Neural Constrained Combinatorial Bandits

Shangshang Wang, Simeng Bian, Xin Liu, Ziyu Shao*
School of Information Science and Technology, ShanghaiTech University, China
Email: {wangshsh2, biansm, liuxin7, shaozy}@shanghaitech.edu.cn

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10228958

***Abstract*—Constrained combinatorial contextual bandits have emerged as trending tools in intelligent systems and networks to model reward and cost signals under combinatorial decision-making. On one hand, both signals are complex functions of the context, *e.g.*, in federated learning, training loss (negative reward) and energy consumption (cost) are nonlinear functions of edge devices' system conditions (context). On the other hand, there are *cumulative* constraints on costs, *e.g.*, the accumulated energy consumption should be budgeted by energy resources. Besides, real-time systems often require such constraints to be guaranteed *anytime or in each round*, *e.g.*, ensuring anytime fairness for task assignment to maintain the credibility of crowdsourcing platforms for workers. This setting imposes a challenge on how to simultaneously achieve reward maximization while subjecting to anytime cumulative constraints. To address such challenge, we propose a primal-dual algorithm (*Neural-PD*) whose primal component adopts *multi-layer perceptrons* to estimate reward and cost functions, and its dual component estimates the Lagrange multiplier with the *virtual queue*. By integrating neural tangent kernel theory and Lyapunov-drift techniques, we prove Neural-PD achieves a *sharp regret bound* and a *zero constraint violation*. We also show Neural-PD outperforms existing algorithms with extensive experiments on both synthetic and real-world datasets.**

I. Introduction

Thanks to their immense simplicity and wide applicability, bandits have emerged as a canonical online learning tool for decision-making problems under uncertainty [1], *e.g.*, client selection in federated learning [2], and task assignment in crowdsourcing systems [3]. In the classical stochastic multi-armed bandits (MAB) model, there are two interleaved processes: 1) The *decision process* where the agent selects one arm periodically; 2) The *feedback process* where the agent receives a random reward after arm selection. Particularly, the reward follows some unknown distribution with *implicit* dependences on the environment. The goal of the agent is to maximize the accumulated rewards over a given horizon.

Despite MAB's popularity, this classical tool often fails to be applied to complicated real-world scenarios. To capture practical needs, two parallel threads in the field generalize MAB by introducing modeling flexibility in both decision and feedback processes: *Combinatorial bandits* extend the decision process to allow multiple selection for a subset of arms [4]; *Contextual bandits* provide the feedback process with more fine-grained modeling of rewards based on the context [5]. Specifically, the context captures *explicit* dependences and side information from the environment that affect the reward distribution. The famous linear (contextual) bandits assume mean reward in each round is a linear function of the context [6]. Though rich in both theory and applications [7], the linear function assumption is often infeasible in practice. For example in federated learning, the edge server (agent) periodically selects a subset of clients (arms) to join the collaborative training. The training loss (negative reward) is a complex function of factors like heterogeneity and size of datasets on clients (context) [8]. Another example is dynamic pricing in online marketplaces. Specifically, the marketer (agent) dynamically adjusts the price (arm) of goods for customers. The revenue (reward) is a nonlinear function of influences like advertising and word-of-mouth effects (context) [9]. Note that such complex function for reward (*i.e.*, the reward function) is often *unknown* to the agent and *general nonlinear* (*e.g.*, nonsmooth and nonconvex) of the context in practice. This makes traditional contextual bandits (*e.g.*, linear and kernel bandits) ineffective because the structure of rewards is not consistent with these assumptions in practice. That is, linear bandits rely on the linearity of the reward function; and kernel bandits require the reward function belongs to some *Reproducing Kernel Hilbert Space* (RKHS) [10]. To address this challenge, *neural bandits* have recently gained popularity to learn the reward function with deep neural networks, where *no* special reward function structure is required [11]–[14]. With strong representation ability, neural-network-based estimator in neural bandits can learn the reward function without domain knowledge or prior information [15]. Combining advances in decision and feedback processes, the *neural combinatorial bandits* model has been studied in works [8], [16].

Besides the reward, selecting an arm usually incurs the *cost*, which may also have complex structure, and we often have operational *constraints* on the cost in practical systems. For example, in crowdsourcing systems, the energy consumption of workers (cost) is affected nonlinearly by the characteristic of allocated task and energy management strategies of workers' devices [31] (context). Notably, constraints on such costs are often *cumulative* ones [23], *e.g.*, the accumulated energy consumption should not exceed the energy budget of workers (*i.e.*, budget constraints) to retain energy efficiency. Conventionally, such constraints are imposed in a time-average sense and need to be satisfied at the end of the time horizon [24]. Well-studied examples are bandits with cumulative budget constraints [28]; bandits with knapsacks where the decision-making process terminates when the total budget has been consumed [20],

This work was partially supported by Natural Science Foundation of Shanghai under Grant 22ZR1441700 and Shanghai Sailing Program under Grant 22YF1428500. (*Corresponding author: Ziyu Shao)

TABLE I
COMPARISON BETWEEN NEURAL CONSTRAINED COMBINATORIAL BANDITS AND MODELS IN PREVIOUS WORKS.

	Stochastic Bandit Models				Arm Selection Manners		Types of Cumulative Constraints		
Models	MAB	Linear	Kernel	Neural	Single	Combinatorial	None†	Horizon-dependent	Anytime
Ours				✓		✓			✓
[8], [16]⋆				✓		✓	✓		
[11]–[14], [17]				✓	✓		✓		
[18]			✓		✓				✓
[19]		✓			✓				✓
[20], [21]		✓			✓			✓	
[22]	✓					✓			✓
[23], [24], [25], [26]	✓					✓		✓	
[27]	✓				✓				✓
[28]–[30]	✓				✓			✓	

Superscript ⋆: there is no theoretical guarantee for works [8], [16]; Superscript †: there is no constraint, *i.e.*, the related model is unconstrained.

[21], [29], [30]; and bandits with asymptotic constraints under which either the average number of each arm selection is larger than a threshold in the long run [25] or the long-term time-average cost is budgeted [26]. We denote the above constraints collectively as *horizon-dependent cumulative constraints*.

Compared with the conventional horizon-dependent cumulative constraints, their generalized *anytime* versions recently attract increasing attention as a stronger and more practical notion [18], [19], [22], [27]. Specifically, the anytime constraints should be guaranteed not only by the end of the horizon but also in *each and every round*. In other words, the practical interpretation of ensuring anytime constraints is the guarantee of *short-term* system performances [32]. For example, in a spatial crowdsourcing platform where spatially distributed tasks are periodically assigned to workers of varying locations [16]. In this case, guaranteeing the anytime fairness of tasks allocated to each worker is crucial. This promises workers a minimal number of profitable tasks as a reliable source of revenue, thus maintaining the credibility and prosperity of the platform.

Given such discussion, we study the neural combinatorial bandits under anytime cumulative constraints, denoted as *neural constrained combinatorial bandits*. With this model, we essentially face the problem that *"Can we achieve both sublinear regret and minimal constraint violation?"* To address this problem, we propose a neural-network-based algorithm with primal-dual optimization called *Neural-PD*. Neural-PD integrates 1) overparameterized multi-layer perceptrons (MLPs) as approximators for the reward and cost functions during online learning; and 2) primal-dual methods to optimize rewards and constraint violations. Such integration is highly non-trivial with the presence of uncertainty. On one hand, the efficiency of primal-dual optimization should be guaranteed *anytime* even with inaccurate estimations (*e.g.*, in the exploration phase of online learning). On the other hand, the online feedback for estimations of both reward and cost is determined by the optimization decisions. This requires a dedicated design to balance reward maximization and violation minimization. Such coupling between learning and optimization imposes challenges in terms of both algorithm design and theoretical analysis. Our main contributions are as follows:

- **Modeling.** We study the neural constrained combinatorial bandits with general nonlinearity. That is, except for boundedness, we *do not assume any special structures of reward and cost functions*. Besides, our modeling of anytime cumulative constraints can capture various constraints like fairness and budget constraints.
- **Algorithm Design.** We propose Neural-PD whose primal component uses *optimistic-pessimistic learning* to handle the uncertainty; the dual component leverages the *virtual queue* to track the constraint violation. It can achieve maximum rewards while keeping violations minimal.
- **Theoretical Analysis.** We integrate the neural tangent kernel (NTK) theory [33] into Lyapunov-drift techniques [34] and show Neural-PD achieves a *sublinear regret* at order $\widetilde{O}\left(\frac{M^3\tilde{d}^2}{\delta^3}+\frac{M^{5/2}\tilde{d}\sqrt{\tau}}{\delta}\right)$ until any round τ in the horizon T and obtains *zero constraint violation* when $\tau > O\left(\frac{M^3\tilde{d}^2\log^2 T}{\delta^4}\right)$, where M is the arm selection number, $\tilde{d}$ is the effective dimension of NTK matrix on contexts, and δ is the Slater's constant. Note that our regret bound (in terms of T) is sharp by matching the lower bound $\Omega(\sqrt{T})$ up to logarithmic factors [1].
- **Applications.** We demonstrate that our model and algorithm design can be potentially applied to a broad range of real-world applications. Through simulation on both synthetic dataset and practical case study of crowdsourcing, we show the effectiveness of Neural-PD compared with state-of-the-art unconstrained and constrained baselines.

Main Notations. We let $[\cdot]^+ \triangleq \max\{\cdot, 0\}$; Let $[k]$ be the set $\{1, \ldots, k\}$ for some positive integer k; Let $\widetilde{O}(g(n)) \triangleq O(g(n)\log^k n)$ with $k > 0$ to hide the logarithmic factors; Let $\mathrm{poly}(\cdot)$ denote the polynomial function in terms of inputs; Let $\mathrm{vec}(\cdot)$ be the vectorization for a matrix, *e.g.*, for $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{mn}$ is the column vector obtained by stacking the columns of $\mathbf{A}$ on top of each other; For vector $\mathbf{x}$ and positive definite matrix $\mathbf{A}$ of proper dimensions, $\|\mathbf{x}\|_{\mathbf{A}} \triangleq \sqrt{\mathbf{x}^\top\mathbf{A}\mathbf{x}}$.

II. Related Works

We present the comparison with related works in this section and a corresponding summary is shown in Table I.

Neural Bandits. The traditional approach to contextual bandits with general nonlinearity is by nonparametric models via RKHS [35], *i.e.*, kernel bandits [10]. As a new entrant to the field, neural bandits offer a solution with the representation ability of deep neural networks and advancements in NTK theory [33], [36], [37]. Work [11] first utilizes the NTK-based approximation on neural networks and presents a provably efficient MLP-based contextual bandit algorithm based on the upper confidence bound (UCB), *i.e.*, NeuralUCB. Follow-up work [12] further extends to the Thompson Sampling (TS) with NeuralTS, whose neural network estimator mainly focuses on approximating the posterior distribution of reward. Another work [13] considers the offline neural bandits where a neural-network-based lower confidence bound method (*i.e.*, NeuralLCB) is adopted to pessimistically approach the unknown. Work [14] proposes the EE-Net which utilizes additional neural networks to determine exploitation, exploration, and decision-making during learning. Work [17] studies the neural bandit problem with general smooth activation functions. Note that all above works consider the canonical single arm selection. Works [8], [16] extend the neural bandits to the combinatorial selection setting and apply them to client selection in federated learning and task assignment in mobile crowdsourcing, respectively. However, both works focus on numerical simulations while lacking theoretical guarantees. Besides, all the above works assume the impractical *unconstrained* settings that hinder further real-world implementation. In comparison, we consider their practical generalization with combinatorial decision-making under general anytime cumulative constraints with complex structures. Particularly, performances of the proposed algorithm are justified with both a theoretical guarantee and extensive numerical experiments.

Bandits with Anytime Cumulative Constraints. Compared with the hard constraints that have to be satisfied in each round with high probability [38], [39], anytime cumulative constraints impose *soft* constraints on the cumulative costs and allow *violations* in any round. They are a generalization of the horizon-dependent cumulative constraints that only guarantee the cumulative constraints over the entire horizon [24], [25]. Work [40] studies on the specific anytime fairness constraints in adversarial contextual bandits. Later works [22], [27] extend the notion of anytime fairness by introducing an unfairness tolerance and they study the regret-fairness tradeoff in stochastic MAB. Our model differs from these works in two aspects: 1) we study the contextual bandits which are a generalized setting of their MAB models; 2) we consider general anytime cumulative constraints rather than the special anytime fairness constraint. Similar to our settings, work [19] studies stochastic linear bandits with anytime cumulative constraints via a pessimistic-optimistic algorithm (APOA). APOA manages to handle linear constraints (if the cost function is known, general nonlinear constraints) with a zero constraint violation.

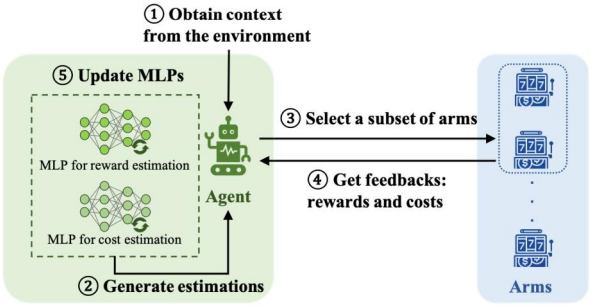


Fig. 1. Illustration of Neural Constrained Combinatorial Bandits Model. In each round, the agent first generates MLP-based estimation with the obtained context. Next, with such estimation, the agent selects a subset of arms and updates the MLPs with feedbacks.

Another related work [18] systematically analyzes the kernel bandits with cumulative constraints (*a.k.a.* constrained kernel bandits, CKB) through different exploration strategies and achieves zero violations. The core of CKB algorithms boils down to methods based on Gaussian processes regression with a *predefined kernel* under the primal-dual optimization framework.

Compared with the kernel-based methods (*i.e.*, CKB algorithms), our Neural-PD has the following advantages:

1) Overparameterized neural networks can represent richer smooth functions than kernel-based methods [41] and are more robust to outliers (*e.g.*, corrupted contexts [42] during wireless transmission) with regularization [43].
2) CKB algorithms rely on a handpicked kernel function to learn the features, which may be non-trivial in practice without prior information. Neural-PD automatically learns the features via the dynamic NTK from online data in an end-to-end manner. On the contrary, our neural-based method may have marginal effects compared with kernel-based methods when the underlying data distribution is known (despite unlikely).
3) Neural-PD performs better than CBK algorithms numerically with complex function structures to learn in both synthetic and real-world cases (see Section VI).

Following this line of bandit research, we generalize the constrained settings from MAB, linear, and kernel to neural bandits, where we make *no* assumption on structures of reward and cost functions and rely on neural networks for online estimation. Note that cumulative constraints are also studied in the field of online convex optimization (OCO) [44], especially the long-term constraints [45], [46]. The main difference between such works and ours is that OCO assumes *full feedback* in each round while bandit learning only observes the feedback of selected arms (*i.e.*, *bandit feedback*).

III. System Model & Problem Formulation

Before we embark on the description of our modeling, refer to Fig. 1 for a brief overview of main model components.

Stochastic Contextual Bandits. We consider a stochastic contextual bandit model with an agent and a set of N arms,

denoted as $\mathcal{N} \triangleq [N]$. In each round t within the given horizon T, the agent observes a context $c(t)$ drawn from the context set $\mathcal{C}$ with an unknown distribution. The contexts $\{c(t)\}_t$ are independent and identically distributed (*i.i.d.*) across rounds.

Combinatorial Selection. In each round, the agent selects a subset of arms (*i.e.*, a super arm). We restrict the size of super arms to be exactly[1] $M \leq N$. Let F be the set of all *feasible super arms*, *i.e.*, $F \triangleq \{S \subseteq \mathcal{N} : |S| = M\}$, where $|S|$ denotes the cardinality of subset S. In round t, we denote the selected super arm as $S(t)$ and use an indicator $X_j(t)$ to represent whether the agent selects arm j, *i.e.*, $X_j(t) = 1, \forall j \in S(t)$ and zero otherwise. Therefore, $\sum_{j \in S(t)} X_j(t) = M, \forall t \in [T]$.

General Nonlinear Rewards & Costs. After arm selection in round t, the agent receives a reward $R(c(t), j)$ and a cost $U(c(t), j)$ for each arm $j \in S(t)$. Both the reward and the cost depend on the context and arm index. Specifically,

$$R(c(t), j) \triangleq r(c(t), j) + \eta(t),$$
$$U(c(t), j) \triangleq u(c(t), j) + \xi(t),$$

where $\forall c(t) \in \mathcal{C}, j \in \mathcal{N}$, the mean reward $r(c(t), j) \triangleq \mathbb{E}[R(c(t), j)] \in [0, 1]$ and the mean cost $u(c(t), j) \triangleq \mathbb{E}[U(c(t), j)] \in [0, 1]$; $\eta(t), \xi(t)$ are additive noises following zero-mean 1-sub-Gaussian distributions[2]. We emphasize that both the (mean) reward function $r(\cdot, \cdot)$ and the (mean) cost function $u(\cdot, \cdot)$ are *unknown* to the agent and are *general nonlinear* functions of (context, arm index) pair. As concrete examples, the functions may vary from linear, generalized linear, to kernel functions with a bounded RKHS norm, or even Gaussian processes [48].

Problem Formulation. As standard in the combinatorial setting, we aim to maximize cumulative compound rewards over the horizon. During each round, the compound reward is directly formed by the summation of individual rewards of arms in the selected super arm[3].

Considering that we always select a super arm of size M and subject to the anytime cumulative constraints, the problem is formulated as follows:

$$\begin{aligned} &\max_{\{X_j(t)\}_{j,t}} \mathbb{E}\left[\sum\nolimits_{t=1}^{T} \sum\nolimits_{j \in \mathcal{N}} R(c(t), j) X_j(t)\right] \\ &\text{s.t.} \sum\nolimits_{j \in S(t)} X_j(t) = M, S(t) \in F, \forall t; \\ &\mathbb{E}\left[\sum_{t'=1}^{\tau} \sum_{j \in \mathcal{N}} U(c(t'), j) X_j(t')\right] \leq \mathbb{E}\left[\sum_{t'=1}^{\tau} B(t')\right], \forall \tau, \end{aligned} \tag{1}$$

where the budget $B(t) \in [0, b_{\max}]$ with $\mathbb{E}[B(t)] = b, \forall t \in [T]$. For simplicity, we consider constraints for one type of cost $U(\cdot, \cdot)$. Our setting can be readily extended with an arbitrarily finite number to simultaneously model various operational constraints like fairness and budget constraints in practical systems [19]. See Section V-B for more discussion on specific operational constraints and theoretical corollaries on them.

[1]Our model can be extended to combinatorial semi-bandits with sleeping arms where the size of super arms is bounded instead of fixed and only a subset of arms is available to be selected in each round [22], [25].

[2]A random variable X follows the zero-mean σ-sub-Gaussian distribution if $\mathbb{E}[\exp(tX)] \leq \exp(t^2\sigma^2/2), \forall t \in \mathbb{R}$ [47].

[3]It requires non-trivial modifications to our model and the theoretical analysis to extend our setting to the general combinatorial setting where the compound reward is not a simple summation of individual rewards. The algorithm design for this extension can be an interesting future work.

IV. Algorithm Design

We first introduce a deterministic (offline) version of the problem (1) to illustrate the main idea behind our algorithm design and then provide the formal statements in **Algorithm 1**.

A. Lagrange-Based Problem Decoupling

Note that problem (1) is essentially a stochastic constrained optimization problem. To effectively solve it, we start with a deterministic variant of problem (1) that replaces all the random variables with their expectations. Further, we construct a dual problem of the deterministic variant with the Lagrangian dual method to handle the anytime cumulative constraint. Finally, we derive a decoupled subproblem based on the context obtained in each round. This allows us to focus on a simplified and decoupled version of the original problem (1), based on which we conduct primal-dual optimization.

Tightened Problem. We consider a deterministic baseline problem in Definition 1. Different from the conventional setup, we additionally introduce a positive *tightness* constant ϵ for the anytime cumulative constraints. ϵ decisively determines the performance of Neural-PD (see Remark 3 and Section V).

Definition 1 (Tightened Problem)**.**

$$\begin{aligned} &\max_{\{\mathbf{q}_c\}_{c \in \mathcal{C}}} \sum\nolimits_{c \in \mathcal{C}} p_c \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} r(c, j) \\ &\text{s.t.} \sum\nolimits_{S \in F} q(c, S) = 1, \forall c \in \mathcal{C}; q(c, S) \geq 0, \forall c \in \mathcal{C}, \forall S \in F; \\ &\sum\nolimits_{c \in \mathcal{C}} p_c \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} u(c, j) + \epsilon \leq b, \end{aligned}$$

where p_c is the probability of observing context c; we denote $\mathbf{q}_c \triangleq [q(c, S), \forall S \in F]$ with $q(c, S)$ as the probability of choosing super arm S given context $c \in \mathcal{C}$.

Dual Problem. We apply the Lagrangian dual method to particularly handle the anytime cumulative constraints and obtain the following dual problem with an augmented objective:

$$\begin{aligned} &\max_{\{\mathbf{q}_c\}_{c \in \mathcal{C}}} \sum\nolimits_{c \in \mathcal{C}} p_c \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} r(c, j) \\ &- \zeta \left(\sum\nolimits_{c \in \mathcal{C}} p_c \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} u(c, j) - b + \epsilon\right) \\ &\text{s.t.} \sum\nolimits_{S \in F} q(c, S) = 1, \forall c \in \mathcal{C}; q(c, S) \geq 0, \forall c \in \mathcal{C}, \forall S \in F, \end{aligned}$$

where $\zeta > 0$ is the Lagrange multiplier associated with the anytime cumulative constraints. Suppose we know and fix ζ, solving the above problem is equivalent to solving $|\mathcal{C}|$ separate subproblems in (2), one for each context $c \in \mathcal{C}$, because variables $\{\mathbf{q}_c\}_{c \in \mathcal{C}}$ are coupled only through the super arms.

$$\begin{aligned} &\max_{\{q(c,S)\}_{S \in F}} \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} r(c, j) \\ &- \zeta \left(\sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} u(c, j) - b + \epsilon\right) \\ &\text{s.t.} \sum\nolimits_{S \in F} q(c, S) = 1, q(c, S) \geq 0, \forall S \in F. \end{aligned} \tag{2}$$

Decoupled Subproblem. Since problem (2) is a linear programming problem, one of the optimal solutions is $q(c,S)=1$ for some $S=S^*$ and zero otherwise. Therefore, we have the decoupled problem for any context $c\in\mathcal{C}$. (This does not mean we have to solve subproblems for each context in each round but only for the context observed in the current round.)

$$\max_{S\in F}\sum\nolimits_{j\in S} r(c,j)-\zeta\left(\sum\nolimits_{j\in S}u(c,j)-b+\epsilon\right). \quad (3)$$

Remark 1. Through the above derivation, we efficiently handle the anytime cumulative constraints with only the greedy selection in (3). This demonstrates the efficiency and superiority of primal-dual optimization compared with projected-based methods (*e.g.*, online projected gradient descent). Specifically, our algorithm avoids the heavy computational overhead (*e.g.*, when $u(\cdot,\cdot)$ is a complex nonlinear function) to project the arm selection decisions into a feasible set formed by the constraints. Besides, our algorithm does not require the knowledge of the probability distribution of the context $c\in\mathcal{C}$. This makes our method feasible and efficient in practice where the context is drawn from some unknown and complex distribution.

B. The Neural-Network-Based Primal-Dual Algorithm

Note that both the primal variables (*i.e.*, $r(c,j)$ and $u(c,j)$), and the dual variable (*i.e.*, ζ) are unknown *apriori* in the decoupled subproblem (3). To address this issue, we present the following online procedure that combines the primal-dual method and neural networks to estimate them on the fly. The procedure is summarized as our *neural-network-based primal-dual algorithm (Neural-PD)*.

Primal Variable Estimation. Since $r(c,j)$ and $u(c,j)$ have general nonlinear structures, we adopt estimation with neural networks as in Definition 2.

Definition 2 (Neural Network Architecture)**.** Let f_r, f_u be *overparameterized* multi-layer perceptrons (MLPs) with depth L and width m for each hidden layer to represent functions r, u, respectively[4]. By saying overparameterized, we mean the setting where the number of network parameters is larger than the number of training data points. f_r and f_u are defined as

$$f_r(x|\boldsymbol{\theta}_r)\triangleq\sqrt{m}\mathbf{W}_r^L\sigma(\mathbf{W}_r^{L-1}\sigma(\cdots\sigma(\mathbf{W}_r^1x)));$$
$$f_u(x|\boldsymbol{\theta}_u)\triangleq\sqrt{m}\mathbf{W}_u^L\sigma(\mathbf{W}_u^{L-1}\sigma(\cdots\sigma(\mathbf{W}_u^1x))),$$

where $\boldsymbol{\theta}_r,\boldsymbol{\theta}_u$ are stacked by $\mathbf{W}_r^l,\mathbf{W}_u^l,\forall l\in[L]$, respectively; Given $x\in\mathbb{R}^d$, $\mathbf{W}_r^1,\mathbf{W}_u^1\in\mathbb{R}^{m\times d}$, $\mathbf{W}_r^l,\mathbf{W}_u^l\in\mathbb{R}^{m\times m}, 2\le l\le L-1$, and $\mathbf{W}_r^L,\mathbf{W}_u^L\in\mathbb{R}^{m\times 1}$; $\sigma(\cdot)$ denotes the ReLU activation function[5]. For convenience, we define the gradients with respect to the neural network parameters as

$$g_r(x|\boldsymbol{\theta}_r)=\nabla_{\boldsymbol{\theta}_r}f_r(x|\boldsymbol{\theta}_r),\ g_u(x|\boldsymbol{\theta}_u)=\nabla_{\boldsymbol{\theta}_u}f_u(x|\boldsymbol{\theta}_u).$$

Remark 2. Motivated by the neural tangent kernel (NTK) theory, we restrict our neural network architecture to the special MLPs [33]. Specifically, by the NTK theory, the training dynamics of overparameterized MLPs with a large width can be characterized by a corresponding NTK matrix (more details in Section V). In other words, to achieve provable efficiency, the setup of MLPs (including later initialization and update) is designed based on the characteristics of the NTK matrix.

$\boldsymbol{\theta}_r,\boldsymbol{\theta}_u$ follow similar initialization and update procedures. We illustrate with $\boldsymbol{\theta}_r$ in Definition 3 and 4, respectively.

Definition 3 (Neural Network Initialization)**.** Initialize $\boldsymbol{\theta}_r$ with $\boldsymbol{\theta}_{r,0}=(\mathrm{vec}(\mathbf{W}_r^1);\cdots;\mathrm{vec}(\mathbf{W}_r^L))\in\mathbb{R}^p$ with $p=dm+m^2(L-1)+m$, where for each $1\le l\le L-1$, $\mathbf{W}_r^l=(\mathbf{W},\mathbf{0};\mathbf{0},\mathbf{W})$, each entry of $\mathbf{W}$ is generated independently from $\mathcal{N}(0,4/m)$; $\mathbf{W}_r^L=(\mathbf{w}^\top,-\mathbf{w}^\top)$, each entry of $\mathbf{w}$ is generated independently from Gaussian $\mathcal{N}(0,2/m)$.

Definition 4 (Neural Network Update)**.** In round t, $\boldsymbol{\theta}_r$ is updated by $\boldsymbol{\theta}_{r,t}$, *i.e.*, the optimal solution to the loss function for neural network training. The loss function is defined as

$$\mathcal{L}(\boldsymbol{\theta})\triangleq\frac{1}{2}\sum\nolimits_{\tau=1}^{t}\sum\nolimits_{j\in S(\tau)}\Big(f_r(c(\tau),j|\boldsymbol{\theta})-R(c(\tau),j)\Big)^2+\frac{m\lambda}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}_{r,0}\|_2^2,$$

where λ is the regularization parameter. We adopt gradient-based methods (*e.g.*, Adam [49]) to optimize the loss function for J steps with step size η:

$$\boldsymbol{\theta}_{r,t}=\boldsymbol{\theta}^{(J)},\boldsymbol{\theta}^{(j+1)}=\boldsymbol{\theta}^{(j)}-\eta\nabla\mathcal{L}(\boldsymbol{\theta}^{(j)}),$$

where $j\in\{0,\ldots,J-1\}$ and we set $\boldsymbol{\theta}^{(0)}=\boldsymbol{\theta}_{r,0}$.

With MLPs f_r,f_u, we construct the neural-network-based estimation $\widehat{R}(\cdot,\cdot)$ and $\widehat{U}(\cdot,\cdot)$ for reward and cost functions as the *primal update* in Algorithm 1. Note that $\widehat{R}(c(t),j)$ and $\widehat{U}(c(t),j)$ are formed in the UCB-style. Their differences lie in that $\widehat{R}(c(t),j)$ is an *optimistic* estimation of $r(c,j)$ since we aim to *maximize* the reward; while $\widehat{U}(c(t),j)$ is a *pessimistic* of $u(c,j)$ since we aim to *minimize* the constraint violation[6].

Similarly, our algorithm design can be extended with 1) Thompson-Sampling-style estimation, *i.e.*, NeuralTS [12]; and 2) UCB-style estimation as EE-Net [14], *i.e.*, exploitation and exploration are determined by two separate MLPs.

Dual Variable Estimation. For the dual variable ζ, we iteratively update its estimation $Q(t)$ as follows:

$$Q(t{+}1)\triangleq\Big[Q(t)+\sum\nolimits_{j\in\mathcal{N}}\widehat{U}(c(t),j)X_j(t)-B(t)+\epsilon_t\Big]^+. \quad (4)$$

From the perspective of Lyapunov optimization [34], $Q(t)$ is regarded as the *virtual queue*. It increases when the estimated cost exceeds the budget $B(t)$, and decreases otherwise. In other words, $Q(t)$ keeps track of the "estimated cumulative constraint violation" until round t. Note that there are two key differences from the canonical virtual queue update: 1) the MLP-based estimation of cost is adopted instead of assuming the knowledge of cost beforehand and 2) a time-varying ϵ_t

[4]For the simplicity of notation, we assume the width of each layer is m and that both MLPs share the same set of hyperparameters, *e.g.*, m, L, λ, η, J in Algorithm 1. In practice, their hyperparameters may vary from each other.

[5]The activation functions can be extended to general smooth ones [17].

[6]Though the form of $\widehat{U}(c(t),j)$ is similar to that in NeuralLCB [13] (which also adopts pessimistic estimation), NeuralLCB is based on a different design principle, *i.e.*, conservative learning in offline settings [50].

Algorithm 1: The Neural-Network-Based Primal-Dual (Neural-PD) Algorithm

Initialization:

$T, M, f_r, f_u, \boldsymbol{\theta}_{r,0}, \boldsymbol{\theta}_{u,0}, \lambda, \eta, J$, Scaling factors $\{\gamma_t\}_t$;

$Q(1) \leftarrow 0, \boldsymbol{\Sigma}_{r,0} \leftarrow \lambda \mathbf{I}, \boldsymbol{\Sigma}_{u,0} \leftarrow \lambda \mathbf{I}$.

for $t \in \{1, \cdots, T\}$ **do**

%% The agent performs all the following steps.

Observation: Obtain context $c(t)$ and budget $B(t)$.

Parameter Setup: Set tunable parameter

$$\epsilon_t \leftarrow O(\log(1+T)/\sqrt{t}),$$
$$V_t \leftarrow O(\log(1+T)\sqrt{t})$$

as in Theorem 1.

Gradient Calculation: For each arm $j \in \mathcal{N}$,

$$g_{r,j} \leftarrow g_r(c(t), j|\boldsymbol{\theta}_{r,t-1});$$
$$g_{u,j} \leftarrow g_u(c(t), j|\boldsymbol{\theta}_{u,t-1}).$$

Primal Update: Optimistically estimate $r(c(t), j)$, pessimistically estimate $u(c(t), j), \forall j \in \mathcal{N}$:

$$\widehat{R}(c(t), j) \leftarrow f_r(c(t), j|\boldsymbol{\theta}_{r,t-1}) + \frac{\gamma_{t-1}}{\sqrt{m}} \|g_{r,j}\|_{\boldsymbol{\Sigma}_{r,t-1}^{-1}};$$
$$\widehat{U}(c(t), j) \leftarrow f_u(c(t), j|\boldsymbol{\theta}_{u,t-1}) - \frac{\gamma_{t-1}}{\sqrt{m}} \|g_{u,j}\|_{\boldsymbol{\Sigma}_{u,t-1}^{-1}};$$

Clip $\widehat{R}(c(t), j)$ and $\widehat{U}(c(t), j)$ to interval $[0, 1]$.

MaxWeight: Select $S(t)$ (breaking a tie randomly):

$$S(t) \in \arg\max_{S \in F} \sum\nolimits_{j \in S} V_t \widehat{R}(c(t), j) - Q(t)\widehat{U}(c(t), j),$$

where recall that $F = \{S \subseteq \mathcal{N} : |S| = M\}$.

Feedbacks: Get rewards and costs of selected arms:

$$\{R(c(t), j), U(c(t), j)\}_{j \in S(t)}.$$

Dual Update: Update the queue length as follows:

$$Q(t+1) \leftarrow \left[Q(t) + \sum\nolimits_{j \in S(t)} \widehat{U}(c(t), j) - B(t) + \epsilon_t\right]^+.$$

Statistics Update:

$$\boldsymbol{\Sigma}_{r,t} \leftarrow \boldsymbol{\Sigma}_{r,t-1} + \frac{1}{m} \sum\nolimits_{j \in S(t)} g_{r,j} g_{r,j}^\top;$$
$$\boldsymbol{\Sigma}_{u,t} \leftarrow \boldsymbol{\Sigma}_{u,t-1} + \frac{1}{m} \sum\nolimits_{j \in S(t)} g_{u,j} g_{u,j}^\top;$$

Obtain $\boldsymbol{\theta}_{r,t}, \boldsymbol{\theta}_{u,t}$ by training MLPs f_r, f_u using collected feedbacks as in Definition 4.

end for

instead of a constant ϵ is adopted to adaptively complement the online estimation $\widehat{U}(c(t), j)$ across round. Specifically, ϵ_t is the time-varying tightness parameter of the anytime cumulative constraints. As shown in Section V, ϵ_t is relatively large at the early rounds to guarantee the constraints when the cost estimation is less accurate; In later rounds, the tightness becomes smaller since it is unlikely to violate the constraints when the cost estimation becomes accurate. These two differences result in the complex coupling between the cost estimation and virtual queue update, and impose additional challenges in our primal-dual-based design.

Remark 3 (Effect of ϵ_t). As shown in the update rule (4), ϵ_t serves as an extra penalty to the update. Intuitively, a larger value of ϵ_t results in a larger increase in the virtual queue length, thus forcing the agent to select arms with smaller costs to minimize constraint violation. Alternatively, recall that ϵ_t is the tightness parameter in the tightened problem in Definition 1, a larger ϵ_t causes a tighter problem, equivalently implying more focus of the agent on the violation. Please refer to Section VI for the corresponding numerical justification.

MaxWeight Selection. Finally, after the substitution, we have the concise rule based on the estimation of primal and dual variables to select super arm $S(t)$ in each round t:

$$S(t) \in \arg\max_{S \in F} \sum\nolimits_{j \in S} \underbrace{V_t \widehat{R}(c(t), j) - Q(t)\widehat{U}(c(t), j)}_{\text{weight of (context, arm index) pair } (c(t), j)}, \quad (5)$$

where recall that $F = \{S \subseteq \mathcal{N} : |S| = M\}$ and V_t is a weight parameter to balance the tradeoff between maximizing reward and minimizing constraint violation.

Remark 4 (Effect of V_t). Intuitively, a larger value of V_t implies a larger weight on reward estimation, thus letting the agent focus more on the reward maximization rather than minimizing the constraint violation. Please refer to Section VI for the corresponding numerical justification.

V. Theoretical Analysis

To evaluate the performance of Neural-PD, we analyze its regret and constraint violation. Following their formal definitions, we provide not only main theorems on the general anytime cumulative constraints but also corollaries on specific constraints like fairness and budget constraints. All detailed proofs are delegated to our online technical report [51].

Definition 5 ((Pseudo-)Regret)**.** For any round $\tau \in [T]$,

$$\mathcal{R}(\tau) \triangleq \mathcal{R}^* - \mathbb{E}\left[\sum\nolimits_{t=1}^{\tau} \sum\nolimits_{j \in \mathcal{N}} R(c(t), j) X_j(t)\right],$$

where $\mathcal{R}^*$ is the optimal value to baseline (Definition 6).

Definition 6 (Baseline Problem)**.** We consider baseline problem as the tightened problem (Definition 1) with $\epsilon = 0$.

We make Assumption 1 for the following theoretical results.

Assumption 1 (Slater's Condition)**.** Without loss of generality, we assume there exist Slater's constant $\delta \in (0, 1]$ and a feasible solution $\{\mathbf{q}_c\}_{c \in \mathcal{C}}$ to the baseline problem (Definition 6) with

$$\sum\nolimits_{c \in \mathcal{C}} p_c \sum\nolimits_{S \in F} q(c, S) \sum\nolimits_{j \in S} u(c, j) + \delta \leq b.$$

Note that Slater's condition holds in an average sense across all possible contexts and arm selections. Particularly, it is a mild assumption since it only requires the existence of a probability distribution over super arms under which the

expected cost is strictly less than the expected budget [18], [19]. Slater's constant can be from a feasible solution that is not necessarily optimal. This implies it is a more relaxed quantity than the safety gap in hard constrained cases, which is defined under the optimal solution [38].

Definition 7 (Constraint Violation)**.** For any round $\tau \in [T]$,

$$\mathcal{V}(\tau) \triangleq \left[\mathbb{E}\left(\sum_{t=1}^{\tau}\left(\sum_{j\in\mathcal{N}} U(c(t),j)X_j(t) - B(t)\right)\right)\right]^+.$$

We choose the best possible (*i.e.*, zero) violation as baseline.

By definition, the violation in a round could be compensated in another one. Therefore, the constraints in our setting are often categorized as the *soft* constraints. In other words, the algorithm may violate the constraint in the early rounds when the estimation of the cost function is inaccurate but manage to achieve zero violation later on. It is worth mentioning that the flexibility of primal-dual optimization makes it possible to extend our model with the *hard constraint* version (please refer to [52]). Specifically, the constraint violation is then defined as $\mathcal{V}(\tau) = \mathbb{E}\big[\sum_{t=1}^{\tau}\big(\sum_{j\in\mathcal{N}} U(c(t),j)X_j(t) - B(t)\big)^+\big]$.

A. Main Theorems

Theorem 1 (Regret Bound)**.** *Given Slater's constant $\delta \in (0,1]$, set tunable parameters*

$$\epsilon_t = \frac{2C_1\tilde{d}\sqrt{M}\log(1+T)}{\sqrt{t}}, V_t = \frac{C_1\tilde{d}\delta\sqrt{t}\log(1+T)}{2\sqrt{M}}$$

with some positive constant C_1 (specified in our proof). Suppose MLPs f_r, f_u are initialized as in Definition 3; the neural network width m, regularization parameter λ, and step size η satisfy that

$$m \geq \mathrm{poly}\left(T, L, N, \lambda^{-1}, \bar{\lambda}^{-1}, \log(T)\right),$$
$$\lambda \geq \max\left\{1, (2\mathbf{r}^\top\mathbf{H}^{-1}\mathbf{r})^{-1}\right\}, \eta = O\Big((mTL+m\lambda)^{-1}\Big),$$

Neural-PD achieves the regret bound

$$\mathcal{R}(\tau) \leq \frac{4(1+6C_1+12C_1^2)M^3\tilde{d}^2\log^2(1+T)}{\delta^3} + \frac{2\sqrt{M}\Big(4C_1^{-1}M^2 + 2C_1M\tilde{d} + 2C_1^{-1}b_{\max}^2\Big)\log(1+T)\sqrt{\tau}}{\delta} + C_2\sqrt{M}\tilde{d}\log(1+T)\sqrt{\tau} + 2M$$

for some positive constants C_1, C_2, the mean reward vector $\mathbf{r} \triangleq [r(c(t),j)]_{t,j}$, and $\bar{\lambda}$ is the smallest eigenvalue of $\mathbf{H}$. $\tilde{d}$ is the effective dimension of the NTK matrix $\mathbf{H}$ [11], i.e.,

$$\tilde{d} = \frac{\log\det(\mathbf{I}+\mathbf{H}/\lambda)}{\log(1+TN/\lambda)}.$$

The effective dimension $\tilde{d}$ measures how quickly the eigenvalues of NTK matrix $\mathbf{H}$ diminish. Suppose there exists a kernel function that maps contexts into a $\widehat{d}$-dimensional RKHS associated with the NTK matrix $\mathbf{H}$, it is known that the effective dimension $\tilde{d} < \widehat{d}$ [11]. In the special case where the eigenvalues are only polynomially diminishing, the effective dimension $\tilde{d}$ depends on T but in a logarithmical sense [10].

Proof sketch. We first decompose the regret into terms of three types, *i.e.*, 1) ϵ_t-tight, 2) Lyapunov drift, and 3) reward mismatch. For the first type, we resort to a carefully constructed feasible solution to the tightened problem (Definition 1) based on Slater's condition and the optimal solution to the baseline problem (Definition 6). For the second type, we adopt Lyapunov-drift techniques [34] (with Lyapunov function $L(t) \triangleq [Q(t)]^2/2$) and doubling tricks [53] with extra handling on mismatch term induced by the unknown cost function. For third type, we utilize a self-normalized bound for the reward estimation based on the NTK theory.

Remark 5 (Condition Interpretation). One may find the complex conditions overwhelming in Theorem 1 and later Theorem 2. The main takeaways for practical implementation are: 1) Balanced tradeoff with tuned parameters ϵ_t, V_t. 2) Over-parameterization with sufficiently large width m. 3) Proper initialization for MLPs f_r, f_u. 4) Sufficient regularization "strength" λ. 5) Small step size η for neural network update.

Remark 6 (Sharp Regret Bound). The regret bound in Theorem 1 is at order $\widetilde{O}\left(\frac{M^3\tilde{d}^2}{\delta^3} + \frac{M^{5/2}\tilde{d}\sqrt{\tau}}{\delta}\right)$, where $\tilde{d}$ is at most of order $\log(T)$ as mentioned in Theorem 1. There are two important observations. 1) The regret bound is independent of the number of arms, the number of contexts, and the dimension of the cost function, which all may be huge (even infinite) in practice [54]. The dominating term (*i.e.*, the second term $M^{5/2}\tilde{d}\sqrt{\tau}/\delta$) only grows sublinearly in τ; and linearly in $\tilde{d}$, the effective dimension of the NTK matrix $\mathbf{H}$; and the inverse of Slater's constant δ. 2) Our regret bound is sharp. In terms of horizon T, the regret bound $\mathcal{R}(T)$ matches the problem-independent lower bound $\Omega(\sqrt{T})$ in unconstrained multi-armed bandits [1] up to logarithmic factors.

Theorem 2 (Zero Constraint Violation)**.** *With the Slater's constant $\delta \in (0,1]$, initialization of f_r, f_u, ϵ_t, V_t, neural network width m, and step size η the same as in Theorem 1, we set the regularization parameter λ to guarantee*

$$\lambda \geq \max\left\{1, (2\mathbf{u}^\top\mathbf{H}^{-1}\mathbf{u})^{-1}\right\},$$

where the mean cost vector $\mathbf{u} = [u(c(t),j)]_{t,j}$. Then under Neural-PD, the constraint violation

$$\mathcal{V}(\tau) \leq \Big[\frac{32C_1^2M^2\tilde{d}^2\log^2(1+T)}{\delta^2} + \frac{48M^2}{\delta}\log\left(\frac{16M}{\delta}\right) + \big(14M^2 + 4b_{\max}^2\big)/\delta + C_1\sqrt{M}\tilde{d}\log(1+T)\big(4-\sqrt{\tau}\big)\Big]^+$$

for the same constant C_1 in Theorem 1. Therefore, we have

$$\mathcal{V}(\tau) = \begin{cases} O\left(\frac{M^2\tilde{d}^2\log^2(1+T)}{\delta^2}\right) & \tau \leq O\left(\frac{M^3\tilde{d}^2\log^2 T}{\delta^4}\right), \\ 0 & \text{otherwise}. \end{cases}$$

That is, Neural-PD requires $O\left(\frac{M^3\tilde{d}^2}{\delta^4}\log^2 T\right)$ rounds to reach zero constraint violation because it takes a necessary online learning procedure to estimate the cost function. That is, if

the cost function is known or trivial (e.g., constant), we can eliminate the dependence on T (please refer to [19]).

Proof sketch. We first prove an bound of constraint violation based on equation (4), which mainly includes two terms: 1) expected queue length and 2) cost mismatch. We leverage a variant of the basic drift lemma [55] to bound the exponential moment of Lyapunov drift $\Delta(t) \triangleq L(t+1) - L(t)$, and further bound queue length. The cost mismatch term is similarly bounded as the reward mismatch.

Remark 7 (Dependence on Slater's Constant). In Theorem 1 and 2, the regret bound and constraint violation increase in the inverse of Slater's constant δ. Intuitively, δ determines the size of the feasible set for solutions to the baseline problem (6) by Definition 1. A smaller δ implies a smaller feasible set, thus a harder procedure to a feasible solution. Therefore, both regret and violation increase as δ decreases since the problem becomes harder and requires more accurate learning.

Remark 8 (Dependence on Effective Dimension). In Theorem 1 and 2, the regret bound and constraint violation both increase in the effective dimension $\tilde{d}$. Intuitively, $\tilde{d}$ describes the underlying dimension in the set of observed contexts over the entire horizon [12]. Specifically, it represents the number of principle directions over which the projection of the context in the RKHS (spanned by NTK matrix $\mathbf{H}$) is spread [10]. Consequently, a larger $\tilde{d}$ means that the contexts are more widely spread in the space. This implies a harder problem because the MLPs need more training rounds to accurately learn reward and cost functions with inputs of larger variation.

B. Corollaries on Specific Operational Constraints

Fairness Constraints. For the anytime fairness constraints, we replace the cost function $U(\cdot,\cdot)$ with constant value, -1, and replace budget $B(t)$ with a negative threshold $-b_j \in [-b_{\max}, 0]$ for each arm to get the form in (6), and main results follow those in Section V with an extra scaling factor of order N since fairness constraints are imposed on each arm.

$$\mathbb{E}\left[\sum_{t=1}^{\tau} X_j(t)\right] \geq b_j \tau, \forall j \in \mathcal{N}, \forall \tau \in [T]. \tag{6}$$

For the horizon-dependent fairness, we have:

Corollary 1 (Horizon-dependent Fairness Constraints)**.** *Define horizon-dependent fairness constraints as*

$$\mathbb{E}\left[\sum_{t=1}^{T} X_j(t)\right] \geq b_j T, \forall j \in \mathcal{N}, b_j \in (0,1).$$

Given $\delta \geq 4\epsilon$ with $\epsilon = \frac{2}{\sqrt{T}}$ and $V = \frac{\delta\sqrt{T}}{2\sqrt{N}M}$, we have

$$\mathcal{R}_{\textit{fairness}}(T) \leq \widetilde{O}\left(\left(MN^{3/2}/\delta + \sqrt{M}\tilde{d}\right)\sqrt{T}\right),$$

$$\mathcal{V}_{\textit{fairness}}(T) \leq N^{3/2}\left(\frac{3}{\rho}\nu^2 \log\frac{2\nu}{\rho} + \nu + \frac{4B}{\delta}\right) + 2MN = O(1),$$

where $B = N(2+\epsilon^2), \rho = \frac{\delta}{2} - \epsilon$ and $\nu = \max\{\rho, 5N\}$.

Budget Constraints. For the anytime budget constraints, the results directly follow those in Section V with proper interpretation of the cost as quantities like energy and money. For the horizon-dependent budget constraints, we have:

Corollary 2 (Horizon-dependent Budget Constraints)**.** *Define horizon-dependent budget constraints for K types of cost as*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j\in\mathcal{N}} U^{(k)}(c(t),j)X_j(t)\right] \leq Tb^{(k)}, \forall k \in [K],$$

where we assume that $u^{(k)}(c(t),j) = \mathbb{E}[U^{(k)}(c(t),j)] \in [-1,1], \forall c(t) \in \mathcal{C}, j \in \mathcal{N}$; $b^{(k)} \in [0, b_{\max}]$ is the average budget for kth type cost. Therefore, by assuming $\delta \geq 4\epsilon$ with $\epsilon = \frac{C_1\sqrt{M}\tilde{d}\log(1+T)}{2\sqrt{T}}$ and $V = \frac{3C_1\delta\tilde{d}\sqrt{T}\log(1+T)}{2\sqrt{KM}}$, we have

$$\mathcal{R}_{\textit{budget}}(T) \leq \widetilde{O}\left(\left(M^{5/2}K^{3/2} + M^{3/2}\tilde{d}\right)\sqrt{T}/\delta\right),$$

$$\mathcal{V}_{\textit{budget}}(T) \leq K^{3/2}\left(\frac{3}{\rho}\nu^2\log\frac{2\nu}{\rho} + \nu + \frac{4B}{\delta}\right) + 2MK = O(1),$$

where $B = K(3M^2 + 2M\epsilon + \epsilon^2 + b_{\max}^2), \rho = \frac{\delta}{2} - \epsilon$ and $\nu = \max\{\rho, K(M+1)\}$.

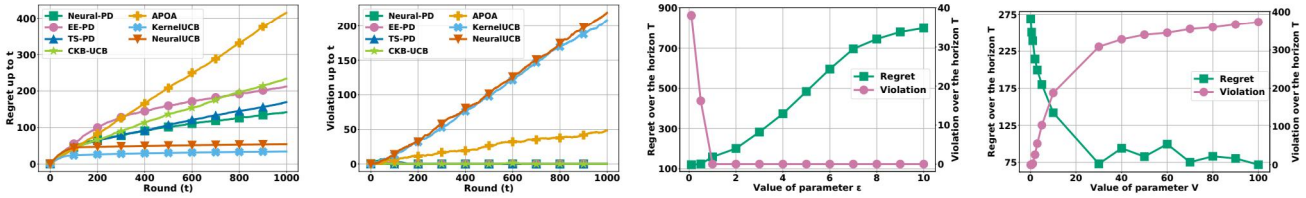
VI. Numerical Results

In this section, we consider four baselines: 1) *KernelUCB* [10], an unconstrained kernelized UCB algorithm that uses a (*e.g.*, Gaussian) kernel function. 2) *NeuralUCB* [11], an unconstrained neural bandits algorithm with UCB-based exploration. 3) *APOA* [19], an algorithm for linear bandits with linear anytime cumulative constraints. 4) *CKB-UCB* [18], a constrained version of KernelUCB (with RBF kernel [56]) with primal-dual optimization. We also consider two variants of Neural-PD: 1) *EE-PD*, whose primal update is replaced with that of EE-Net [14]. It follows the same UCB-style estimation but introduces extra neural networks to determine exploitation and exploration. 2) *TS-PD*, a Thompson-Sampling-based neural constrained combinatorial bandits algorithm whose primal update is adopted from NeuralTS [12].

In the following, we choose a two-layer MLP with network width $m = 128$ for neural-network-based bandit algorithms. We use the Adam optimizer [49] for neural network training and set regularization parameter $\lambda = 6.25 \times 10^{-4}$, step size $\eta = 3 \times 10^{-4}$, and number of gradient-based updates $J = 8$. Our analysis in Section V guarantees the theoretical superiority of Neural-PD only with overparameterized MLPs. In the practical implementation, the neural networks under the framework of Neural-PD can extend to broader options of advanced architectures like convolutional neural networks (CNNs) [57], graph neural networks (GNNs) [58] and Transformers [59].

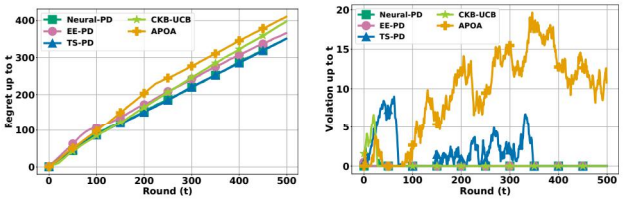
Synthetic Dataset. We consider a contextual bandit model[7] of $N = 10$ arms and context $c(t) \in \mathbb{R}^{10\times 8}$ that follows a standard multivariate Gaussian distribution with proper normalization. We set the horizon $T = 1000$ and the size of super arms $M = 3$. The mean reward $r(c(t),j) = |\langle \theta^r, c_j^\top(t)\rangle|$ where $c_j(t)$ is the jth row of $c(t)$ and column vector

[7]Neural-PD can handle the large-scale setting with a large number of arms N and a large arm selection number M. Our synthetic simulation is meant for illustration to justify our theoretical analysis and Neural-PD's outperformance.



(a) Regret. (b) Constraint Violation. (c) Effect of ϵ_t on Neural-PD. (d) Effect of V_t on Neural-PD.

Fig. 2. Simulation Results on the Synthetic Dataset.



(a) Regret. (b) Constraint Violation.

Fig. 3. Simulation Results on the Real-world Dataset of Crowdsourcing.

$\theta^r = [0.46, 0.1, 0.25, 0.58, 0.48, -0.25, 0.24, -0.03]^\top$. Similarly, the mean cost $u(c(t), j) = |\langle \theta^u, c_j^\top(t) \rangle|$ where $\theta^u = [-0.06, 0.16, 0.04, 0.57, 0.29, 0.06, 0.17, 0.13]^\top$. The budget $B(t) = 0.6$. The noises $\eta(t), \xi(t)$ follow standard Gaussian distributions. The parameters $\epsilon_t = 1/\sqrt{t}$ and $V_t = \sqrt{t}$.

In Fig. 2(a) and 2(b), we present the performances of all algorithms in terms of regret and constraint violation, respectively. Compared with others, unconstrained baselines (KernelUCB and NeuralUCB) incur lower regret while higher constraint violations since they solely maximize rewards without considering the cost. Compared with constrained baselines (APOA and CKB-UCB), Neural-PD and its variants (EE-PD and TS-PD) achieve lower regrets with zero constraint violations. Specifically, APOA achieves both higher regrets and constraint violation given linear functions for estimation. CKB-UCB achieves higher regrets since it models both reward and cost functions using kernel-based methods (less fine-grained modeling compared with overparameterized MLPs), thus a worse tradeoff between regret and constraint violation compared with Neural-PD and its variants.

In Fig. 2(c) and 2(d), we present the performances of Neural-PD under varying values of ϵ_t and V_t. Specifically, when $\epsilon_t = \frac{\epsilon}{\sqrt{t}}$ with ϵ varying from 0.1 to 10, the regret increases as the a larger ϵ_t implies a harder *constrained* problem, Neural-PD thus focusing more on the violation. When $V_t = V\sqrt{t}$ with V varying from 0.1 to 100, Neural-PD focuses more on the regret as a larger V_t in (5) makes the weight of reward maximization greater, thus less regret. The above results justify the discussion in Remark 3.

Task Assignment in Crowdsourcing. We consider a spatial crowdsourcing system [16] with $N = 50$ workers (arms). In each round, a task is published in a specific location and requires $M = 5$ workers. The context $c(t) \in \mathbb{R}^{50\times 2}$, and for jth row $c_j(t)$, its first element $c_{j,1}(t)$ represents the (normalized) distance between worker j and the target task in round t; and the second element $c_{j,2}(t)$ indicates the rating or quality of worker j in round t. The location-related context $c_{j,1}(t)$ is sampled from the Gowalla dataset [60], and the rating-related context $c_{j,2}(t)$ is sampled from a uniform distribution over $[0, 1]$. The mean reward $r(c(t), j) = \varphi(c_{j,1}(t)) \cdot (c_{j,2}(t))^2$ where $\varphi(\cdot)$ is the PDF of the standard Gaussian distribution [16]. The noise $\eta(t)$ follows the standard Gaussian distribution. The mean cost is set as $u(c(t), j) = |\langle \theta^{u'}, c_j^\top(t) \rangle|$ where $\theta^{u'} = [-0.2, 0.5]^\top$. The noise $\xi(t)$ follows the uniform distribution over $[0, 1]$. The budget $B(t) = 1.2$. The horizon $T = 500$. The parameters $\epsilon_t = 1/\sqrt{t}$ and $V_t = \sqrt{t}/3$.

In Fig. 3, we evaluate the performances of Neural-PD, its variants (EE-PD and TS-PD), and constrained baselines (APOA and CKB-UCB) for task assignment in crowdsourcing on the real-world dataset. Compared with APOA, Neural-PD and its variants achieve both lower regrets and zero constraint violations. Similar to the synthetic case, Neural-PD and its variants outperform CKB-UCB in terms of regret due to flexible representation ability in the face of complex functions. The results justify the effectiveness of our algorithm in terms of adapting to real-world scenarios.

VII. Conclusion & Future Work

We studied the neural constrained combinatorial bandits under general anytime cumulative constraints. We proposed an efficient neural-network-based primal-dual algorithm (Neural-PD) to minimize both regret and constraint violation. With neural tangent kernel theory and Lyapunov-drift techniques, we showed Neural-PD achieves a sharp regret bound and a zero constraint violation. We also showed its effectiveness compared with baselines on synthetic and real-world datasets.

In the future, we would like to consider extensions of our work in two aspects: 1) Bandits variants, *e.g.*, graphical bandits [61] and dueling bandits [62]; 2) Advanced neural network architectures, *e.g.*, CNNs, GNNs, and Transformers.

References

[1] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.
[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
[3] D. C. Brabham, *Crowdsourcing*. MIT Press, 2013.
[4] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proceedings of ICML*, 2013.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of WWW*, 2010.

[6] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[7] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proceedings of NeurIPS*, 2011.

[8] H. Cao, Q. Pan, Y. Zhu, and J. Liu, "Birds of a feather help: Context-aware client selection for federated learning," in *Proceedings of FL-AAAI*, 2022.

[9] S. Agrawal, S. Yin, and A. Zeevi, "Dynamic pricing and learning under the bass model," in *Proceedings of ACM EC*, 2021.

[10] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini, "Finite-time analysis of kernelised contextual bandits," in *Proceedings of UAI*, 2013.

[11] D. Zhou, L. Li, and Q. Gu, "Neural contextual bandits with ucb-based exploration," in *Proceedings of ICML*, 2020.

[12] W. Zhang, D. Zhou, L. Li, and Q. Gu, "Neural thompson sampling," in *Proceedings of ICLR*, 2021.

[13] T. Nguyen-Tang, S. Gupta, A. T. Nguyen, and S. Venkatesh, "Offline neural contextual bandits: Pessimism, optimization and generalization," in *Proceedings of ICLR*, 2021.

[14] Y. Ban, Y. Yan, A. Banerjee, and J. He, "Ee-net: Exploitation-exploration neural networks in contextual bandits," in *Proceedings of ICLR*, 2022.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[16] S. Lin, Y. Yao, P. Zhang, H. Y. Noh, and C. Joe-Wong, "A neural-based bandit approach to mobile crowdsourcing," in *Proceedings of HotMobile*, 2022.

[17] S. Salgia, S. Vakili, and Q. Zhao, "Provably and practically efficient neural contextual bandits," *arXiv preprint arXiv:2206.00099*, 2022.

[18] X. Zhou and B. Ji, "On kernelized multi-armed bandits with constraints," *arXiv preprint arXiv:2203.15589*, 2022.

[19] X. Liu, B. Li, P. Shi, and L. Ying, "An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraints," in *Proceedings of NeurIPS*, 2021.

[20] A. Badanidiyuru, J. Langford, and A. Slivkins, "Resourceful contextual bandits," in *Proceedings of COLT*, 2014.

[21] S. Agrawal and N. Devanur, "Linear contextual bandits with knapsacks," in *Proceedings of NeurIPS*, 2016.

[22] J. Steiger, B. Li, and N. Lu, "Learning from delayed semi-bandit feedback under strong fairness guarantees," in *Proceedings of IEEE INFOCOM*, 2022.

[23] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "Learning with guarantee via constrained multi-armed bandit: Theory and network applications," *IEEE Transactions on Mobile Computing*, 2022 Early Access.

[24] X. Liu, B. Li, P. Shi, and L. Ying, "Pond: Pessimistic-optimistic online dispatching," *arXiv preprint arXiv:2010.09995*, 2020.

[25] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *Proceedings of IEEE INFOCOM*, 2019.

[26] X. Gao, X. Huang, Y. Tang, Z. Shao, and Y. Yang, "History-aware online cache placement in fog-assisted iot systems: An integration of learning and control," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 683–14 704, 2021.

[27] V. Patil, G. Ghalme, V. Nair, and Y. Narahari, "Achieving fairness in the stochastic multi-armed bandit problem." in *Proceedings of AAAI*, 2021.

[28] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1, pp. 245–257, 2015.

[29] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks," *Journal of the ACM (JACM)*, vol. 65, no. 3, pp. 1–55, 2018.

[30] S. Cayci, A. Eryilmaz, and R. Srikant, "Budget-constrained bandits over general cost and reward distributions," in *Proceedings of AISTATS*, 2020.

[31] Z. Ou, J. Dong, S. Dong, J. Wu, A. Ylä-Jääski, P. Hui, R. Wang, and A. W. Min, "Utilize signal traces from others? a crowdsourcing perspective of energy saving in cellular data communication," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 194–207, 2014.

[32] I.-H. Hou, "On the modeling and optimization of short-term performance for real-time wireless networks," in *Proceedings of IEEE INFOCOM*, 2016.

[33] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proceedings of NeurIPS*, 2018.

[34] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[35] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. 95, pp. 2651–2667, 2006.

[36] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," in *Proceedings of NeurIPS*, 2019.

[37] Y. Cao and Q. Gu, "Generalization bounds of stochastic gradient descent for wide and deep neural networks," in *Proceedings of NeurIPS*, 2019.

[38] S. Amani, M. Alizadeh, and C. Thrampoulidis, "Linear stochastic bandits under safety constraints," in *Proceedings of NeurIPS*, 2019.

[39] A. Moradipari, S. Amani, M. Alizadeh, and C. Thrampoulidis, "Safe linear thompson sampling," *arXiv preprint arXiv:1911.02156*, 2019.

[40] Y. Chen, A. Cuellar, H. Luo, J. Modi, H. Nemlekar, and S. Nikolaidis, "Fair contextual multi-armed bandits: Theory and experiments," in *Proceedings of UAI*, 2020.

[41] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "When do neural networks outperform kernel methods?" in *Proceedings of NeurIPS*, 2020.

[42] D. Bouneffouf, "Online learning with corrupted context: Corrupted contextual bandits," *arXiv preprint arXiv:2006.15194*, 2020.

[43] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," *arXiv preprint arXiv:1710.10686*, 2017.

[44] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[45] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2503–2528, 2012.

[46] R. Jenatton, J. Huang, and C. Archambeau, "Adaptive algorithms for online convex optimization with long-term constraints," in *Proceedings of ICML*, 2016.

[47] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, 2018.

[48] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of ICML*, 2010.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Proceedings of NeurIPS*, 2020.

[51] S. Wang, S. Bian, X. Liu, and Z. Shao, "Neural constrained combinatorial bandits," ShanghaiTech University, Tech. Rep., 2022. [Online]. Available: http://faculty.sist.shanghaitech.edu.cn/faculty/shaozy/Neural-PD.pdf

[52] H. Guo, X. Liu, H. Wei, and L. Ying, "Online convex optimization with hard constraints: Towards the best of two worlds and beyond," in *Proceedings of NeurIPS*, 2022.

[53] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Proceedings of IEEE FOCS*, 1995.

[54] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, "Parametric bandits: The generalized linear case," in *Proceedings of NeurIPS*, 2010.

[55] M. J. Neely, "Energy-aware wireless scheduling with near-optimal backlog and convergence time tradeoffs," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2223–2236, 2015.

[56] J. Shawe-Taylor, N. Cristianini *et al.*, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[57] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[58] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NeurIPS*, 2017.

[60] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of ACM SIGKDD*, 2011.

[61] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *Proceedings of NeurIPS*, 2011.

[62] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1538–1556, 2012.