{TH,err}(t)}{n \Big| SMC_\tau |_{t_0}^{t} \Big|}, \tag{3}$$

where $a_i^{TH,err}(t) = 1(\text{or } 0)$ indicates an erroneous (or correct) TH neuron activation at time t. Intuitively, every neuron in TH, and $\Big| SMC_\tau |_{t_0}^{t} \Big|$ is the cumulative number of SMC inputs received between the initial time t_0 and the current step t. Note that EI is bounded to the range $[0, 1]$ because EI is defined as a ratio. The goal for the DBS controller is to maintain EI as low as possible.

2) Beta-band Power Spectral Density (P_β): In a PD brain, the GPi region exhibits pathological oscillations of neurons at frequencies within the $13Hz - 35Hz$ band (*i.e.,* beta band), which do not exist in a healthy brain. P_β is defined as

$$P_{\beta j}^{GPi} = \int_{\omega = 2\pi \cdot 13Hz}^{2\pi \cdot 35Hz} P_j^{GPi}(\omega) d\omega, \tag{4}$$

where $P_j^{GPi}(\omega)$ is the single-sided power spectral density of the j^{th} neuron's potential in the GPi region. Therefore, the beta band power for the entire region with n neurons can be computed as

$$P_\beta = \frac{1}{n} \sum_{j=1}^{n} P_{\beta j}^{GPi}. \tag{5}$$

Note that EI can directly distinguish healthy brains from PD ones, but is intractable to be obtained in the real world [18]. On the other hand, P_β can be noisy and sometimes may not distinguish healthy and PD brains at the same level of EI [18], P_β can be easily obtained by typical DBS systems in clinical practice [21]; thus, it also serves as an imperative biomarker to quantify the severity of PD. The details of how we evaluate the feasibility of P_β will be described in Section V-A.

B. Multi-armed Bandit (MAB)

We first introduce the basics of MAB, which is necessary for reviewing the preliminaries of CMAB. The MAB problem is a sequential game between a bandit learner and the environment. The game is played over T rounds[2], where T is a positive integer called the horizon. At each time $t \in [0, T]$, the bandit learner first chooses an action a_t from a given set $\mathcal{A}$, and then the environment receives the corresponding reward $R_t \in \mathbb{R}$. Actions are often called *arms*, so K-armed bandits indicate that the cardinality of $\mathcal{A}$ is K. The bandit learner should choose a_t depending on the history $D = (a_1, R_1, ..., a_{t-1}, R_{t-1})$. The common objective of the bandit learner is to learn a policy, which is a mapping from history to the next action, to maximize the cumulative reward over all T rounds. The final performance is evaluated by *regret*, which is defined as follows.

Definition 1 (Regret): [27] The regret of the bandit learner with respect to a policy π is the difference between the total expected reward obtained by using policy π for T rounds and the total expected reward collected by the bandit learner over T rounds. The regret relative to a set of policies Π is the maximum regret relative to any policy $\pi \in \Pi$ in the set.

The main challenge in the bandit problem is addressing the exploitation-exploration trade-off, which targets reaching a subtle balance between following the myopically better arm and choosing an under-sampled worse arm. Existing algorithms for maximizing the cumulative reward in bandits problems mainly follow either one of the following two algorithmic frameworks – upper confidence bound (UCB) and Thompson sampling (TS), as introduced in Section II-B1 and Section II-B2, respectively.

1) Upper Confidence Bound: The UCB algorithm leverages the principle of optimism in the face of uncertainty. The optimal principle means using the data observed so far to assign to each arm a value (*i.e.,* UCB which with high probability is an overestimate of the unknown mean). Assuming the upper confidence bound assigned to the optimal arm is indeed an

[1]Healthy brains could also respond to SMC_τ erroneously with a low probability ($< 0.1\%$).

[2]In this work, we alternatively use **round**, **time**, and **time step** according to the corresponding context but with the same meaning.

overestimate, then another arm can only be played if its UCB is larger than that of the optimal arm, which in turn is larger than the mean of the optimal arm. Then the additional data provided by playing a suboptimal arm means that the UCB for this arm will eventually fall below that of the optimal arm.

UCB is defined formally as follows. Let $(R_t)_{t=1}^T$ be a sequence of independent 1-subgaussian random variable with mean μ and $\hat{\mu} = \frac{1}{n}\sum_{t=1}^T R_t$. Then

$$\mathbb{P}\left(\mu \geq \hat{\mu} + \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leq \delta, \forall \delta \in (0,1). \tag{6}$$

When considering its options in time t, the bandit learner has observed $T_k(t-1)$ samples from arm k and received rewards from that arm with an empirical mean of $\hat{\mu}_k(t-1)$. Then a reasonable candidate for the unknown mean of the k^{th} arm as large as plausibly possible is

$$UCB_k(t-1,\delta) = \begin{cases} \infty & \text{if } T_k(t-1) = 0 \\ \hat{\mu}_k(t-1) + G & \text{otherwise,} \end{cases} \tag{7}$$

where $G = \sqrt{\frac{2\log(1/\delta)}{(T_k(t-1))}}$. The index of UCB is the sum of the empirical mean of rewards experienced so far and the exploration bonus (*i.e.,* confidence width). Note that the exploration bonus has different versions according to different types of UCB algorithms, such as asymptotic optimality and minimax optimality. The high-level structure of the UCB-based algorithm is to start with the inputs of the number of arms K and the error probability δ. For each time $t \in [T]$, the bandits leaner choose arm $a_t = \text{argmax}_i\, UCB_k(t-1,\delta)$ right before observing reward R_t and updating UCB. Following this algorithm, the bandit learner explores arms more often if they are (a) promising because $\hat{\mu}_k(t-1)$ is large or (b) not well explored because $T_k(t-1)$ is small.

2) Thompson sampling: TS, also called, posterior sampling, tackles MAB problems using a Bayesian approach. TS maintains a probability distribution for each arm's expected reward, representing their uncertainty about the true reward distribution. Given the set of history D, TS approach aims to learn the parameter $\boldsymbol{\theta}$ of the true reward distribution. TS starts with a prior distribution and a posterior distribution of θ_k for each arm $k \in [0, K-1]$, if we view $\boldsymbol{\theta}$ as the concatenation of θ_k. This posterior distribution can be acquired by the Bayes rule, $P(\boldsymbol{\theta}|D) \propto P(R_t|a_t,\boldsymbol{\theta})P(\boldsymbol{\theta})$, where $P(R_t|a_t,\boldsymbol{\theta})$ is a parametric likelihood function.

In the vanilla TS, the algorithm samples from the corresponding posterior distributions $\theta_k(t)$ for all $k \in [0, K-1]$, and selects the best arm $a_t = \text{argmax}_{k\in[0,K-1]}\, \theta_k(t)$ right before observing reward R_t and updating the corresponding posterior distribution. We refer to the details of TS in [27].

C. Contextual Bandits

Contextual bandits are a wide class of sequential decision problems, where the bandit learner makes the decision based on an observation of an action set consisting of feature vectors as contexts for different actions. In particular, at time $t \in [0, T]$, the bandit learner observes the context $\boldsymbol{x}$ consisting of K context vectors $\{\boldsymbol{x}_{t,k} \in \mathbb{R}^d | k \in [0, K-1]\}$. The bandit learner then selects an action $a_t \in \mathcal{A}$ and receives the corresponding reward $R_{t,a_t} = h(\boldsymbol{x}_{t,a_t}, \boldsymbol{\theta}) + \xi_t$, where $h : \mathbb{R}^d \to \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}^d$ is an unknown weight parameter for bandit learner; $\xi_t \in \mathbb{R}$ is a random noise incurred in the observation, which is standard in the stochastic bandit literature [28], [29]. To simplify our notation, we can assume that the reward is independent of the time t and the noise ξ_t can be ignored. We can further formulate our reward function with the whole context $\boldsymbol{x}$ expressed as $R = h(\boldsymbol{x}, \boldsymbol{\theta})$ by dropping the subscripts. For instance, we have $h(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{x}^\top \boldsymbol{\theta}$ in linear contextual bandits [28], [30], [31], and $h(\boldsymbol{x}, \boldsymbol{\theta}) = \mu(\boldsymbol{x}^\top \boldsymbol{\theta})$ for generalized linear bandits [32]–[35], where $\mu(\cdot)$ is a link function. Our work aligns with the neural contextual bandits [36]–[39] so that $h(\boldsymbol{x}, \boldsymbol{\theta})$ is a neural network, where $\boldsymbol{\theta}$ is the concatenation of all weight parameters and $\boldsymbol{x}$ is the input.

Within the realm of contextual bandit problems, algorithms grounded in Optimism in the Face of Uncertainty (OFU) are often required to solve a bi-linear optimization problem, which makes them computationally expensive to implement outside of simple problems despite their stronger theoretical guarantees. In contrast, Thompson Sampling (TS) algorithms offer a more computationally efficient alternative. These methods only require solving a linear optimization problem on the set of available arms. This efficiency stems from the fact that the inherent uncertainty encapsulated within the posterior distribution naturally accommodates exploration in the parameter space. Moreover, it is noteworthy that TS has been observed to be empirically competitive with or even superior to OFU-based algorithms in practical scenarios [40].

III. Problem Formulation

We formulate a K-armed CMAB problem for selecting the frequency of the stimulus for PD in DBS with P_β as context inputs. Specifically, the context features s_t at a discrete round, which can be defined as a sequence of P_β at a fixed rate, $m \in \mathbb{Z}^+$, over a window of size T_w. *i.e.,*

$$s_t = [\beta_{(t)}, \beta_{(t+m)}, \beta_{(t+2m)}, ..., \beta_{(t+T_w-m)}], \tag{8}$$

where $\beta_{(\cdot)}$ represents P_β evaluated at $l = T_w/m$ number of equally-spaced intervals within the window. The bandit learner can select its action in time t as a_t from K arms, where $K = 13$ in our problem setting. We limit the maximum stimulus frequency to $180Hz$ in the computational BGM.

To have a better action mapping strategy, according to each arm $k \in [0, K-1]$, we can have $F = 15k$ (*e.g.,* when $k = 12$, the stimulus frequency achieves $180Hz$). Then the selected arm a_t can be mapped back to the action space for the BGM. We change the stimulation frequency every T_w steps, so the mapped action u_t that the bandit learner can take at time t is

$$u_t = [u_{(t)}, u_{(t+m)}, u_{(t+2m)}, ..., u_{(t+T_w-m)}], \tag{9}$$

$$u_{(t+j)} = \begin{cases} 1 & \text{if a pulse is triggered at time } t+j \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

Algorithm 1 ϵ-Neural Thompson Sampling (ϵ-NeuralTS))

1: **Input**: number of rounds T, exploration variance ν, initialized weight of neural network $\boldsymbol{\theta}_0$ with network width m, regularization parameter λ, exploration probability ϵ
2: $\boldsymbol{U}_0 = \lambda \boldsymbol{I}$
3: **for** $t = 1, ..., T$ **do**
4: **for** $k = 1, ..., K$ **do**
5:

$$R_{t,k} \begin{cases} \sim \mathcal{N}(f(\boldsymbol{x}_{t,k}), \nu^2 \sigma_{t,k}^2) & \text{w.p. } \epsilon \\ = f(\boldsymbol{x}_{t,k}) & \text{w.p. } 1-\epsilon \end{cases}$$

6: **end for**
7: Pull arm a_t and receive reward R_{t,a_t}, where $a_t = \text{argmax}_{k \in [0,K-1]} R_{t,k}$
8: Set $\boldsymbol{\theta}_t$ as the output of gradient descent for solving (14)
9: $\boldsymbol{U}_t = \boldsymbol{U}_{t-1} + g(\boldsymbol{x}_{t,a_t}; \boldsymbol{\theta}_t) g(\boldsymbol{x}_{t,a_t}; \boldsymbol{\theta}_t)^\top / m$
10: **end for**

where $j \in [0, T_w - m)$. Finally, we define our reward function as $R_{t,k} = -\bar{s}_{t+1} - C \cdot a_t$, where $\bar{s}_{t+1}$ is the mean of the whole vector of s_{t+1} in (8) and $C \cdot a_t$ is the selected action from the bandit learner multiplied by a constant coefficient $C \in \mathbb{R}$. Again, here the value of a_t is $k \in [0, K-1]$.

Therefore, $C \cdot a_t$ can be viewed as a penalty on the frequency value, which can encourage the bandit learner to reduce the P_β defined in s_{t+1} as well as to consume less energy, resulting in energy efficiency and relatively mild side effects, which is highly relevant to safety issues and therapy effectiveness [41], [42]. Typically the goal in CMAB is to choose actions that maximize the cumulative reward over T steps, which is equivalent to minimizing the cumulative regret $r(T)$, defined as the difference between the maximum possible context-dependent reward and the actually received reward

$$r(T) = \mathbb{E}\left[\sum_{t=1}^{T} (R_{t,k^*}^* - R_{t,k})\right], \tag{11}$$

where R_{t,k^*}^* is the reward with optimal action $a_t = k^*$ and $k^* \in \text{argmax}_k \mathbb{E}[R_{t,k}]$. However, note that EI is the oracle (ground truth) to evaluate the severity of PD symptoms and the optimal value of EI can be minimized to be closer to 0. With the property of EI, we can introduce EI as our regret for the final evaluation[3] and we quantify the task performance of different algorithms by comparing each cumulative regret.

Finally, besides evaluating the task performance, our goal is to also extract the energy consumption component from the reward function as the evaluation of energy efficiency.

IV. ϵ-NeuralTS

NeuralTS is a CMAB method designed to harness the potential of deep neural networks for both exploration and exploitation [37]. Central to this algorithm is an innovative approach to modeling the posterior distribution of rewards. Specifically, compared to the typical ways of implementing TS with neural network sampling the weight parameters, NeuralTS samples from the posterior distribution of the scalar reward with the mean determined by the neural network approximator and the variance constructed based on the neural tangent features associated with the corresponding neural network. Therefore, NeuralTS is simpler and more efficient because the number of parameters can be large in practice.

During learning, the reward function is unknown to the bandit learner. To estimate the unknown reward given a contextual vector $\boldsymbol{x}$, we build a fully connected neural network $f(\boldsymbol{x}, \boldsymbol{\theta})$ for approximation [37], defined recursively by

$$f_1 = \boldsymbol{W}_1 \boldsymbol{x},$$

$$f_l = \boldsymbol{W}_l \text{ReLU}(f_{l-1}), 2 \leq l \leq L,$$

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sqrt{m} f_L, \tag{12}$$

where $\text{ReLU}(x) := \max\{x, 0\}$, m is the width of the neural network, and $\boldsymbol{W}_i$ denotes as the weight parameters of i^{th} layer in the full neural network. Therefore, $\boldsymbol{\theta} = \big(\text{vec}(\boldsymbol{W}_1); ...; \text{vec}(\boldsymbol{W}_L)\big)$ is the collection of parameters of the whole neural network. Finally, $g(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta})$ is the gradient of $f(\boldsymbol{x}, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$.

We summarize our ϵ-NeuralTS in Algorithm 1. We firstly input the number of rounds T, exploration variance $\nu > 0$, initialized neural network, regularization parameter λ, and exploration probability ϵ. Then we initialize a covariance matrix $\boldsymbol{U}_0 = \lambda \boldsymbol{I}$, where $\boldsymbol{I}$ is an identity matrix.

Inspired by the recent work ϵ-TS [43], [44] for non-contextual and weight parameter sampling, our novel ϵ-NeuralTS builds upon NeuralTS with ϵ exploring. Specifically, for each time $t \in [0, T]$, we estimate the reward for each arm $k \in [0, K-1]$. When selecting an arm, it only explores with sampling the reward from its posterior distribution with probability ϵ while the arm is played based on empirical mean rewards with probability $1 - \epsilon$, where $\epsilon \in (0, 1)$ is a user-defined parameter in Line 5 in Algorithm 1. Note that the $\sigma_{t,k}$ in $\mathcal{N}(f(\boldsymbol{x}_{t,k}), \nu^2 \sigma_{t,k}^2)$ is calculated by

$$\sigma_{t,k}^2 = \lambda g^\top(\boldsymbol{x}_{t,k}; \boldsymbol{\theta}_{t-1}) \boldsymbol{U}_{t-1}^{-1} g(\boldsymbol{x}_{t,k}; \boldsymbol{\theta}_{t-1}) / m. \tag{13}$$

Therefore, ϵ-NeuralTS can improve both sample and computational efficiency by reducing the number of calculations. Then the bandit learner pulls the arm with the maximum estimated reward in Line 7. Once the reward is observed, it updates the posterior (Lines 8 & 9). The mean of the posterior distribution is set to the output of the neural network, whose parameter is the solution to the following l_2-regularized square loss minimization problem:

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \sum_{i=1}^{t} [f(\boldsymbol{x}_{i,a_i}, \boldsymbol{\theta}) - R_{i,a_i}]^2 / 2 + m\lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 / 2, \tag{14}$$

where the regularization term centers at the randomly initialized network parameter $\boldsymbol{\theta}_0$.

[3]Note that the EI is not involved in the reward function and the context feature during learning.

V. Experiments

In this section, we evaluate our proposed ϵ-NeuralTS against other contextual bandits algorithms and the controller with periodic stimulation patterns that are employed in [6], [45], [17] over computational BGM. For a fair comparison, we set up the sampling duration $l = T_w = 2$ seconds for all contextual bandits algorithms, indicating that the effect of every arm will last within this duration.

Recall that our context feature s_t is defined as a sequence of P_β sampled at a fixed rate m over a window size T_w. Since the context feature is shared among all arms, we follow [33], [36] to construct context vectors $\boldsymbol{x}$ for different arms in the following way: given a context feature $s \in \mathbb{R}^d$ (*i.e.,* $d = T_w$), we transform it into K contextual vectors $\boldsymbol{x} = [\boldsymbol{x}^{(1)}; ...; \boldsymbol{x}^{(K)}] \in \mathbb{R}^{Kd}$ (*e.g.,* $\boldsymbol{x}^{(1)} = (s, 0, ..., 0)$ and $\boldsymbol{x}^{(K)} = (0, ..., 0, s)$). We learn the parameters $\boldsymbol{\theta}$ of the neural network, discussed in Section IV, with $\mathbf{x}$ as inputs for our ϵ-NeuralTS. Specifically, we build a fully connected neural network with a sequence of 3 layers (32 neurons per layer) followed by the ReLU activation function.

All our experiments are run on Nvidia RTX A5000 with 24GB RAM. In particular, our experiments focused on the following tasks:

1) To evaluate the feasibility of using P_β as a PD biomarker during learning for context feature and reward function, we conduct an experiment to find the correlation between P_β and EI.
2) We tune the coefficient of the penalty term in the reward function and the ϵ value of ϵ-NeuralTS.
3) We compare our proposed ϵ-NeuralTS against existing CMAB methods and classical periodic controllers. Note that we do not explicitly compare our methods with other aDBS approaches because our environment setups are mostly not the same. In addition, although our experiment shares similar computational BGM with [18], we only consider P_β as the input state, which contains less information but be more realistic.
4) We evaluate the impact of different ϵ on the running time ϵ-NeuralTS algorithm.
5) Finally, we evaluate ϵ-NeuralTS on the robustness to delayed rewards.

A. Feasibility of P_β as a PD Biomarker

To evaluate the feasibility of using P_β as the PD biomarker during learning for context feature and reward function, we firstly experiment to find the correlation between P_β and EI. In particular, we randomly deploy 9 pulses within $200ms$ on the computational BGM and collect the corresponding P_β and EI. The distribution is shown in Figure 3 with Pearson's Correlation Coefficient = 0.866. With the high correlation between P_β and EI (closer to 1.0), P_β can be seen as an indicator of PD symptoms with noises. Therefore, we adopt P_β for the feature context and reward function as described in Section III.

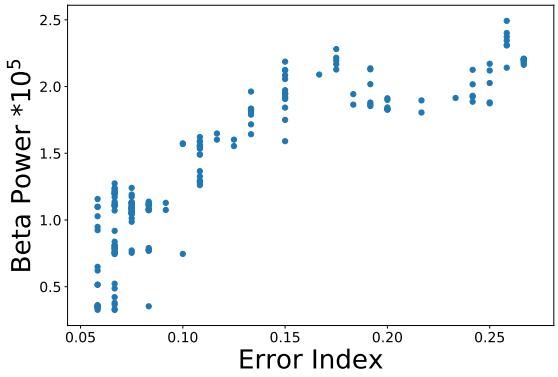


Fig. 3. Correlation between two QoC (*i.e., P_β and EI*) with Pearson's Correlation Coefficient: 0.866.

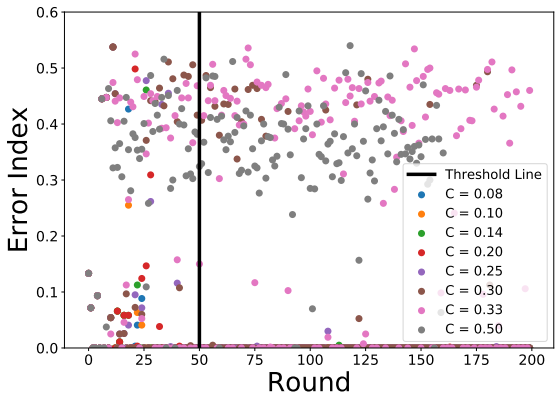


Fig. 4. Learning curve with different penalty coefficients using NeuralTS (lower EI is better).

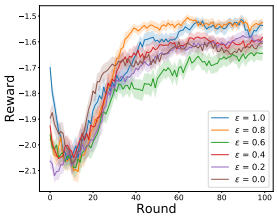
TABLE I
Average Frequency with different penalty coefficient after Threshold Line for NeuralTS.

Penalty Coefficient C	**0.08**	**0.10**	**0.14**	**0.20**	**0.25**	**0.30**	**0.33**	**0.50**
Average Arm k	8.6	8.1	8.0	9.0	8.9	7.9	4.1	0.6

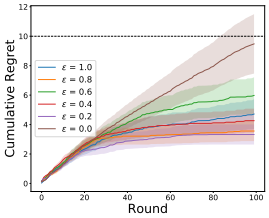
B. Hyper-parameter Tuning of Penalty Coefficient

Recall that our reward function is defined as $R_{t,k} = -\bar{s}_{t+1} - C \cdot a_t$, where $\bar{s}_{t+1}$ is the mean of the whole vector of s_{t+1} in (8) and $C \cdot a_t$ is the penalty of the stimulus with higher frequency accompanied by a constant coefficient $C \in \mathbb{R}$. Note that in [18], the reward function is designed with 4 discrete categories according to the values of P_β and EI, which requires access to EI and more engineering work on deciding the reward values for 4 different categories. Our penalty coefficient C can be tuned within a smaller search space. Since the value of penalty coefficient C will influence the reward function and EI can serve as the final evaluation, we aim to find a suitable C so that the learned policy can maintain a low EI (< 0.1) and lower stimulation frequency with less energy consumption and side effects.

Intuitively, higher stimulation frequency can be more effective in suppressing PD symptoms and larger C will discourage

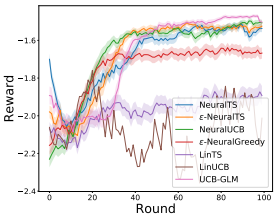


(a) Reward

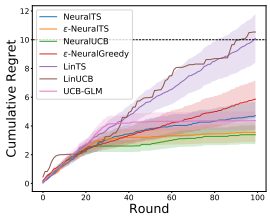


(b) Cumulative Regret

Fig. 5. Task Performance for ϵ-NeuralTS with different ϵ averaged over 10 seeds. Shaded areas denote the standard error: (a) task reward (higher is better), (b) cumulative regret (lower is better).



(a) Reward



(b) Cumulative Regret

Fig. 6. Task Performance for ϵ-NeuralTS against other methods averaged over 10 seeds. Shaded areas denote the standard error: (a) task reward (higher is better), (b) cumulative regret (lower is better).

the policy from selecting a higher simulation frequency. Thus, a trade-off exists between the task and safety (*e.g.,* side effects) performance. Task performance is our priority condition before we select the lowest average stimulation frequency.

To have a fair comparison, we consider our strong baseline NeuralTS [37] for hyper-parameter tuning. Figure 4 shows the EI values with different coefficients $C = [0.08, 0.10, 0.14, 0.20, 0.25, 0.30, 0.33, 0.50]$. We observe that most of the settings have EI values smaller than 0.1 after $t = 50$ rounds (*i.e.,* threshold line) while $C = [0.30, 0.33, 0.50]$ cannot converge even with longer rounds. We average the frequency after the threshold line for all the settings, resulting in $C = 0.14$ the lowest average frequency (*i.e.,* stimulation frequency $F = 15i$, where $k \in [0, K-1]$) with low EI (< 0.1) in Table I; here, $K = 13$. Hence, we adopt $C = 0.14$ in our reward function for the remaining experiments.

The results for the average frequency are reported in Table I.

C. Hyper-parameter Tuning of ϵ for ϵ-NeuralTS

Before comparing with other CMAB methods, we investigate the optimal ϵ via our reward function, which has already been decided as a better trade-off between task performance (*i.e.,* P_β) and penalty on frequency value with coefficient. Note that NeuralTS can be viewed as the special case of ϵ-NeuralTS with $\epsilon = 1.0$, so we also include it in comparison.

We conduct all the setups with different ϵ for 10 random trials to represent 10 different patients. We record the task reward and cumulative regret (*i.e.,* cumulative EI) with different ϵ in Figure 5. We observe that worse performance does happen with insufficient exploration when $\epsilon < 0.8$ in our task. However, when $\epsilon = 0.8$, we reduce by 20% the number of sampling and calculations with gradient descent as well as speed up the convergence with competitive performance compared with vanilla NeuralTS (*i.e.,* $\epsilon = 1.0$). On the other hand, we realize that the cumulative regret diverges when $\epsilon = 0.0$, which is reasonable due to the extreme imbalance between exploration and exploitation. However, $\epsilon = 0.2$ results in the minimum cumulative regret, showing that the suppression of PD with only 20% computational resource has minimum EI.

Since we do not utilize any information from EI during learning, we believe that the mismatching between the reward and cumulative regret comes from two reasons: (*i*) the correlation between P_β and EI is not completely perfect, and (*ii*) the designed reward function considers the regularization of the frequency values. This mismatching appears mainly with the extremely lower ϵ, which is acceptable. In addition, ϵ-NeuralTS with $\epsilon = 0.8$ still receives a lower cumulative regret versus NeuralTS, demonstrating the consistency of performance improvement on ϵ-NeuralTS via less exploration.

Overall, most of the settings except for $\epsilon = 0.0$ can converge below the dashed line, indicating that they can alleviate PD symptoms successfully because the dashed line is with cumulative regret $r_{(100)} = 0.1(\text{EI}) \times 100(\text{rounds})$; note that the healthy brains are with EI < 0.1. This observation effectively shows that we can consider less exploration carefully to achieve an improvement of NeuralTS. Therefore, we will further compare ϵ-NeuralTS with $\epsilon = 0.8$ with other methods.

D. ϵ-NeuralTS against CMAB Algorithms

In addition to considering the vanilla NeuralTS as our baseline, we also perform comparison to other existing CMAB algorithms, including linear bandit (*e.g.,* LinUCB [31] and LinTS [28]), generalized linear bandits (*e.g.,* UCB-GLM [46]), and neural bandits (*e.g.,* NeuralUCB [38] and Neural ϵ-greedy [47]). Note that the ϵ in Neural ϵ-greedy is not the same as the role of our ϵ in ϵ-NeuralTS. Instead, ϵ in Neural ϵ-greedy is for deriving a probability for randomly selecting action as exploration. Also, this probability for exploration will keep decreasing with increasing rounds so that the learning converges.

In Figure 6, we report the mean and the standard error of the cumulative regret of different algorithms over 10 runs. We demonstrate that our ϵ-NeuralTS with $\epsilon = 0.8$ is still competitive compared to the other algorithms. The results for linear bandit-based approaches (*i.e.,* LinUCB and LinTS) are the worst in both the reward and cumulative regret. UCB-GLM and vanilla NeuralTS perform well with high rewards while they receive relatively worse cumulative regrets, reflecting higher EI. We notice that NeuralUCB is the strongest baseline for our task, demonstrating similar reward and cumulative regret as ϵ-NeuralTS with $\epsilon = 0.8$ does.

We emphasize our contribution to improving the task performance of the vanilla NeuralTS using ϵ-exploring strategy so that the branch of NeuralTS can be competitive with

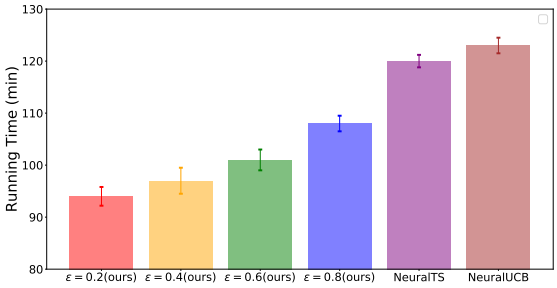


Fig. 7. Running time comparison of different ϵ for our ϵ-NeuralTS versus NeuralTS and NeuralUCB. Each trial takes 100 rounds of interaction with the computational BGM.

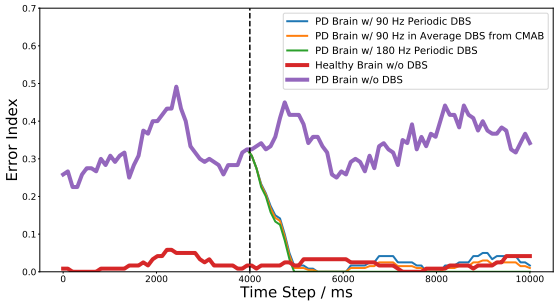


Fig. 8. Error Index (EI) over time in model PD brains without and with various types of stimulation, as well as model healthy brains.

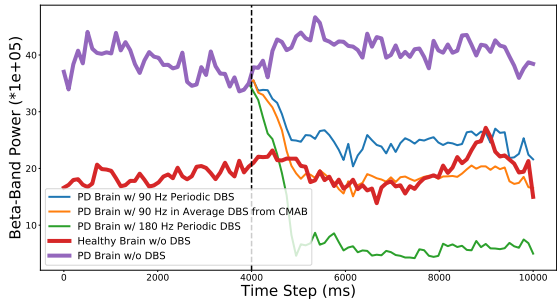


Fig. 9. Beta power spectral density (P_β) over time in model PD brains without and with various types of stimulation, as well as in healthy brains

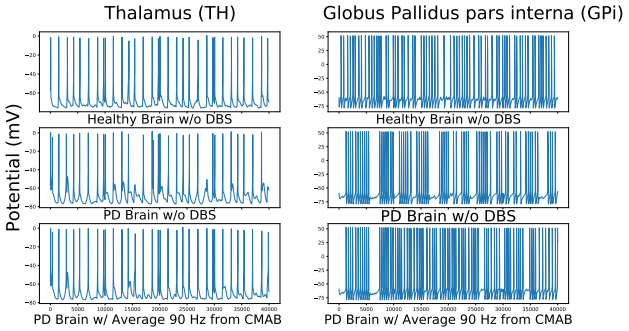


Fig. 10. Activity of model neurons in TH and GPi. The effects of patho-physiological patterns can be reduced using our ϵ-NeuralTS with an average frequency of 90 Hz (bottom row).

NeuralUCB in this task. Also, less exploration means that we reduce the risk of searching for unknown scenarios, which is critical for medical devices and procedures in the real world.

E. Relative Computation Time of ϵ-NeuralTS

Since the sampling with exploration in the vanilla NeuralTS requires computation including additional taking gradient descent w.r.t $\boldsymbol{\theta}$: $g(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x},\boldsymbol{\theta})$, which is described in (13), exploration reduction can also reduce the computation time during learning. We compare the running time of each trial under different ϵ against the vanilla NeuralTS and NeuralUCB; the results are summarized in Figure 7. Specifically, each trial takes 100 rounds of interaction with the computational BGM. We notice that NeuralTS runtime is smaller than NeuralUCB with the same number of rounds, which is consistent with the mathematical perspectives mentioned in Section II-C.

Also, reducing ϵ, the ϵ-NeuralTS run time will decrease. Since our best setting in the task of suppressing PD symptoms is with $\epsilon = 0.8$. We report that the running time is about 10% less than the standard NeuralTS with $\epsilon = 1.0$. Note that the interaction with computational BGM also occupies a huge portion of running time, which is the same for all algorithms.

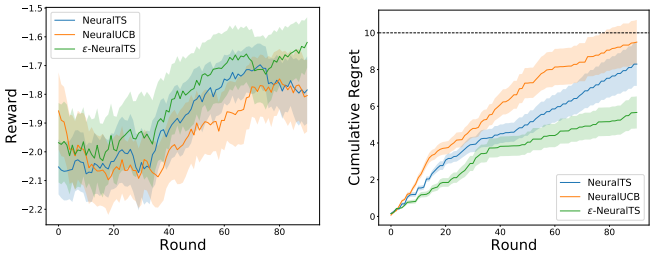
F. ϵ-NeuralTS against Classical Controllers

In addition to comparing with existing CMAB methods, we compared our ϵ-NeuralTS ($\epsilon = 0.8$) with the periodic controllers for which the stimulation pulses are equally spaced in terms of time steps. We firstly calculate the average stimulation frequency of ϵ-NeuralTS ($\epsilon = 0.8$) after convergence as about $k = 6$, *i.e.,* $90Hz$, which is smaller than the best setting of the vanilla NeuralTS in Table I. In other words, less exploration can also help reduce the frequency of the stimulation, improving energy efficiency and reducing side effects. To have a fair comparison, we mainly compared our CMAB-based controller with periodic cDBS at $90Hz$.

In Figure 8 and Figure 9, we evaluate the performance of our ϵ-NeuralTS controller online after learning. Specifically, the whole evaluation period is divided by a dashed line, indicating that all DBS controllers will be turned on to output their corresponding execution after 4000 rounds. Therefore, except for the healthy brain, all the other controllers start from the same oscillation with a higher EI and P_β.

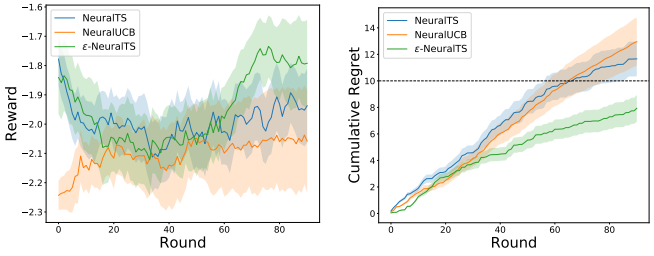
We observe that $90Hz$ frequency is still high and effective enough for periodic cDBS, so periodic DBS at $90Hz$ can easily reduce EI from the PD brain w/o DBS (purple curve) and even reaches a similar EI value as the periodic DBS at $180Hz$ (*i.e.,* maximum value in our setting). Therefore, the difference between our method (orange) and periodic DBS at $90Hz$ (blue) is not significant enough in Figure 8. However, we do improve P_β in Figure 9 compared with the periodic cDBS at $90Hz$. Our method can successfully achieve a similar P_β value as the healthy brain has, supporting that having varying stimulation frequency according to different observed context features can have better mitigation of PD symptoms even though the overall average frequency is the same.

The TH neuron activations and the firing pattern in GPi in BGM will be different between healthy brain and a PD brain without DBS [18]. In a healthy brain, the neuron



(a) Reward (5 steps) (b) Cumulative Regret (5 steps)

Fig. 11. Comparison of ϵ-NeuralTS with NeuralTS and NeuralUCB under 5 steps of delay. **Left:** rewards (higher is better) and **Right:** cumulative regret (lower is better) The total regret measures cumulative EI. Results are averaged over 10 runs with standard errors shown as shaded areas.



(a) Reward (10 steps) (b) Cumulative Regret (10 step)

Fig. 12. Comparison of ϵ-NeuralTS with NeuralTS and NeuralUCB under 10 steps of delay. **Left:** rewards (higher is better) and **Right:** cumulative regret (lower is better) The total regret measures cumulative EI. Results are averaged over 10 runs with standard errors shown as shaded areas.

activations of GPi and TH follow sporadic spiking at a stable firing rate. However, the brain affected by PD will result in pathological neuron activations within TH and GPi, which can be captured by reduced triggering potentials and clustered spiking, respectively. Therefore, in Figure 10, we visualize the activity of model neurons in TH and GPi with a healthy brain, a PD brain without DBS, and a PD brain stimulated with our CMAB (*i.e.,* ϵ-NeuralTS). We demonstrate that substantial pathophysiological patterns in the PD brain without DBS stimulation in the middle row of Figure 10 can be mitigated by our ϵ-NeuralTS, which makes the pattern of the activity in both GPi and TH much similar to that of the healthy brain.

G. Robustness of ϵ-NeuralTS (Reward Delay)

Although data-driven approaches for decision-making (e.g., RL and CMAB) can demonstrate impressive performance in the training environment, they may not always provide a robust solution to internal conditions and external disturbances [48], [49]. Specifically, this experiment is inspired by practical scenarios where the reward signals are delayed, due to various constraints when the algorithms are deployed in the real world [40]. We study the robustness of the two most competitive CMAB from Section V-D (i.e., ϵ-NeuralTS and NeuralUCB) when the rewards are delayed. Particularly, the bandit learner will not receive the reward right after taking an action. Instead, the rewards will arrive in batches when the algorithm updates its model.

In this evaluation, we only vary the batch size, which is the amount of the reward delay, considering 5 and 10 steps. Since Neural-TS is an instance of our ϵ-NeuralTS, we also include its results for comparison. We report the reward and cumulative regret of three methods in Figure 11 and Figure 12.

We firstly observe that NeuralTS outperforms NeuralUCB, which differs the standard setting without the reward delay. We notice that the gap between the vanilla NeuralTS and NeuralUCB in the standard task is small. Then according to [37], the core method for TS using randomized exploration encourages exploration between batches. Moreover, less exploration in ϵ-NeuralTS also reduces delayed explorative information, leading to a better performance in both the task reward and cumulative regret. Specifically, the increase in the cumulative regret with 5 steps reward delay is not obvious for ϵ-NeuralTS. When the amount of the reward delay is increased up to 10 steps, all of the learning rewards start to fluctuate, indicating that the reward delay does influence the learning process for all methods. Nevertheless, our ϵ-NeuralTS still has a relatively smaller cumulative regret (below the dashed line) in Figure 12 compared to the other two approaches.

VI. Conclusion

Existing commercial DBS devices only support pre-defined periodic stimulation with fixed high frequency. To address this limitation, RL-based approaches have been proposed to search for a flexible and efficient stimulation frequency according to the status of the brain. Yet, in general, the use of RL requires a huge amount of training data. In addition, the required computational resources for learning RL policies in the real world also serves as an obstacle for deployment. Thus, in this work, we formulate the treatment of PD symptoms using DBS as a CMAB problem so that we only consider single-step decision-making, getting rid of the heavy computation required for RL training. We then propose a novel method using ϵ-exploring strategy to reach more energy, sampling, and computationally efficient learning. Our method outperforms the existing CMAB baselines and results in a lower P_β compared to the periodic DBS with the same average stimulation frequency. The potential future direction is to investigate a better approximation of the mapping between P_β and the oracle EI so that the bandit learner will tend to select the action with much lower frequency but with efficiency without access to EI. In addition, how to adapt our proposed method to real patient data will also be an avenue for future work.

References

[1] C. Marras, J. Beck, and et al., "Prevalence of parkinson's disease across north america," in *NPJ Parkinson's disease*, 2018.
[2] A. L. Benabid, "Deep brain stimulation for parkinson's disease," in *Current opinion in neurobiology*, 2003, pp. 696–706.
[3] G. Deuschl, C. Schade-Brittinger, and et al., "A randomized trial of deep-brain stimulation for parkinson's disease," in *New England Journal of Medicine*, 2006, pp. 896–908.
[4] K. A. Follett, F. M. Weaver, and et al., "Pallidal versus subthalamic deep-brain stimulation for parkinson's disease," in *New England Journal of Medicine*, 2010, pp. 2077–2091.
[5] M. S. Okun, "Deep-brain stimulation for parkinson's disease," in *New England Journal of Medicine*, 2012, pp. 1529–1538.

[6] J. Pineau, A. Guez, R. Vincent, G. Panuccio, and M. Avoli, "Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach," in *Int. Journal of Neural Systems*, 2009, pp. 227–240.

[7] M. Beudel and P. Brown, "Adaptive deep brain stimulation in parkinson's disease," in *Parkinsonism & related disorders*, 2016, pp. 123–126.

[8] M. Arlotti, M. Rosa, and et al., "The adaptive deep brain stimulation challenge," in *Parkinsonism & related disorders*, 2016, pp. 12–17.

[9] M. Arlotti, L. Rossi, and et al., "An external portable device for adaptive deep brain stimulation (adbs) clinical research in advanced parkinson's disease," in *Medical engineering & physics*, 2016, pp. 498–505.

[10] L. S, P. A, and et al., "Adaptive deep brain stimulation in advanced parkinson disease," in *Ann Neurol*, 2013, pp. 449–457.

[11] S. Little, E. Tripoliti, and et al., "Adaptive deep brain stimulation for parkinson's disease demonstrates reduced speech side effects compared to conventional stimulation in the acute setting," in *J Neurol Neurosurg Psychiatry*, 2016, pp. 1388–1389.

[12] Q. Gao, S. L. Schmidt, K. Kamaravelu, D. A. Turner, W. M. Grill, and M. Pajic, "Offline policy evaluation for learning-based deep brain stimulation controllers," in *13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2022, pp. 80–91.

[13] Q. Gao, S. L. Schmidt, A. Chowdhury, G. Feng, J. J. Peters, K. Genty, W. M. Grill, D. A. Turner, and M. Pajic, "Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment," in *ACM/IEEE 14th International Conference on Cyber-Physical Systems (ICCPS)*, 2023, pp. 44–55.

[14] S. L. Schmidt, A. H. Chowdhury, K. T. Mitchell, J. J. Peters, Q. Gao, H.-J. Lee, K. Genty, S.-C. Chow, W. M. Grill, M. Pajic, and D. A. Turner, "At home adaptive dual target deep brain stimulation in Parkinson's disease with proportional control," *Brain*, vol. 147, no. 3, pp. 911–922, 12 2023.

[15] J. Habets, M. Heijmans, and et al., "An update on adaptive deep brain stimulation in parkinson's disease," in *Movement Disorders*, 2018, pp. 1834–1843.

[16] P. Sarikhani, H.-L. Hsu, and B. Mahmoudi, "Automated tuning of closedloop neuromodulation control systems using bayesian optimization," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022, pp. 1734–1737.

[17] V. Nagaraj, A. Lamperski, and T. I. Netoff, "Seizure control in a computational model using a reinforcement learning stimulation paradigm," in *International J. of Neural Sys*, 2017.

[18] Q. Gao, M. Naumann, I. Jovanov, V. Lesi, K. Kumaravelu, W. Grill, and M. Pajic, "Model-based design of closed loop deep brain stimulation controller using reinforcement learning," in *ACM/IEEE 11th Int. Conf. on Cyber-Physical Systems (ICCPS)*, 2020, pp. 108–118.

[19] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," in *Machine Learning*, 2002, pp. 235–256.

[20] L. Cannelli, G. Nuti, M. Sala, and O. Szehr, "Hedging using reinforcement learning: Contextual k-armed bandit versus q-learning," *The Journal of Finance and Data Science*, vol. 9, 2023.

[21] T. Y and et al., "Towards adaptive deep brain stimulation: clinical and technical notes on a novel commercial device for chronic brain sensing," in *Journal of Neural Engineering*, vol. 18, 2021.

[22] K. Kumaravelu, D. T. Brocker, and W. M. Grill, "A biophysical model of the cortex-basal ganglia-thalamus network in the 6-ohda lesioned rat model of parkinson's disease," in *Journal of computational neuroscience*, 2016, pp. 207–229.

[23] R. Q. So, A. R. Kent, and W. M. Grill, "Relative contributions of local cell and passing fiber activation and silencing to changes in thalamic fidelity during deep brain stimulation and lesioning: a computational modeling study," in *Journal of computational neuroscience*, vol. 32, 2012, pp. 499–519.

[24] I. Jovanov, M. Naumann, K. Kumaravelu, W. M. Grill, and M. Pajic, "Platform for model-based design and testing for deep brain stimulation," in *ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, April 2018, pp. 263–274.

[25] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," in *Biometrika*, 1933, pp. 285–294.

[26] D. T. Brocker and et al., "Optimized temporal pattern of brain stimulation designed by computational evolution," in *Science Translational Medicine*, 2017.

[27] T. Lattimore and C. Szepesv´ari, "Bandit algorithms," *Cambridge University Press*, 2018.

[28] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013, pp. 127–135.

[29] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," in *Foundations and Trends in Machine Learning*, 2011, pp. 1–122.

[30] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011.

[31] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. of the 19th Int. Conf. on World Wide Web*, 2010, pp. 661–670.

[32] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, "Parametric bandits: The generalized linear case," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010.

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

[34] L. Li, Y. Lu, and D. Zhou, "Provably optimal algorithms for generalized linear contextual bandits," in *International Conference on Machine Learning*, 2017.

[35] Q. Ding, C.-J. Hsieh, and J. Sharpnack, "An efficient algorithm for generalized linear bandit: Online stochastic gradient descent and thompson sampling," *ArXiv*, vol. abs/2006.04012, 2020.

[36] C. Riquelme, G. Tucker, and J. Snoek, "Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling." in *International Conference on Learning Representation (ICLR)*, 2018.

[37] W. Zhang, D. Zhou, L. Li, and Q. Gu, "Neural thompson sampling," in *International Conference on Learning Representations*, 2021.

[38] D. Zhou, L. Li, and Q. Gu, "Neural contextual bandits with ucb-based exploration," in *International Conference on Machine Learning*, 2020, pp. 11 492–11 502.

[39] P. Xu, Z. Wen, H. Zhao, and Q. Gu, "Neural contextual bandits with deep representation and shallow exploration," in *International Conference on Learning Representations*, 2022.

[40] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2011, pp. 2249–2257.

[41] H.-L. Hsu, Q. Huang, and S. Ha, "Improving safety in deep reinforcement learning using unsupervised action planning," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5567–5573.

[42] P. Sarikhani, H.-L. Hsu, O. Kara, J. K. Kim, H. Esmaeilzadeh, and B. Mahmoudi, "Neuroweaver: a platform for designing intelligent closed-loop neuromodulation systems," *Brain Stimulation*, vol. 14, no. 6, p. 1661, 2021.

[43] T. Jin, X. Yang, X. Xiao, and P. Xu, "Thompson sampling with less exploration is fast and optimal," in *International Conference on Machine Learning*, 2023.

[44] T. Jin, H.-L. Hsu, W. Chang, and P. Xu, "Finite-time frequentist regret bounds of multi-agent thompson sampling on sparse hypergraphs," in *Annual AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[45] A. Guez, R. D. Vincent, M. Avoli, and J. Pineau, "Adaptive treatment of epilepsy via batch-mode reinforcement learning," in *AAAI*, 2008, pp. 1671–1678.

[46] L. Li, Y. Lu, and D. Zhou., "Provably optimal algorithms for generalized linear contextual bandits," in *International Conference on Machine Learning*, 2017, pp. 2071–2080.

[47] C. Riquelme, G. Tucker, and J. Snoek., "Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling," in *International Conference on Learning Representations*, 2018.

[48] J. Dong, H.-L. Hsu, Q. Gao, V. Tarokh, and M. Pajic, "Robust reinforcement learning through efficient adversarial herding," *https://arxiv.org/abs/2306.07408*, 2023.

[49] H.-L. Hsu, H. Meng, S. Luo, J. Dong, V. Tarokh, and M. Pajic, "Reforma: Robust reinforcement learning via adaptive adversary for drones flying under disturbances," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

TABLE II
THE NOTATION USED IN THE PAPER.

Symbol	Description
n	number of neurons in each sub-region q in the brain
ν_j^q	j^{th} neuron's electrical potential with the corresponding sub-region $q \in \{STN, GPe, GPi, TH\}$
$\boldsymbol{v}^q(t)$	vector of electrical potential with the corresponding sub-region $q \in \{STN, GPe, GPi, TH\}$ at time t
EI	Error Index: portion of erroneous TH neuron activations in response to SMC inputs
P_β	Beta-band Power Spectral Density
T	maximum number of rounds defined as a horizon in multi-armed bandit problem
K	number of arms
A	action set
a_t	action in time t in multi-armed bandit problem
u_t	mapped action from a_t to computational BGM at time t
s_t	context feature in time t
x_t	context vector transformed from s_t in time t
π	policy
R_{t,a_t}	reward in time t corresponding to a_t
D	history storing a sequence of tuple (a_t, R_t)
T_w	window of size
$r_(T)$	cumulative regret up to time T
$\nu > 0$	exploration variance
λ	regularization parameter
ϵ	exploration probability

APPENDIX

To improve readability of the paper, we provide a summary of the employed notation in Table II.