Multi-facet Contextual Bandits: A Neural Network Perspective

Yikun Ban
University of Illinois at Urbana-Champaign
yikunb2@illinois.edu

Jingrui He
University of Illinois at Urbana-Champaign
jingrui@illinois.edu

Curtiss B. Cook
Mayo Clinic Arizona
cook.curtiss@mayo.edu

ABSTRACT

Contextual multi-armed bandit has shown to be an effective tool in recommender systems. In this paper, we study a novel problem of multi-facet bandits involving a group of bandits, each characterizing the users' needs from one unique aspect. In each round, for the given user, we need to select one arm from each bandit, such that the combination of all arms maximizes the final reward. This problem can find immediate applications in E-commerce, healthcare, etc. To address this problem, we propose a novel algorithm, named MuFasa, which utilizes an assembled neural network to jointly learn the underlying reward functions of multiple bandits. It estimates an Upper Confidence Bound (UCB) linked with the expected reward to balance between exploitation and exploration. Under mild assumptions, we provide the regret analysis of MuFasa. It can achieve the near-optimal $\widetilde{O}((K+1)\sqrt{T})$ regret bound where K is the number of bandits and T is the number of played rounds. Furthermore, we conduct extensive experiments to show that MuFasa outperforms strong baselines on real-world data sets.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Display advertising**; • **Theory of computation** → **Online learning algorithms**.

KEYWORDS

Contextual Bandits; Neural Network; Regret Analysis

ACM Reference Format:
Yikun Ban, Jingrui He, and Curtiss B. Cook. 2021. Multi-facet Contextual Bandits: A Neural Network Perspective. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467299

1 INTRODUCTION

The personalized recommendation is ubiquitous in web applications. Conventional approaches that rely on sufficient historical records, e.g., collaborative filtering [37, 47], have proven successful both theoretically and empirically. However, with the cold-start problem and the rapid change of the recommendation content, these methods might render sub-optimal performance [23, 28]. To solve the dilemma between the exploitation of historical data and the exploration of new information, Multi-Armed Bandit (MAB) [1, 7, 8, 10, 26] turns out to be an effective tool, which has been adapted to personalized recommendation [23, 28], online advertising [40], clinical trials [12, 20], etc.

In the conventional contextual bandit problem setting [28], i.e., single MAB, the learner is presented with a set of arms in each round, where each arm is represented by a feature vector. Then the learner needs to select and play one arm to receive the corresponding reward that is drawn from an unknown distribution with an unknown mean. To achieve the goal of maximizing the accumulated rewards, the learner needs to consider the arms with the best historical feedback as well as the new arms for potential gains. The single MAB problem has been well studied in various settings. With respect to the reward function, one research direction [1, 19, 23, 28, 29] assumes that the expected reward is linear with respect to the arm's feature vector. However, in many real applications, this assumption fails to hold. Thus many exiting works turn to focus on the nonlinear or nonparametric bandits [13, 36] with mild assumptions such as the Lipschitz continuous property [13] or embedding in Reproducing Kernel Hilbert Space [18, 39]. Furthermore, the single MAB problem has been extended to best arm identification [6, 7], outlier arm identification [10, 22], Top-K arm problems [14], and so on.

In this paper, we define and study a novel problem of *multi-facet contextual bandits*. In this problem, the users' needs are characterized from multiple aspects, each associated with one bandit. Consider a task consisting of K bandits, where each bandit presents a set of arms separately and the learner needs to choose and play one arm from each bandit. Therefore, a total of K arms are played in one round. In accordance with the standard bandit problem, the learner can observe a reward after playing one arm from the bandit, which we call "sub-reward", and thus K sub-rewards are received in total. In addition, a reward that is a function with respect to these K sub-rewards, called "final reward", is observed to represent the overall feedback with respect to the K selected arms. Note that the functions of final reward and K sub-rewards are allowed to be either linear or non-linear. The goal of the learner in the multi-facet bandit problem is to maximize the final rewards of all the played rounds.

This problem finds many applications in real-world problems. For instance, in the recommender system, instead of the single item recommendation, an E-commerce company launches a promotion campaign, which sells collections of multiple types of products such as snacks, toiletries, and beverages. Each type of item can be formulated as a multi-armed bandit and the learner aims to select the best combination of snack, toiletry, and beverage. As a result, the final reward is the review of this combined recommendation, while the sub-reward is the review for a particular product . This problem also exists in healthcare. For a diabetes patient, the doctor usually

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '21, August 14–18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467299

provides a comprehensive recommendation including medication, daily diet, and exercise, where each type has several options. Here, the final reward can be set as the change of key biomarkers for diabetes (e.g., HbA1c) and the sub-reward can be the direct impact of each type of recommendation (e.g., blood pressure change for a medicine).

A major challenge of the proposed multi-facet bandit problem is the partial availability of sub-rewards, as not every sub-reward is easy to observe. For example, regarding the combined recommendation of E-commerce, the user may rate the combination but not individual items; regarding the comprehensive recommendation for a diabetes patient, some sub-rewards can be difficult to measure (e.g., the impact of low-calorie diets on the patient's overall health conditions). Therefore, in our work, we allow only a subset of all sub-rewards to be observed in each round, which increases the flexibility of our proposed framework.

To address these challenges, we aim to learn the mappings from the selected K arms (one from each bandit) to the final rewards, incorporating two crucial factors: (1) the collaborative relations exist among these bandits as they formulate the aspects from one same user; (2) the bandits contribute to the task with various weights because some aspects (bandits) are decisive while some maybe not. Hence, we propose a novel algorithm, MuFasa, to learn K bandits jointly. It utilizes an assembled neural networks to learn the final reward function combined with K bandits. Although the neural networks have been adapted to the bandit problem [34, 42, 45], they are designed for the single bandit with one selected arm and one reward in each round. To balance the exploitation and exploration of arm sets, we provide a comprehensive upper confidence bound based on the assembled network linking the predicted reward with the expected reward. When the sub-rewards are partially available, we introduce a new approach to leverage them to train bandits jointly. Furthermore, we carry out the theoretical analysis of MuFasa and prove a near-optimal regret bound under mild assumptions. Our major contributions can be summarized as follows:

(1) **Problem**. We introduce the problem of multi-facet contextual bandits to characterize the users' needs from multiple aspects, which can find immediate applications in E-commerce, healthcare, etc.

(2) **Algorithm**. We propose a novel algorithm, MuFasa, which exploits the final reward and up to K sub-rewards to train the assembled neural networks and explores potential arm sets with a UCB-based strategy.

(3) **Theoretical analysis**. Under mild assumptions, we provide the upper confidence bounds for a neural network and the assembled neural networks. Then, we prove that MuFasa can achieve the $\widetilde{O}((K+1)\sqrt{T})$ regret bound, which is near-optimal compared to a single contextual bandit.

(4) **Empirical performance**. We conduct extensive experiments to show the effectiveness of MuFasa, which outperforms strong baselines on real-world data sets even with partial sub-rewards.

2 RELATED WORK

Multi-armed bandit. The multi-armed bandit was first introduced by [38] and then further studied by many works that succeeded in both theory and practice such as ϵ-greedy [26], Thompson sampling[2], and upper confidence bound [7]. In the contrast with traditional bandits [7, 10], the contextual bandit [1, 28, 40] has the better representation capacity where each arm is represented by a context vector instead of a scalar to infer the reward. Among them, the linear contextual bandits are extensively studied and many of them use the UCB strategy, achieving $\widetilde{O}(\sqrt{T})$ regret bound [1, 11, 23]. To further generalize the reward function, many works use a nonlinear regression model drawn from the reproducing kernel Hilbert space to learn the mapping from contexts to rewards such as the kernel-based methods [18, 39].

Neural bandits. The authors of [4] use a neural work to model an arm and then applied ϵ-greedy strategy to select an arm. In contrast, MuFasa utilizes a UCB-based strategy working on K bandits instead of one set of arms. In addition, the Thompson sampling has been combined with deep neural networks [9, 31, 34, 42]. For instance, [34, 42] regard the last layer of the neural network as the embeddings of contexts and then apply the Thompson sampling to play an arm in each round. NeuUCB [45] first uses the UCB-based approach constructed on a fully-connected neural network, while it only fits on the single bandit with one set of arms. On the contrary, MuFasa constructs an assembled neural networks to learn K bandits jointly. Deep neural network in multi-view learning has been well-studied [21, 25, 43, 44, 46], to extract useful information among multiple sources, which inspires one of the core ideas of MuFasa.

Other variant bandit setting. In the non-contextual bandit, a number of works [16, 17, 32] study playing K arms at the same time in a single bandit, while these approaches have limited representation power in the recommender system. The most similar setting is the contextual combinatorial MAB problem[30, 33], where the learner tends to choose the optimal subset of arms with certain constraints like the K-size. One key difference is that all the arms are from the same single bandit where only one reward function exists. On the contrary, in the multi-faced bandits, the selected K arms come from K different bandits with K different reward functions and the sub-rewards are allowed to be partially available. There is another line of works [11, 23, 29] for bandit clustering, where a bandit is constructed for each user. They try to leverage the dependency among users to improve the recommendation performance. However, in these works, they still play one arm in each round and the reward function is required to be linear.

3 PROBLEM DEFINITION

In this section, we formulate the problem of multi-facet bandits, with a total of K bandits, where the learner aims to select the optimal set of K arms in each round, in order to maximize the final accumulated rewards.

Suppose there are T rounds altogether. In each round $t \in [T]$ ($[T] = \{1, \ldots, T\}$), the learner is faced with K bandits, and each bandit $k \in [K]$ has a set of arms $\mathbf{X}_t^k = \{\mathbf{x}_{t,1}^k, \ldots, \mathbf{x}_{t,n_k}^k\}$, where $|\mathbf{X}_t^k| = n_k$ is the number of arms in this bandit. In the bandit k, for each arm $\mathbf{x}_{t,i}^k \in \mathbf{X}_t^k$, it is represented by a d_k-dimensional feature vector and we assume $\|\mathbf{x}_{t,i}^k\|_2 \leq 1$. Subsequently, in each round t, the learner will observe K arm sets $\{\mathbf{X}_t^k\}_{k=1}^K$ and thus a total of $\sum_{k=1}^K n_k$ arms. As only one arm can be played within each bandit, the learner needs to select and play K arms denoted as

$\mathbf{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^k, \ldots, \mathbf{x}_t^K\}$ in which $\mathbf{x}_t^k \in \mathbf{X}_t$ represents the selected arm from $\mathbf{X}_t^k$.

Once the selected arm $\mathbf{x}_t^k$ is played for bandit k, a sub-reward r_t^k will be received to represent the feedback of this play for bandit k separately. The sub-reward is assumed to be governed by an unknown reward function:

$$r_t^k(\mathbf{x}_t^k) = h_k(\mathbf{x}_t^k).$$

where h_k can be either a linear [1, 28] or non-linear reward function [18, 39]. As a result, in each round t, the learner needs to play K arms in $\mathbf{X}_t$ and then receive K sub-rewards denoted by $\mathbf{r}_t = \{r_t^1, \ldots, r_t^k, \ldots, r_t^K\}$.

As the K bandits characterize the users' needs from various aspects, after playing K arms in each round t, a final reward R_t will be received to represent the overall feedback of the group of K bandits. The final reward R_t is considered to be governed by an unknown function with respect to $\mathbf{r}_t$:

$$R_t(\mathbf{r}_t) = H\left(\left(h_1(\mathbf{x}_t^1), \ldots, h_k(\mathbf{x}_t^k) \ldots, h_K(\mathbf{x}_t^K)\right)\right) + \epsilon_t.$$

where ϵ_t is a noise drawn from a Gaussian distribution with zero mean. In our analysis, we make the following assumptions regarding h_k and $H(\text{vec}(\mathbf{r}_t))$:

(1) If $\mathbf{x}_t^k = \mathbf{0}$, then $h(\mathbf{x}_t^k) = 0$; If $\text{vec}(\mathbf{r}_t) = (0, \ldots, 0)$, then $H(\text{vec}(\mathbf{r}_t)) = 0$.
(2) $\bar{C}$**-Lipschitz continuity**. $H(\text{vec}(\mathbf{r}_t))$ is assumed to be $\bar{C}$-Lipschitz continuous with respect to the $\mathbf{r}_t$. Formally, there exists a constant $\bar{C} > 0$ such that

$$|H(\text{vec}(\mathbf{r}_t)) - H(\text{vec}(\mathbf{r}_t'))| \leq \bar{C}\sqrt{\sum_{k\in K}[r_t^k - r_t^{k\prime}]^2}.$$

Both assumptions are mild. For (1), if the input is zero, then the reward should also be zero. For (2), the Lipschitz continuity can be applied to many real-world applications. For the convenience of presentation, given any set of selected K arms $\mathbf{X}_t$, we denote the expectation of R_t by:

$$\mathcal{H}(\mathbf{X}_t) = \mathbb{E}[R_t|\mathbf{X}_t] = H\left(\left(h_1(\mathbf{x}_t^1), \ldots, h_k(\mathbf{x}_t^k) \ldots, h_K(\mathbf{x}_t^K)\right)\right). \quad (1)$$

Recall that in multi-facet bandits, the learner aims to select the optimal K arms with the maximal final reward R_t^* in each round. First, we need to identify all possible combinations of K arms, denoted by

$$\mathbf{S}_t = \{(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^k, \ldots, \mathbf{x}_t^K) \mid \mathbf{x}_t^k \in \mathbf{X}_t^k, k \in [K]\}, \quad (2)$$

where $|\mathbf{S}_t| = \prod_{k=1}^K n_k$ because bandit k has n_k arms for each $k \in [K]$. Thus, the regret of multi-facet bandit problem is defined as

$$\begin{aligned}\textbf{Reg} &= \mathbb{E}[\sum_{t=1}^T (R_t^* - R_t)] \\ &= \sum_{t=1}^T \left(\mathcal{H}(\mathbf{X}_t^*) - \mathcal{H}(\mathbf{X}_t)\right),\end{aligned}$$

where $\mathbf{X}_t^* = \arg\max_{\mathbf{X}_t \in \mathbf{S}_t} \mathcal{H}(\mathbf{X}_t)$. Therefore, our goal is to design a bandit algorithm to select K arms every round in order to minimize the regret. We use the standard O to hide constants and $\widetilde{O}$ to hide logarithm.

Availability of sub-rewards. In this framework, the final R_t is required to be known, while the sub-rewards $\mathbf{r}_t$ are allowed to be partially available. Because the feedback of some bandits cannot be directly measured or is simply not available in a real problem. This increases the flexibility of our proposed framework.

More specifically, in each round t, ideally, the learner is able to receive $K + 1$ rewards including K sub-rewards $\{r_t^1, \ldots, r_t^K\}$ and a final reward R_t. As the final reward is the integral feedback of the entire group of bandits and reflects how the bandits affect each other, R_t is required to be known. However, the K sub-rewards are allowed to be partially available, because not every sub-reward is easy to obtain or can be measured accurately.

This is a new challenge in the multi-facet bandit problem. Thus, to learn $\mathcal{H}$, the designed bandit algorithm is required to handle the partial availability of sub-rewards.

4 PROPOSED ALGORITHM

In this section, we introduce the proposed algorithm, MuFasa. The presentation of MuFasa is divided into three parts. First, we present the neural network model used in MuFasa; Second, we detail how to collect training samples to train the model in each round; In the end, we describe the UCB-based arm selection criterion and summarize the workflow of MuFasa.

4.1 Neural network model

To learn the reward function $\mathcal{H}$, we use $K + 1$ fully-connected neural networks to learn K bandits jointly, where a neural network f_k is built for each bandit $k \in [K]$ to learn its reward function h_k, and a shared neural network F is constructed to learn the mapping from the K neural networks $(f_1, \ldots, f_K)$ to the final reward R_t.

First, in round t, for each bandit $k \in [K]$, given any context vector $\mathbf{x}_t^k \in \mathbb{R}^{d_k}$, we use a L_1-layer fully-connected network to learn h_k , denoted by f_k:

$$f_k(\mathbf{x}_t^k; \boldsymbol{\theta}^k) = \sqrt{m_1}\mathbf{W}_{L_1}\sigma(\mathbf{W}_{L_1-1}\sigma(\ldots\sigma(\mathbf{W}_1\mathbf{x}_t^k))),$$

where $\sigma(x)$ is the rectified linear unit (ReLU) activation function. Without loss of generality, we assume each layer has the same width m_1 for the sake of analysis. Therefore, $\boldsymbol{\theta}^k = \left(\text{vec}(\mathbf{W}_{L_1})^\intercal, \ldots, \text{vec}(\mathbf{W}_1)^\intercal\right)^\intercal \in \mathbb{R}^{P_1}$, where $\mathbf{W}_1 \in \mathbb{R}^{m_1\times d_k}$, $\mathbf{W}_i \in \mathbb{R}^{m_1\times m_1}, \forall i \in [1 : L_1 - 1]$, and $\mathbf{W}_{L_1} \in \mathbb{R}^{\widehat{m}\times m_1}$. Note that $f_k(\mathbf{x}_t^k; \boldsymbol{\theta}^k) \in \mathbb{R}^{\widehat{m}}$, where $\widehat{m}$ is set as a tuneable parameter to connect with the following network F. Denote the gradient $\nabla_{\boldsymbol{\theta}^k} f_k(\mathbf{x}_t^k; \boldsymbol{\theta}^k)$ by $g(\mathbf{x}_t^k; \boldsymbol{\theta}^k)$.

Next, to learn the final reward function H, we use a L_2-layer fully-connected network to combine the outputs of the above K neural networks, denoted by F:

$$F\left(\mathbf{f}_t; \boldsymbol{\theta}^\Sigma\right) = \sqrt{m_2}\mathbf{W}_{L_2}\sigma(\ldots\sigma(\mathbf{W}_1(\mathbf{f}_t)))$$

where $\mathbf{f}_t = \left(f_1(\mathbf{x}_t^1; \boldsymbol{\theta}^1)^\intercal, \ldots, f_K(\mathbf{x}_t^K; \boldsymbol{\theta}^K)^\intercal\right)^\intercal \in \mathbb{R}^{\widehat{m}K}$. Also, we assume that each layer has the same width m_2. Therefore, $\boldsymbol{\theta}^\Sigma = (\text{vec}(\mathbf{W}_{L_2})^\intercal, \ldots, \text{vec}(\mathbf{W}_1)^\intercal)^\intercal \in \mathbb{R}^{P_2}$, where $\mathbf{W}_1 \in \mathbb{R}^{m_2\times\widehat{m}K}$, $\mathbf{W}_i \in \mathbb{R}^{m_2\times m_2}, \forall i \in [L_2 - 1]$ and $\mathbf{W}_{L_2} \in \mathbb{R}^{1\times m_2}$, Denote the gradient $\nabla_{\boldsymbol{\theta}^\Sigma} F\left(\mathbf{f}_t; \boldsymbol{\theta}^\Sigma\right)$ by $G(\mathbf{f}_t; \boldsymbol{\theta}^\Sigma)$.

Therefore, for the convenience of presentation, the whole assembled neural networks can be represented by $\mathcal{F}$ to learn $\mathcal{H}$ (Eq.(6)),

Algorithm 1 MuFasa

Input: $\mathcal{F}, T, K, \delta, \eta, J$

Initialize $\boldsymbol{\theta}_0 = (\boldsymbol{\theta}_0^{\Sigma}, \boldsymbol{\theta}_0^1, \ldots, \boldsymbol{\theta}_0^K)$
for each $t \in [T]$ **do**
  for each bandit $k \in [K]$ **do**
    Observe context vectors $\mathbf{X}_t^k = \{\mathbf{x}_{t,1}^k, \ldots, \mathbf{x}_{t,n_k}^k\}$
  Collect $\mathbf{S}_t$ (Eq. (2))
  Choose K arms, $\mathbf{X}_t$, by:

$$\mathbf{X}_t = \arg\max_{\mathbf{X}_t' \in \mathbf{S}_t} \left(\mathcal{F}(\mathbf{X}_t'; \boldsymbol{\theta}_{t-1}) + \mathrm{UCB}(\mathbf{X}_t') \right). \quad (\text{Theorem 5.3})$$

  Play $\mathbf{X}_t$ and observe rewards R_t and $\mathbf{r}_t$.
  if $|\mathbf{r}_t| = K$ **then** ## *sub-rewards are all available.*
    $\boldsymbol{\theta}_t = GradientDescent_{All}(\mathcal{F}, \{\mathbf{X}_i\}_{i=1}^t, \{R_i\}_{i=1}^t, \{\mathbf{r}_i\}_{i=1}^t, J, \eta)$
  else ## *sub-rewards are partially available.*
    Collect $\{\Omega_i\}_{i=1}^t$ (Eq.(4))
    $\boldsymbol{\theta}_t = GradientDescent_{Partial}(\mathcal{F}, \{\Omega_i\}_{i=1}^t, J, \eta)$
  Update $\mathrm{UCB}(\mathbf{X}_t')$.

given the K selected arms $\mathbf{X}_t$:

$$\mathcal{F}(\mathbf{X}_t; \boldsymbol{\theta}) = \left(F(\cdot; \boldsymbol{\theta}^{\Sigma}) \circ \left(f_1(\cdot; \boldsymbol{\theta}^1), \ldots, f_K(\cdot; \boldsymbol{\theta}^K) \right) \right)(\mathbf{X}_t),$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}^{\Sigma}, \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^K)$.

Initialization. $\boldsymbol{\theta}$ is initialized by randomly generating each parameter from the Gaussian distribution. More specifically, for $\boldsymbol{\theta}^k, k \in [K]$, $\mathbf{W}_l$ is set to $\begin{pmatrix} \mathbf{w} & \mathbf{0} \\ \mathbf{0} & \mathbf{w} \end{pmatrix}$ for any $l \in [L_1]$ where $\mathbf{w}$ is drawn from $N(0, 4/m_1)$. For $\boldsymbol{\theta}^{\Sigma}$, $\mathbf{W}_l$ is set to $\begin{pmatrix} \mathbf{w} & \mathbf{0} \\ \mathbf{0} & \mathbf{w} \end{pmatrix}$ for any $l \in [L_2 - 1]$ where $\mathbf{w}$ is drawn from $N(0, 4/m_2)$; $\mathbf{W}_{L_2}$ is set to $(\mathbf{w}^\intercal, -\mathbf{w}^\intercal)$ where $\mathbf{w}$ is drawn from $N(0, 2/m_2)$.

4.2 Training process

Only with the final reward R_t and K selected arms $\mathbf{X}_t$, the training of the neural network model $\mathcal{F}$ is the following minimization problem:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T} \left(\mathcal{F}(\mathbf{X}_t; \boldsymbol{\theta}) - R_t \right)^2 / 2 + m_2 \lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 / 2. \quad (3)$$

where $\mathcal{L}(\boldsymbol{\theta})$ is essentially l_2-regularized square loss function and $\boldsymbol{\theta}_0$ is the randomly initialized network parameters. However, once the sub-rewards are available, we should use different methods to train $\mathcal{F}$, in order to leverage more available information. Next, we will elaborate our training methods using the gradient descend.

Collection of training samples. Depending on the availability of sub-rewards, we apply different strategies to update $\boldsymbol{\theta}$ in each round. When the sub-rewards are all available, the learner receives one final reward and K sub-rewards. We apply the straightforward way to train each part of $\mathcal{F}$ accordingly based on the corresponding input and the ground-truth rewards in each round, referring to the details in Algorithm 2, where $\widetilde{m}$ in $\mathcal{F}$ should be set as 1.

Algorithm 2 $GradientDescent_{All}$

Input: $\mathcal{F}, \{\mathbf{X}_i\}_{i=1}^t, \{R_i\}_{i=1}^t, \{\mathbf{r}_i\}_{i=1}^t, J, \eta$
Output: $\boldsymbol{\theta}_t$

for each $k \in [K]$ **do**
  Define $\mathcal{L}(\boldsymbol{\theta}^k) = \sum_{i=1}^t \left(f_k(\mathbf{x}_i^k; \boldsymbol{\theta}^k) - r_i^k \right)^2 / 2 + m_1 \lambda \|\boldsymbol{\theta}^k - \boldsymbol{\theta}_0\|_2^2 / 2$
  for each $j \in [J]$ **do**
    $\boldsymbol{\theta}_j^k = \boldsymbol{\theta}_{j-1}^k - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{j-1}^k)$
Define $\mathcal{L}(\boldsymbol{\theta}^{\Sigma}) = \sum_{i=1}^t \left(F(\mathrm{vec}(\mathbf{r}_i); \boldsymbol{\theta}^{\Sigma}) - R_i \right)^2 / 2 + m_2 \lambda \|\boldsymbol{\theta}^{\Sigma} - \boldsymbol{\theta}_0\|_2^2 / 2$.
for each $j \in [J]$ **do**
  $\boldsymbol{\theta}_j^{\Sigma} = \boldsymbol{\theta}_{j-1}^{\Sigma} - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{j-1}^{\Sigma})$
return $(\boldsymbol{\theta}_J^{\Sigma}, \boldsymbol{\theta}_J^1, \ldots, \boldsymbol{\theta}_J^K)$

Algorithm 3 $GradientDescent_{Partial}$

Input: $\mathcal{F}, \{\Omega_i\}_{i=1}^t, J, \eta$
Output: $\boldsymbol{\theta}_t$

Define $\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^t \sum_{(\mathbf{X},R) \in \Omega_i} \left(\mathcal{F}(\mathbf{X}; \boldsymbol{\theta}) - R \right)^2 / 2 + m_2 \lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 / 2$.
for each $j \in [J]$ **do**
  $\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{j-1})$
return $\boldsymbol{\theta}_j$

However, when the sub-rewards are partially available, the above method is not valid anymore because the bandits without available sub-rewards cannot be trained. Therefore, to learn the K bandits jointly, we propose the following training approach focusing on the empirical performance.

As the final reward is always available in each round, we collect the first training sample $(\mathbf{X}_t, R_t)$. Then, suppose there are $\mathcal{K}$ available sub-rewards $\mathbf{r}_t$, $\mathcal{K} < K$. For each available sub-reward $r_t^k \in \mathbf{r}_t$ and the corresponding context vector $\mathbf{x}_t^k$, we construct the following pair:

$$\widetilde{\mathbf{X}}_{t,k} = \{\mathbf{0}, \ldots, \mathbf{x}_t^k, \ldots, \mathbf{0}\} \text{ and } \widetilde{\mathbf{r}}_{t,k} = \{0, \ldots, r_t^k, \ldots, 0\}.$$

We regard $\widetilde{\mathbf{X}}_{t,k}$ as a new input of $\mathcal{F}$. Now, we need to determine the ground-truth final reward $\mathcal{H}(\widetilde{\mathbf{X}}_{t,k}) = H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k}))$.

Unfortunately, $H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k}))$ is unknown. Inspired by the UCB strategy, we determine $H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k}))$ by its upper bound. Based on Lemma 4.1, we have $H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k})) \leq \bar{C} r_t^k$. Therefore, we set $H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k}))$ as:

$$H(\mathrm{vec}(\widetilde{\mathbf{r}}_{t,k})) = \bar{C} r_t^k$$

because it shows the maximal potential gain for the bandit k. Then, in round t, we can collect additional $\mathcal{K}$ sample pairs:

$$\{(\widetilde{\mathbf{X}}_{t,k}, \bar{C} r_t^k)\}_{k \in [\mathcal{K}]}.$$

where $[\mathcal{K}]$ denotes the bandits with available sub-rewards.

Accordingly, in each round t, we can collect up to $\mathcal{K}+1$ samples for training $\mathcal{F}$, denoted by Ω_t,:

$$\Omega_t = \{(\widetilde{\mathbf{X}}_{t,k}, \bar{C} r_t^k)\}_{k \in [K]} \bigcup \{(\mathbf{X}_t, R_t)\}. \quad (4)$$

Therefore, in each round, we train $\mathcal{F}$ integrally, based on $\{\Omega_i\}_{i=1}^t$, as described in Algorithm 3.

Lemma 4.1. *Let $\mathbf{0} = (0, \ldots, 0)$ and $|\mathbf{0}| = K$. Given $\widetilde{\mathbf{X}}_{t,k}$ and $\widetilde{\mathbf{r}}_{t,k}$, then we have $H(vec(\widetilde{\mathbf{r}}_{t,k})) \leq \bar{C} r_t^k$.*

Prove 4.1. *As H is $\bar{C}$-Lipschitz continuous, we have*

$$|H(vec(\widetilde{\mathbf{r}}_{t,k})) - H(\mathbf{0})| \leq \bar{C} \sqrt{\sum_{r \in \widetilde{\mathbf{r}}_{t,k}} (r-0)^2} = \bar{C} r_t^k.$$

4.3 Upper confidence bound

In this subsection, we present the arm selection criterion based on the upper confidence bound provided in Section 5 and then summarize the high-level idea of MuFasa.

In each round t, given an arm combination $\mathbf{X}_t$, the confidence bound of $\mathcal{F}$ with respect to $\mathcal{H}$ is defined as:

$$\mathbb{P}\left(|\mathcal{F}(\mathbf{X}_t; \boldsymbol{\theta}_t) - \mathcal{H}(\mathbf{X}_t)| > \text{UCB}(\mathbf{X}_t)\right) \leq \delta,$$

where $\text{UCB}(\mathbf{X}_t)$ is defined in Theorem 5.3 and δ usually is a small constant. Then, in each round, given the all possible arm combinations $\mathbf{S}_t$, the selected K arms $\mathbf{X}_t$ are determined by:

$$\mathbf{X}_t = \arg\max_{\mathbf{X}'_t \in \mathbf{S}_t} \left(\mathcal{F}(\mathbf{X}'; \boldsymbol{\theta}_t) + \text{UCB}(\mathbf{X}'_t)\right). \tag{5}$$

With this selection criterion, the workflow of MuFasa is depicted in Algorithm 1.

5 REGRET ANALYSIS

In this section, we provide the upper confidence bound and regret analysis of MuFasa when the sub-rewards are all available.

Before presenting Theorem 5.3, let us first focus on an L-layer fully-connected neural network $f(\mathbf{x}_t; \boldsymbol{\theta})$ to learn a ground-truth function $h(\mathbf{x}_t)$, where $\mathbf{x} \in \mathbb{R}^d$. The parameters of f are set as $\mathbf{W}_1 \in \mathbb{R}^{m \times d}$, $\mathbf{W}_i \in \mathbb{R}^{m \times m}, \forall i \in [1 : L-1]$, and $\mathbf{W}_L \in \mathbb{R}^{1 \times m}$. Given the context vectors by $\{\mathbf{x}_i\}_{i=1}^T$ and corresponding rewards $\{r_t\}_{t=1}^T$, conduct the gradient descent with the loss $\mathcal{L}$ to train f.

Built upon the Neural Tangent Kernel (NTK) [15, 24], $h(\mathbf{x}_t)$ can be represented by a linear function with respect to the gradient $g(\mathbf{x}_t; \boldsymbol{\theta}_0)$ introduced in [45] as the following lemma.

Lemma 5.1 (Lemma 5.1 in [45]). *There exist a positive constant C such that with probability at least $1-\delta$, if $m \geq CT^4L^6 \log(T^2L/\delta)/\lambda^4$ for any $\mathbf{x}_t \in \{\mathbf{x}_t\}_{t=1}^T$, there exists a $\boldsymbol{\theta}^*$ such that*

$$h(\mathbf{x}_t) = \langle g(\mathbf{x}_t; \boldsymbol{\theta}_0), \boldsymbol{\theta}^* \rangle$$

Then, with the above linear representation of $h(\mathbf{x}_t)$, we provide the following upper confidence bound with regard to f.

Lemma 5.2. *Given a set of context vectors $\{\mathbf{x}_t\}_{t=1}^T$ and the corresponding rewards $\{r_t\}_{t=1}^T$, $\mathbb{E}(r_t) = h(\mathbf{x}_t)$ for any $\mathbf{x}_t \in \{\mathbf{x}_i\}_{i=1}^T$. Let $f(\mathbf{x}_t; \boldsymbol{\theta})$ be the L-layers fully-connected neural network where the width is m, the learning rate is η, and $\boldsymbol{\theta} \in \mathbb{R}^P$. Assuming $\|\boldsymbol{\theta}^*\|_2 \leq S/\sqrt{m}$, then, there exist positive constants C_1, C_2 such that if*

$$m \geq \max\{O\left(T^7\lambda^{-7}L^{21}(\log m)^3\right), O\left(\lambda^{-1/2}L^{-3/2}(\log(TL^2/\delta))^{3/2}\right)\}$$
$$\eta = O(TmL + m\lambda)^{-1}, \quad J \geq \widetilde{O}(TL/\lambda),$$

then, with probability at least $1-\delta$, for any $\mathbf{x}_t \in \{\mathbf{x}_t\}_{t=1}^T$, we have the following upper confidence bound:

$$\begin{aligned}\left|h(\mathbf{x}_t) - f(\mathbf{x}_t; \boldsymbol{\theta}_t)\right| \leq & \gamma_1 \|g(\mathbf{x}_t; \boldsymbol{\theta}_t)/\sqrt{m}\|_{\mathbf{A}_t^{-1}} + \gamma_2 \|g(\mathbf{x}_t; \boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t'^{-1}} \\ & + \gamma_1\gamma_3 + \gamma_4, \quad \textit{where}\end{aligned}$$

$$\gamma_1(m, L) = (\lambda + tO(L)) \cdot ((1 - \eta m\lambda)^{J/2}\sqrt{t/\lambda}) + 1$$
$$\gamma_2(m, L, \delta) = \sqrt{\log\left(\frac{\det(\mathbf{A}'_t)}{\det(\lambda \mathbf{I})}\right) - 2\log\delta} + \lambda^{1/2}S$$
$$\gamma_3(m, L) = C_2 m^{-1/6}\sqrt{\log m} t^{1/6}\lambda^{-7/6}L^{7/2}$$
$$\gamma_4(m, L) = C_1 m^{-1/6}\sqrt{\log m} t^{2/3}\lambda^{-2/3}L^3$$
$$\mathbf{A}_t = \lambda \mathbf{I} + \sum_{i=1}^t g(\mathbf{x}_t; \boldsymbol{\theta}_t) g(\mathbf{x}_t; \boldsymbol{\theta}_t)^\intercal / m$$
$$\mathbf{A}'_t = \lambda \mathbf{I} + \sum_{i=1}^t g(\mathbf{x}_t; \boldsymbol{\theta}_0) g(\mathbf{x}_t; \boldsymbol{\theta}_0)^\intercal / m.$$

Now we are ready to provide an extended upper confidence bound for the proposed neural network model $\mathcal{F}$.

Theorem 5.3. *Given the selected contexts $\{\mathbf{X}_t\}_{t=1}^T$, the final rewards $\{R_t\}_{t=1}^T$, and all sub-rewards $\{\mathbf{r}_t\}_{t=1}^T$, let $\mathcal{F}$ be the neural network model in MuFasa. In each round t, with the conditions in Lemma 5.2 and suppose $\widetilde{m} = 1$, then, with probability at least $1-\delta$, for any $t \in [T]$, we have the following upper confidence bound:*

$$|\mathcal{F}(\mathbf{X}_t; \boldsymbol{\theta}_t) - \mathcal{H}(\mathbf{X}_t)| \leq \bar{C} \sum_{k=1}^K \mathcal{B}^k + \mathcal{B}^F = UCB(\mathbf{X}_t), \ \textit{where}$$

$$\begin{aligned}\mathcal{B}^k = & \gamma_1 \|g_k(\mathbf{x}_t^k; \boldsymbol{\theta}_t^k)/\sqrt{m_1}\|_{\mathbf{A}_t^{k-1}} + \gamma_2 (\frac{\delta}{k+1}) \|g_k(\mathbf{x}_t^k; \boldsymbol{\theta}_0^k)/\sqrt{m_1}\|_{\mathbf{A}_t^{k'-1}} \\ & + \gamma_1\gamma_3 + \gamma_4\end{aligned}$$

$$\begin{aligned}\mathcal{B}^F = & \gamma_1 \|G(\mathbf{f}_t; \boldsymbol{\theta}_t^\Sigma)/\sqrt{m_2}\|_{\mathbf{A}_t^{F-1}} + \gamma_2 (\frac{\delta}{k+1}) \|G(\mathbf{f}_t; \boldsymbol{\theta}_0^\Sigma)/\sqrt{m_2}\|_{\mathbf{A}_t^{F'-1}} \\ & + \gamma_1\gamma_3 + \gamma_4\end{aligned}$$

$$\mathbf{A}_t^k = \lambda \mathbf{I} + \sum_{i=1}^t g_k(\mathbf{x}_i^k; \boldsymbol{\theta}_t^k) g_k(\mathbf{x}_i^k; \boldsymbol{\theta}_t^k)^\intercal / m_1$$
$$\mathbf{A}_t^{k'} = \lambda \mathbf{I} + \sum_{i=1}^t g_k(\mathbf{x}_i^k; \boldsymbol{\theta}_0^k) g_k(\mathbf{x}_i^k; \boldsymbol{\theta}_0^k)^\intercal / m_1$$
$$\mathbf{A}_t^F = \lambda \mathbf{I} + \sum_{i=1}^t G(\mathbf{f}_i; \boldsymbol{\theta}_t^\Sigma) G(\mathbf{f}_i; \boldsymbol{\theta}_t^\Sigma)^\intercal / m_2$$
$$\mathbf{A}_t^{F'} = \lambda \mathbf{I} + \sum_{i=1}^t G(\mathbf{f}_i; \boldsymbol{\theta}_0^\Sigma) G(\mathbf{f}_i; \boldsymbol{\theta}_0^\Sigma)^\intercal / m_2$$

With the above UCB, we provide the following regret bound of MuFasa.

Theorem 5.4. *Given the number of rounds T and suppose that the final reward and all the sub-wards are available, let $\mathcal{F}$ be the neural network model of MuFasa, satisfying the conditions in Theorem 5.3. Then, assuming $\bar{C} = 1$, $m_1 = m_2 = m$, $L_1 = L_2 = L$ and thus*

$P_1 = P_2 = P$, with probability at least $1-\delta$, the regret of MuFasa is upper bounded by:

$$\textbf{Reg} \leq (\bar{C}K+1)\sqrt{T}2\sqrt{\widetilde{P}\log(1+T/\lambda)+1/\lambda+1} \cdot \left(\sqrt{(\widetilde{P}-2)\log\left(\frac{(\lambda+T)(1+K)}{\lambda\delta}\right)+1/\lambda}+\lambda^{1/2}S+2\right)+2(\bar{C}K+1),$$

where $\widetilde{P}$ is the effective dimension defined in Appendix (Definition 8.3).

Prove 5.4. First, the regret of one round t :

$$\begin{aligned}\text{Reg}_t &= \mathcal{H}(\mathbf{X}_t^*) - \mathcal{H}(\mathbf{X}_t)\\ &\leq |\mathcal{H}(\mathbf{X}_t^*) - \mathcal{F}(\mathbf{X}_t^*)| + \mathcal{F}(\mathbf{X}_t^*) - \mathcal{H}(\mathbf{X}_t)\\ &\leq \text{UCB}(\mathbf{X}_t^*) + \mathcal{F}(\mathbf{X}_t^*) - \mathcal{H}(\mathbf{X}_t)\\ &\leq \text{UCB}(\mathbf{X}_t) + \mathcal{F}(\mathbf{X}_t) - \mathcal{H}(\mathbf{X}_t) \leq 2\text{UCB}(\mathbf{X}_t)\end{aligned}$$

where the third inequality is due to the selection criterion of MuFasa, satisfying $\mathcal{F}(\mathbf{X}_t^*) + \text{UCB}(\mathbf{X}_t^*) \leq \mathcal{F}(\mathbf{X}_t) + \text{UCB}(\mathbf{X}_t)$. Thus, it has

$$\textbf{Reg} = \sum_{t=1}^{T}\text{Reg}_t \leq 2\sum_{t=1}^{T}\text{UCB}(\mathbf{X}_t) \leq 2\sum_{t=1}^{T}\left(\bar{C}\sum_{k=1}^{K}\mathcal{B}^k + \mathcal{B}^F\right)$$

First, for any $k \in [K]$, we bound

$$\sum_{t=1}^{T}\mathcal{B}^k \leq \underbrace{\gamma_1\sum_{t=1}^{T}\|g(\mathbf{x}_t;\boldsymbol{\theta}_t)/\sqrt{m}\|^2_{\mathbf{A}_t^{-1}}}_{\mathbf{I}_1} + \underbrace{\gamma_2\sum_{t=1}^{T}\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|^2_{\mathbf{A}_t'^{-1}}}_{\mathbf{I}_2} + T\gamma_1\gamma_3 + T\gamma_4$$

Because the Lemma 11 in [1], we have

$$\begin{aligned}\mathbf{I}_1 &\leq \gamma_1\sqrt{T\left(\sum_{t=1}^{T}\|g(\mathbf{x}_t;\boldsymbol{\theta}_t)/\sqrt{m}\|^2_{\mathbf{A}_t^{-1}}\right)} \leq \gamma_1\sqrt{T\left(\log\frac{\det(\mathbf{A}_T)}{\det(\lambda\mathbf{I})}\right)}\\ &\leq \gamma_1\sqrt{T\left(\log\frac{\det(\mathbf{A}_T')}{\det\lambda\mathbf{I})} + |\log\frac{\det(\mathbf{A}_T)}{\det(\lambda\mathbf{I})} - \log\frac{\det(\mathbf{A}_T')}{\det(\lambda\mathbf{I})}|\right)}\\ &\leq \gamma_1\sqrt{T\left(\widetilde{P}\log(1+T/\lambda)+1/\lambda+1\right)}\end{aligned}$$

where the last inequality is based on Lemma 8.4 and the choice of m. Then, applying Lemma 11 in [1] and Lemma 8.4 again, we have

$$\begin{aligned}\mathbf{I}_2 &\leq \gamma_2\sqrt{T\left(\log\frac{\det(\mathbf{A}_T')}{\det(\lambda\mathbf{I})}\right)}\\ &\leq \left(\sqrt{(\widetilde{P}-2)\log\left(\frac{(\lambda+T)(1+K)}{\lambda\delta}\right)+1/\lambda}+\lambda^{1/2}S\right)\\ &\cdot\sqrt{T\left(\widetilde{P}\log(1+T/\lambda)+1/\lambda\right)}\end{aligned}$$

As the choice of J, $\gamma_1 \leq 2$. Then, as m is sufficiently large, we have

$$T\gamma_1\gamma_3 \leq 1, T\gamma_4 \leq 1.$$

Then, because $m_1 = m_2$, $L_1 = L_2$ and $\bar{C} = 1$, we have

$$\textbf{Reg} \leq 2(\bar{C}K+1)\sum_{t=1}^{T}\mathcal{B}^k.$$

Putting everything together proves the claim.

$\widetilde{P}$ is the effective dimension defined by the eigenvalues of the NTK (Definition 8.3 in Appendix). Effective dimension was first introduced by [39] to analyze the kernelized context bandit, and then was extended to analyze the kernel-based Q-learning[41] and the neural-network-based bandit [45]. $\widetilde{P}$ can be much smaller than the real dimension P, which alleviates the predicament when P is extremely large.

Theorem 5.4 provides the $\widetilde{O}\left((K+1)\sqrt{T}\right)$ regret bound for MuFasa, achieving the near-optimal bound compared with a single bandit ($\widetilde{O}(\sqrt{T})$) that is either linear [1] or non-linear [39, 45]. With different width m_1, m_2 and the Lipschitz continuity $\bar{C}$, the regret bound of MuFasa becomes $\widetilde{O}((\bar{C}K+1)\sqrt{T})$.

6 EXPERIMENTS

To evaluate the empirical performance of MuFasa, in this section, we design two different multi-facet bandit problems on three real-world data sets. The experiments are divided into two parts to evaluate the effects of final rewards and availability of sub-rewards. The code has been released [1].

Recommendation:Yelp[2]. Yelp is a data set released in the Yelp data set challenge, which consists of 4.7 million rating entries for 1.57×10^5 restaurants by 1.18 million users. In addition to the features of restaurants, this data set also provides the attributes of each user and the list of his/her friends. In this data set, we evaluate MuFasa on personalized recommendation, where the learner needs to simultaneously recommend a restaurant and a friend (user) to a served user. Naturally, this problem can be formulated into 2 bandits in which one set of arms $\mathbf{X}_t^1$ represent the candidate restaurants and the other set of arms $\mathbf{X}_t^2$ formulates the pool of friends for the recommendation. We apply LocallyLinearEmbedding[35] to train a 10-dimensional feature vector $\mathbf{x}_{t,i}^1$ for each restaurant and a 6-dimensional feature vector $\mathbf{x}_{t,j}^2$ for each user. Then, for the restaurant, we define the reward according to the rating star: The reward r_t^1 is 1 if the number of rating stars is more than 3 (5 in total); Otherwise, the reward r_t^1 is 0. For friends, the reward $r_t^2 = 1$ if the recommended friend is included in the friend list of the served user in fact; Otherwise $r_t^2 = 0$. To build the arm sets, we extract the rating entries and friends lists of top-10 users with the most ratings. In each round t, we build the arm set $\mathbf{X}_t^1$ and $\mathbf{X}_t^2$ by picking one restaurant/friend with 1 reward and then randomly picking the other 9 restaurants/friends with 0 rewards. Thus $|\mathbf{X}_t^1| = |\mathbf{X}_t^2| = 10$.

Classification:Mnist [27] + NotMnist. These are two well-known 10-class classification data sets. The evaluation of contextual bandit has been adapted to the classification problem [18, 39, 45]. Therefore, we utilize these two similar classification data sets to construct 2 bandits, where the 10-class classification is converted into a 10-armed contextual bandit. Considering a sample figure $\mathbf{x} \in \mathbb{R}^d$, we tend to classify it from 10 classes. Under the contextual bandit setting, $\mathbf{x}$ is transformed into 10 arms: $\mathbf{x}_1 = (\mathbf{x}, \mathbf{0}, \ldots, \mathbf{0}); \mathbf{x}_2 = (\mathbf{0}, \mathbf{x}, \ldots, \mathbf{0}); \ldots; \mathbf{x}_{10} = (\mathbf{0}, \mathbf{0}, \ldots, \mathbf{x}) \in \mathbb{R}^{10d}$ matching the 10 classes. In consequence, the reward is 1 if the learner plays the arm that

[1] https://github.com/banyikun/KDD2021_MuFasa
[2] https://www.yelp.com/dataset

matches the real class of $\mathbf{x}$; Otherwise, the reward is 0. Using this way, we can construct two contextual bandits for these two data sets, denoted by $(\mathbf{X}_t^1, r_t^1)$ and $(\mathbf{X}_t^2, r_t^2)$. Then, in each round, the arm pools will be $|\mathbf{X}_t^1| = |\mathbf{X}_t^2| = 10$.

To evaluate the effect of different final reward function, with the sub-rewards $\mathbf{r}_t = \{r_t^1, r_t^2\}$, we design the following final reward function:

$$H_1(\text{vec}(\mathbf{r}_t)) = r_t^1 + r_t^2; \;\; H_2(\text{vec}(\mathbf{r}_t)) = 2r_t^1 + r_t^2. \tag{6}$$

For (1), it describes the task where each bandit contributes equally. Of (2), it represents some tasks where each bandit has different importance.

As the problem setting is new, there are no existing algorithms that can directly adapt to this problem. Therefore, we construct baselines by extending the bandit algorithms that work on a single bandit, as follows:

(1) **(K-)LinUCB**. LinUCB [28] is a linear contextual bandit algorithm where the reward function is assumed as the dot product of the arm feature vector and an unknown user parameter. Then, apply the UCB strategy to select an arm in each round. To adapt to the multi-facet bandit problem, we duplicate LinUCB for K bandits. For example, in Yelp data set, we use two LinUCB to recommend restaurants and friends, respectively.
(2) **(K-)KerUCB** . KerUCB [39] makes use of a predefined kernel matrix to learn the reward function and then build a UCB for exploration. We replicate K KerUCB to adapt to this problem.
(3) **(K-)NeuUCB**. NeuUCB[45] uses a fully-connected neural network to learn one reward function with the UCB strategy. Similarly, we duplicate it to K bandits.

Configurations. For MuFasa, each sub-network $f_k(\mathbf{x}_t^k; \boldsymbol{\theta}^k)$ is set as a two-layer network: $f_k(\mathbf{x}_t^k; \boldsymbol{\theta}^k) = \sqrt{m_1}\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x}_t^k)$, where $\mathbf{W}_1 \in \mathbb{R}^{m_1 \times d_k}, \mathbf{W}_2 \in \mathbb{R}^{\widetilde{m} \times m_1}$, and $m_1 = \widetilde{m} = 100$. Then, the shared layers $F(\mathbf{f}_t; \boldsymbol{\theta}^\Sigma) = \sqrt{m_2}\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{f}_t)$, where $\mathbf{W}_1 \in \mathbb{R}^{m_2 \times 2\widetilde{m}}, \mathbf{W}_2 \in \mathbb{R}^{1 \times m_2}$ and $m_2 = 100$. For the H_1, $\tilde{C}$ is set as 1 and set as 2 for H_2. To learn K bandits jointly, in the experiments, we evaluate the performance of Algorithm 1 + 3. For K-NeuUCB, for each NeuUCB, we set it as a 4-layer fully-connected network with the same width $m = 100$ for the fair comparison. The learning rate η is set as 0.01 and the upper bound of ground-truth parameter $S = 1$ for these two methods. To accelerate the training process, we update the parameters of the neural networks every 50 rounds. For the KerUCB, we use the radial basis function (RBF) kernel and stop adding contexts to KerUCB after 1000 rounds, following the same setting for Gaussian Process in [34, 45]. For all the methods, the confidence level $\delta = 0.1$, the regularization parameter $\lambda = 1$. All experiments are repeatedly run 5 times and report the average results.

6.1 Result 1: All sub-rewards with different H

With the final reward function H_1 (Eq.(6)), Figure 1 and Figure 3 report the regret of all methods on Yelp and Mnist+NotMnist data sets, where the first top-two sub-figure shows the regret of each bandit and the bottom sub-figure shows the regret of the whole task (2 bandits). These figures show that MuFasa achieves the best performance (the smallest regret), because it utilizes a comprehensive upper confidence bound built on the assembled neural

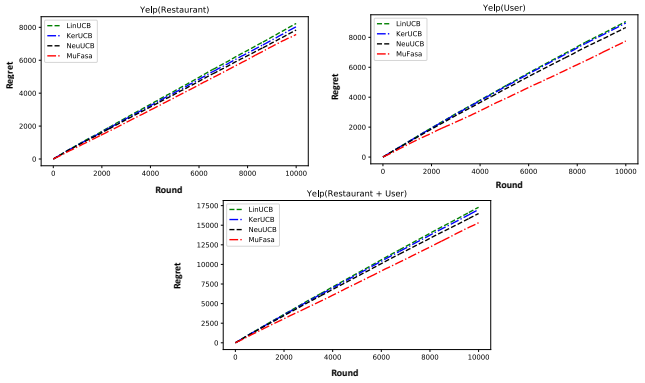

Figure 1: Regret comparison on Yelp with H_1 final reward function.

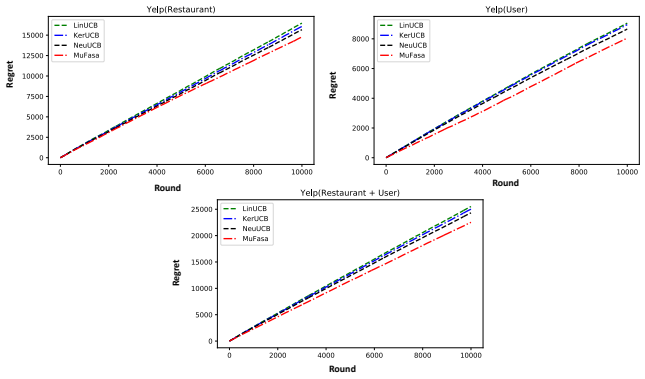

Figure 2: Regret comparison on Yelp with H_2 final reward function.

networks to select two arms jointly in each round. This indicates the good performance of MuFasa on personalized recommendation and classification. Among these baselines, NeuUCB achieves the best performance, which thanks to the representation power of neural networks. However, it chooses each arm separately, neglecting the collaborative relation of K bandits. For KerUCB, it shows the limitation of the simple kernels like the radial basis function compared to neural network. LinUCB fails to handle each task, as it assume a linear reward function and thus usually cannot to learn the complicated reward functions in practice.

With the final reward function H_2 (Eq.(6)), Figure 2 and Figure 4 depict the regret comparison on Yelp and Mnist+NotMnist data sets. The final reward function H_2 indicates that the bandit 1 weights more than bandit 2 in the task. Therefore, to minimize the regret, the algorithm should place the bandit 1 as the priority when making decisions. As the design of MuFasa, the neural network $\mathcal{F}$ can learn the relation among the bandits. For example, on the Mnist and NotMnist data sets, consider two optional select arm sets $\{x_{t,i_1}^1, x_{t,i_2}^2\}$ and $\{x_{t,j_1}^1, x_{t,j_2}^2\}$. The first selected arm set receives 1 reward on Mnist while 0 reward on NotMnist. In contrast, the second selected arm set receives 0 reward on Mnist while 1 reward on NotMnist. However, these two bandits have different weights

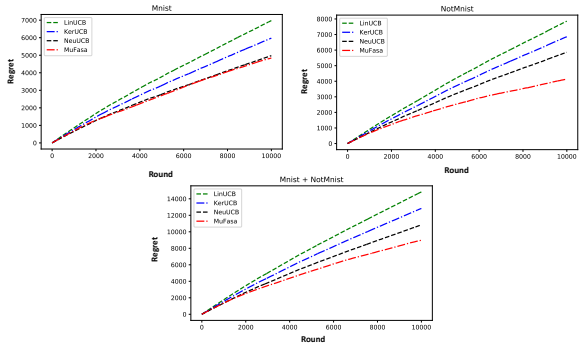

Figure 3: Regret comparison on Mnist+NotMnist with H_1.

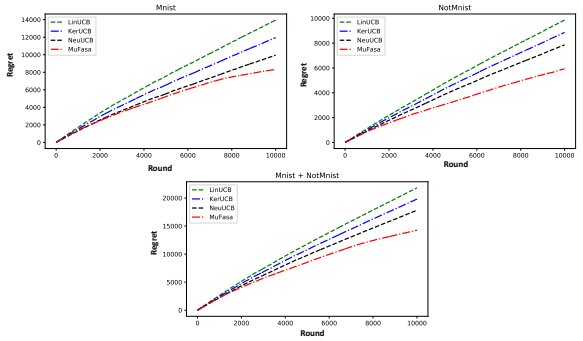

Figure 4: Regret comparison on Mnist+NotMnist with H_2.

and thus the two arm sets have different final rewards, i.e., $R_t^1 = 2$ and $R_t^2 = 1$, respectively. To maximize the final reward, the learner should select the first arm set instead of the second arm set. As MuFasa can learn the weights of bandits, it will give more weight to the first bandit and thus select the first arm set. On the contrary, all the baselines treat each bandit equally, and thus they will select these two arm sets randomly. Therefore, under the setting of H_2, with this advantage, MuFasa further decreases the regret on both Yelp and Mnist+NotMnist data sets. For instance, on Mnist+NotMnist data sets, MuFasa with H_2 decrease 20% regret over NeuUCB while MuFasa with H_1 decrease 17.8% regret over NeuUCB.

6.2 Result 2: Partial sub-rewards

As the sub-rewards are not always available in many cases, in this subsection, we evaluate MuFasa with partially available sub-rewards on Yelp and Mnist+NotMnist data sets. Therefore, we build another two variants of MuFasa: (1) MuFasa (One sub-reward) is provided with the final reward and one sub-reward of the first bandit; (2) MuFasa (No sub-reward) does not receive any sub-rewards except the final reward. Here, we use the H_1 as the final reward function.

Figure 5 and Figure 6 show the regret comparison with the two variants of MuFasa, where MuFasa exhibits the robustness with respect to the lack of sub-rewards. Indeed, the sub-reward can provide more information to learn, while MuFasa (One sub-reward) still outperforms all the baselines, because the final reward enables MuFasa to learn the all bandits jointly and the sub-reward strengthens the capacity of learning the exclusive features of each bandit.

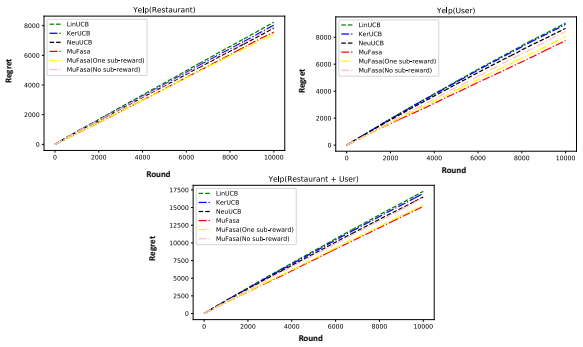

Figure 5: Regret comparison on Yelp with different reward availability.

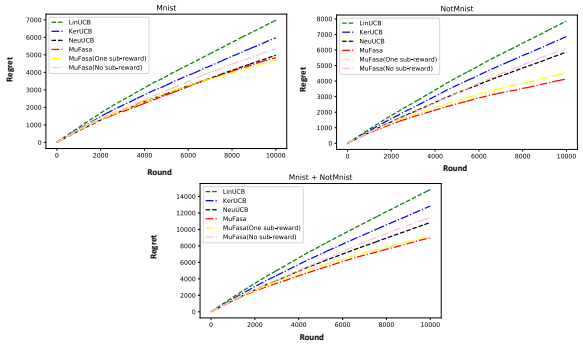

Figure 6: Regret comparison on Mnist+NotMnist with different reward availability.

In contrast, all the baselines treat each bandit separately. Without any-rewards, MuFasa still achieves the acceptable performance. On the Yelp data set, the regret of MuFasa (No sub-reward) is still lower than the best baseline NeuUCB while lacking considerable information. On the Mnist+NotMnist data set, although MuFasa (No sub-reward) does not outperform the baselines, its performance is still closed to NeuUCB. Therefore, as long as the final reward is provided, MuFasa can tackle the multi-facet problem effectively. Moreover, MuFasa can leverage available sub-rewards to improve the performance.

7 CONCLUSION

In this paper, we define and study the novel problem of the multi-facet contextual bandits, motivated by real applications such as comprehensive personalized recommendation and healthcare. We propose a new bandit algorithm, MuFasa. It utilizes the neural networks to learn the reward functions of multiple bandits jointly and explores new information by a comprehensive upper confidence bound. Moreover, we prove that MuFasa can achieve the $\widetilde{O}((K+1)\sqrt{T})$ regret bound under mild assumptions. Finally, we conduct extensive experiments to show the effectiveness of MuFasa on personalized recommendation and classification tasks, as well as the robustness of MuFasa in the lack of sub-rewards.

ACKNOWLEDGEMENT

This work is supported by National Science Foundation under Award No. IIS-1947203 and IIS-2002540, and the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-03-03. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*. 2312–2320.

[2] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*. PMLR, 127–135.

[3] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*. PMLR, 242–252.

[4] Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. 2014. A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*. Springer, 374–381.

[5] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. 2019. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*. 8141–8150.

[6] Jean-Yves Audibert and Sébastien Bubeck. 2010. Best arm identification in multi-armed bandits. In *Conference on Learning Theory (COLT)*. 41–53.

[7] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[8] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[9] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. 2018. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 1–9.

[10] Yikun Ban and Jingrui He. 2020. Generic Outlier Detection in Multi-Armed Bandit. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 913–923.

[11] Yikun Ban and Jingrui He. 2021. Local Clustering in Contextual Multi-Armed Bandits. *arXiv preprint arXiv:2103.00063* (2021).

[12] Hamsa Bastani and Mohsen Bayati. 2020. Online decision making with high-dimensional covariates. *Operations Research* 68, 1 (2020), 276–294.

[13] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. 2011. X-Armed Bandits. *Journal of Machine Learning Research* 12, 5 (2011).

[14] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. 2013. Multi-armed bandits in the presence of side observations in social networks. In *52nd IEEE Conference on Decision and Control*. IEEE, 7309–7314.

[15] Yuan Cao and Quanquan Gu. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*. 10836–10846.

[16] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*. PMLR, 151–159.

[17] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research* 17, 1 (2016), 1746–1778.

[18] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. 2017. Multi-task learning for contextual bandits. In *Advances in neural information processing systems*. 4848–4856.

[19] Maria Dimakopoulou, Zhengyuan Zhou, Susan Athey, and Guido Imbens. 2019. Balanced linear contextual bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3445–3453.

[20] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. 2018. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare Conference*. 67–82.

[21] Dongqi Fu, Zhe Xu, Bo Li, Hanghang Tong, and Jingrui He. 2020. A View-Adversarial Framework for Multi-View Network Embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2025–2028.

[22] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1253–1262.

[23] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online clustering of bandits. In *International Conference on Machine Learning*. 757–765.

[24] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*. 8571–8580.

[25] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. HDMI: High-order Deep Multiplex Infomax. *arXiv preprint arXiv:2102.07810* (2021).

[26] John Langford and Tong Zhang. 2008. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*. 817–824.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[28] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[29] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 539–548.

[30] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual combinatorial cascading bandits. In *International conference on machine learning*. PMLR, 1245–1253.

[31] Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[32] Haoyang Liu, Keqin Liu, and Qing Zhao. 2011. Logarithmic weak regret of non-bayesian restless multi-armed bandit. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1968–1971.

[33] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. 2014. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 461–469.

[34] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* (2018).

[35] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[36] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995* (2009).

[37] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[38] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.

[39] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. 2013. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869* (2013).

[40] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. 2016. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 529–538.

[41] Lin Yang and Mengdi Wang. 2020. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*. PMLR, 10746–10756.

[42] Tom Zahavy and Shie Mannor. 2019. Deep neural linear bandits: Overcoming catastrophic forgetting through likelihood matching. *arXiv preprint arXiv:1901.08612* (2019).

[43] Lecheng Zheng, Yu Cheng, Hongxia Yang, Nan Cao, and Jingrui He. 2021. Deep Co-Attention Network for Multi-View Subspace Learning. *arXiv preprint arXiv:2102.07751* (2021).

[44] Dawei Zhou, Jingrui He, K Selçuk Candan, and Hasan Davulcu. 2015. MUVIR: Multi-View Rare Category Detection.. In *IJCAI*. Citeseer, 4098–4104.

[45] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with UCB-based exploration. In *International Conference on Machine Learning*. PMLR, 11492–11502.

[46] Dawei Zhou, Lecheng Zheng, Yada Zhu, Jianbo Li, and Jingrui He. 2020. Domain adaptive multi-modality neural attention network for financial forecasting. In *Proceedings of The Web Conference 2020*. 2230–2240.

[47] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabadi, Evren Korpeoglu, Kannan Achan, and Jingrui He. 2020. GAN-based Recommendation with Positive-Unlabeled Sampling. *arXiv preprint arXiv:2012.06901* (2020).

8 APPENDIX

Definition 8.1. Given the context vectors $\{\mathbf{x}_t\}_{t=1}^T$ and the rewards $\{r_t\}_{t=1}^T$, then we define the estimation $\boldsymbol{\theta}'$ via ridge regression:

$$\mathbf{A}_t' = \lambda\mathbf{I} + \sum_{i=1}^{t} g(\mathbf{x}_t;\boldsymbol{\theta}_0)g(\mathbf{x}_t;\boldsymbol{\theta}_0)^\top/m$$
$$\mathbf{b}_t' = \sum_{i=1}^{t} r_t g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}$$
$$\boldsymbol{\theta}' = \mathbf{A}_t^{-1}\mathbf{b}_t$$
$$\mathbf{A}_t = \lambda\mathbf{I} + \sum_{i=1}^{t} g(\mathbf{x}_t;\boldsymbol{\theta}_t)g(\mathbf{x}_t;\boldsymbol{\theta}_t)^\top/m$$

Definition 8.2 (NTK [15, 24]). Let $\mathcal{N}$ denote the normal distribution. Define

$$\mathbf{M}_{i,j}^0 = \Sigma_{i,j}^0 = \langle \mathbf{x}_i, \mathbf{x}_j\rangle,\ \ \mathbf{N}_{i,j}^l = \begin{pmatrix}\Sigma_{i,i}^l & \Sigma_{i,j}^l\\ \Sigma_{j,i}^l & \Sigma_{j,j}^l\end{pmatrix}$$
$$\Sigma_{i,j}^l = 2\mathbb{E}_{a,b\sim\mathcal{N}(\mathbf{0},\mathbf{N}_{i,j}^{l-1})}[\sigma(a)\sigma(b)]$$
$$\mathbf{M}_{i,j}^l = 2\mathbf{M}_{i,j}^{l-1}\mathbb{E}_{a,b\sim\mathcal{N}(\mathbf{0},\mathbf{N}_{i,j}^{l-1})}[\sigma'(a)\sigma'(b)] + \Sigma_{i,j}^l.$$

Then, given the contexts $\{\mathbf{x}_t\}_{t=1}^T$, the Neural Tangent Kernel is defined as $\mathbf{M} = (\mathbf{M}^L + \Sigma^L)/2$.

Definition 8.3 (Effective Dimension [45]). Given the contexts $\{\mathbf{x}_t\}_{t=1}^T$, the effective dimension $\widetilde{P}$ is defined as

$$\widetilde{P} = \frac{\log\det(\mathbf{I}+\mathbf{M}/\lambda)}{\log(1+T/\lambda)}.$$

Proof of Lemma 5.2. Given a set of context vectors $\{\mathbf{x}\}_{t=1}^T$ with the ground-truth function h and a fully-connected neural network f, we have

$$\begin{aligned}&\left|h(\mathbf{x}_t) - f(\mathbf{x}_t;\boldsymbol{\theta}_t)\right|\\ \le&\left|h(\mathbf{x}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right| + \left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right|\end{aligned}$$

where $\boldsymbol{\theta}'$ is the estimation of ridge regression from Definition 8.1. Then, based on the Lemma 5.1, there exists $\boldsymbol{\theta}^* \in \mathbf{R}^P$ such that $h(\mathbf{x}_t) = \langle g(\mathbf{x}_i,\boldsymbol{\theta}_0), \boldsymbol{\theta}^*\rangle$. Thus, we have

$$\begin{aligned}&\left|h(\mathbf{x}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right|\\ =&\left|\left\langle g(\mathbf{x}_i,\boldsymbol{\theta}_0)/\sqrt{m}, \sqrt{m}\boldsymbol{\theta}^*\right\rangle - \left\langle g(\mathbf{x}_i,\boldsymbol{\theta}_0)/\sqrt{m}, \boldsymbol{\theta}'\right\rangle\right| \le\\ &\left(\sqrt{\log\left(\frac{\det(\mathbf{A}_t')}{\det(\lambda\mathbf{I})}\right) - 2\log\delta} + \lambda^{1/2}S\right)\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t'^{-1}}\end{aligned}$$

where the final inequality is based on the the Theorem 2 in [1], with probability at least $1-\delta$, for any $t \in [T]$.

Second we need to bound

$$\begin{aligned}&\left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right|\\ \le&\left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0\rangle\right|\\ &+\left|\langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0\rangle - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right|\end{aligned}$$

To bound the above inequality, we first bound

$$\begin{aligned}&\left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0\rangle\right|\\ =&\left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - f(\mathbf{x}_t;\boldsymbol{\theta}_0) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0\rangle\right|\\ \le&C_2\tau^{4/3}L^3\sqrt{m\log m} \le C_2 m^{-1/6}\sqrt{\log m}t^{2/3}\lambda^{-2/3}L^3.\end{aligned}$$

where $f(\mathbf{x}_t;\boldsymbol{\theta}_0) = 0$ due to the random initialization of $\boldsymbol{\theta}_0$. The first inequality is derived by Lemma 8.5. According to the Lemma 8.7, it has $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2 \le 2\sqrt{\frac{t}{m\lambda}}$. Then, replacing τ by $2\sqrt{\frac{t}{m\lambda}}$, we obtain the second inequality.

Next, we need to bound

$$\begin{aligned}&|\langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0\rangle - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle|\\ =&|\langle g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}, \sqrt{m}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \boldsymbol{\theta}'/\sqrt{m})\rangle|\\ \le&\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\cdot\sqrt{m}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \boldsymbol{\theta}'/\sqrt{m}\|_{\mathbf{A}_t}\\ \le&\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\cdot\sqrt{m}\|\mathbf{A}_t\|_2\cdot\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \boldsymbol{\theta}'/\sqrt{m}\|_2.\end{aligned}$$

Due to the Lemma 8.6 and Lemma 8.7, we have

$$\begin{aligned}&\sqrt{m}\|\mathbf{A}_t\|_2\cdot\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \boldsymbol{\theta}'/\sqrt{m}\|_2 \le \sqrt{m}(\lambda + tO(L))\\ &\cdot\left((1-\eta m\lambda)^{J/2}\sqrt{t/(m\lambda)} + C_4 m^{-2/3}\sqrt{\log m}L^{7/2}t^{5/3}\lambda^{-5/3}(1+\sqrt{t/\lambda})\right)\\ &\le (\lambda + tO(L))\\ &\cdot((1-\eta m\lambda)^{J/2}\sqrt{t/\lambda} + C_4 m^{-1/6}\sqrt{\log m}L^{7/2}t^{5/3}\lambda^{-5/3}(1+\sqrt{t/\lambda}))\\ &\le (\lambda + tO(L))\cdot((1-\eta m\lambda)^{J/2}\sqrt{t/\lambda}) + 1\end{aligned}$$

where the last inequality is because m is sufficiently large. Therefore, we have

$$\begin{aligned}&\left|f(\mathbf{x}_t;\boldsymbol{\theta}_t) - \langle g(\mathbf{x}_t;\boldsymbol{\theta}_0), \boldsymbol{\theta}'/\sqrt{m}\rangle\right|\\ \le&\left((\lambda + tO(L))\cdot((1-\eta m\lambda)^{J/2}\sqrt{t/\lambda}) + 1\right)\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\\ &+ C_2 m^{-1/6}\sqrt{\log m}t^{2/3}\lambda^{-2/3}L^3\end{aligned}$$

And we have

$$\begin{aligned}&\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\\ =&\|g(\mathbf{x}_t;\boldsymbol{\theta}_t) + g(\mathbf{x}_t;\boldsymbol{\theta}_0) - g(\mathbf{x}_t;\boldsymbol{\theta}_t)\|_{\mathbf{A}_t^{-1}}/\sqrt{m}\\ \le&\|g(\mathbf{x}_t;\boldsymbol{\theta}_t)/\sqrt{m}\|_{\mathbf{A}_t^{-1}} + \|\mathbf{A}_t^{-1}\|_2\|g(\mathbf{x}_t;\boldsymbol{\theta}_0) - g(\mathbf{x}_t;\boldsymbol{\theta}_t)\|_2/\sqrt{m}\\ \le&\|g(\mathbf{x}_t;\boldsymbol{\theta}_t)/\sqrt{m}\|_{\mathbf{A}_t^{-1}} + \lambda^{-1}m^{-1/6}\sqrt{\log m}t^{1/6}\lambda^{-1/6}L^{7/2}\end{aligned}$$

where the last inequality is because of Lemma 8.8 with Lemma 8.7 and $\|\mathbf{A}_t\|_2 \ge \|\lambda\mathbf{I}\|_2$.

Finally, putting everything together, we have

$$\begin{aligned}\left|h(\mathbf{x}_t) - f(\mathbf{x}_t;\boldsymbol{\theta}_t)\right| \le&\ \gamma_1\|g(\mathbf{x}_t;\boldsymbol{\theta}_t)/\sqrt{m}\|_{\mathbf{A}_t^{-1}} + \gamma_2\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)/\sqrt{m}\|_{\mathbf{A}_t'^{-1}}\\ &+\gamma_1\gamma_3 + \gamma_4.\end{aligned}$$

Proof of Theorem 5.3. First, considering an individual bandit $k \in [K]$ with the set of context vectors $\{\mathbf{x}_t^k\}_{t=1}^T$ and the set of sub-rewards $\{r_t^k\}_{t=1}^T$, we can build upper confidence bound of $f_k(\mathbf{x}_t^k;\boldsymbol{\theta}_t^k)$ with respect to $h_k(\mathbf{x}_t^k)$ based on the Lemma 5.2. Denote the UCB by $\mathcal{B}(\mathbf{x}_t^k, m, L_1, \delta')$, with probability at least $1-\delta'$, for any $t \in [T]$ we have

$$|f_k(\mathbf{x}_t^k;\boldsymbol{\theta}_t^k) - h_k(\mathbf{x}_t^k)| \le \mathcal{B}(\mathbf{x}_t^k, m, L_1, \delta', t) = \mathcal{B}^k.$$

Next, apply the union bound on the $K+1$ networks, we have $\delta' = \delta/(K+1)$ in each round t.

Next we need to bound

$$\begin{aligned}&\left|H\left(\text{vec}(\mathbf{r}_t)\right) - H\left(\mathbf{f}_t\right)\right| \\ \leq &\bar{C}\sqrt{\sum_{k=1}^{K}|f_k(\mathbf{x}_t^k;\boldsymbol{\theta}_t^k) - h_k(\mathbf{x}_t^k)|^2} \\ \leq &\bar{C}\sqrt{\sum_{k=1}^{K}(\mathcal{B}^k)^2} \leq \bar{C}\sum_{k=1}^{K}\mathcal{B}^k\end{aligned}$$

where the first inequality is because H is a $\bar{C}$-lipschitz continuous function. Therefore, we have

$$\begin{aligned}&|\mathcal{F}(\mathbf{X}_t) - \mathcal{H}(\mathbf{X}_t)| = |F(\mathbf{f}_t;\boldsymbol{\theta}^\Sigma) - H(\text{vec}(\mathbf{r}_t))| \\ \leq &\left|F(\mathbf{f}_t;\boldsymbol{\theta}^\Sigma) - H\left(\mathbf{f}_t\right)\right| + \left|H\left(\mathbf{f}_t\right) - H(\text{vec}(\mathbf{r}_t))\right| \leq \bar{C}\sum_{k=1}^{K}\mathcal{B}^k + \mathcal{B}^F.\end{aligned}$$

This completes the proof of the claim.

Lemma 8.4. *With probability at least $1-\delta'$, we have*

$$\begin{aligned}(1)&\|\mathbf{A}_t\|_2, \|\mathbf{A}_t'\|_2 \leq \lambda + tO(L) \\ (2)&\log\frac{\det(\mathbf{A}_t')}{\det(\lambda\mathbf{I})} \leq \widetilde{P}\log(1+T/\lambda) + 1/\lambda \\ (3)&|\log\frac{\det(\mathbf{A}_t)}{\det(\lambda\mathbf{I})} - \log\frac{\det(\mathbf{A}_t')}{\det(\lambda\mathbf{I})}| \leq O(m^{-1/6}\sqrt{\log m}L^4t^{5/3}\lambda^{-1/6}).\end{aligned}$$

where (3) *is referred from Lemma B.3 in [45].*

Proof of Lemma 8.4. For (1), based on the Lemma 8.6, for any $\mathbf{x}_t \in \{\mathbf{x}_i\}_{i=1}^T$,
$\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)\|_F \leq O(\sqrt{mL})$. Then, for the first item:

$$\begin{aligned}\|\mathbf{A}_t\|_2 &= \|\lambda\mathbf{I} + \sum_{i=1}^{t} g(\mathbf{x}_i;\boldsymbol{\theta}_t)g(\mathbf{x}_i;\boldsymbol{\theta}_t)^\intercal/m\|_2 \\ &\leq \|\lambda\mathbf{I}\|_2 + \|\sum_{i=1}^{t} g(\mathbf{x}_i;\boldsymbol{\theta}_t)g(\mathbf{x}_i;\boldsymbol{\theta}_t)^\intercal/m\|_2 \\ &\leq \lambda + \sum_{i=1}^{t}\|g(\mathbf{x}_i;\boldsymbol{\theta}_t)\|_2^2/m \leq \lambda + \sum_{i=1}^{t}\|g(\mathbf{x}_i;\boldsymbol{\theta}_t)\|_F^2/m \\ &\leq \lambda + tO(L).\end{aligned}$$

Same proof workflow for $\|\mathbf{A}_t'\|_2$. For (2), we have

$$\begin{aligned}\log\frac{\det(\mathbf{A}_t')}{\det(\lambda\mathbf{I})} &= \log\det(\mathbf{I} + \sum_{t=1}^{T} g(\mathbf{x}_t;\boldsymbol{\theta}_0)g(\mathbf{x}_t;\boldsymbol{\theta}_0)^\intercal/(m\lambda)) \\ &= \det(\mathbf{I} + \mathbf{G}\mathbf{G}^\intercal/\lambda)\end{aligned}$$

where $\mathbf{G} = (g(\mathbf{x}_1;\boldsymbol{\theta}_0), \ldots, g(\mathbf{x}_T;\boldsymbol{\theta}_0))/\sqrt{m}$.

According to the Theorem 3.1 in [5], when $m = \Omega(\frac{L^6\log L/\delta}{\epsilon^4})$, with probability at least $1-\delta$, for any $\mathbf{x}_i, \mathbf{x}_j \in \{\mathbf{x}_t\}_{t=1}^T$, it has

$$|g(\mathbf{x}_i;\boldsymbol{\theta}_0)^\intercal g(\mathbf{x}_j;\boldsymbol{\theta}_0)/m - \mathbf{M}_{i,j}| \leq \epsilon.$$

Therefore, we have

$$\begin{aligned}\|\mathbf{G}\mathbf{G}^\intercal - \mathbf{M}\|_F &= \sqrt{\sum_{i=1}^{T}\sum_{j=1}^{T}|g(\mathbf{x}_i;\boldsymbol{\theta}_0)^\intercal g(\mathbf{x}_j;\boldsymbol{\theta}_0)/m - \mathbf{M}_{i,j}|^2} \\ &\leq T\epsilon.\end{aligned}$$

Then, we have

$$\begin{aligned}&\log\det(\mathbf{I} + \mathbf{G}\mathbf{G}^\intercal/\lambda) \\ &= \log\det(\mathbf{I} + \mathbf{M}\lambda + (\mathbf{G}\mathbf{G}^\intercal - \mathbf{M})/\lambda) \\ &\leq \log\det(\mathbf{I} + \mathbf{M}\lambda) + \langle(\mathbf{I} + \mathbf{M}\lambda)^{-1}, (\mathbf{G}\mathbf{G}^\intercal - \mathbf{M})/\lambda\rangle \\ &\leq \log\det(\mathbf{I} + \mathbf{M}\lambda) + \|(\mathbf{I} + \mathbf{M}\lambda)^{-1}\|_F\|\mathbf{G}\mathbf{G}^\intercal - \mathbf{M}\|_F/\lambda \\ &\leq \log\det(\mathbf{I} + \mathbf{M}\lambda) + \sqrt{T}\|\mathbf{G}\mathbf{G}^\intercal - \mathbf{M}\|_F/\lambda \\ &\leq \log\det(\mathbf{I} + \mathbf{M}\lambda) + \lambda^{-1} \\ &= \widetilde{P}\log(1 + T/\lambda) + \lambda^{-1}.\end{aligned}$$

The first inequality is because the concavity of $\log\det$; The third inequality is due to $\|(\mathbf{I} + \mathbf{M}\lambda)^{-1}\|_F \leq \|\mathbf{I}^{-1}\|_F \leq \sqrt{T}$; The last inequality is because of the choice the m; The last equality is because of the Definition 8.3.

Lemma 8.5 (Lemma 4.1 in [15]). *There exist constants $\{\bar{C}_{i=1}^3\} \geq 0$ such that for any $\delta \geq 0$, if τ satisfies that*

$$\tau \leq \bar{C}_2L^{-6}[\log m]^{-3/2},$$

then with probability at least $1-\delta$, for all $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2$ satisfying $\|\boldsymbol{\theta}^1 - \boldsymbol{\theta}_0\| \leq \tau$, $\|\boldsymbol{\theta}^2 - \boldsymbol{\theta}_0\| \leq \tau$ and for any $\mathbf{x}_t \in \{\mathbf{x}_t\}_{t=1}^T$, we have

$$|f(\mathbf{x};\boldsymbol{\theta}^1) - f(\mathbf{x};\boldsymbol{\theta}^2) - \langle(g(\mathbf{x};\boldsymbol{\theta}^2), \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2)\rangle| \leq \bar{C}_3\tau^{4/3}L^3\sqrt{m\log m}.$$

Lemma 8.6 (Lemma B.3 in [15]). *There exist constants $\{C_i\}_{i=1}^2$ such that for any $\delta > 0$, if τ satisfies that*

$$\tau \leq C_1L^{-6}(\log m)^{-3/2},$$

then, with probability at least $1-\delta$, for any $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\| \leq \tau$ and $\mathbf{x}_t \in \{\mathbf{x}_t\}_{t=1}^T$ we have $\|g(\mathbf{x}_t;\boldsymbol{\theta})\|_2 \leq C_2\sqrt{mL}$.

Lemma 8.7 (Lemma B.2 in [45]). *For the L-layer full-connected network f, there exist constants $\{C_i\}_{i=1}^5 \geq 0$ such that for $\delta > 0$, if for all $t \in [T]$, η, m satisfy*

$$\begin{aligned}&2\sqrt{t/(m\lambda)} \geq C_1m^{-3/2}L^{-3/2}[\log(TL^2/\delta)]^{3/2}, \\ &2\sqrt{t/(m\lambda)} \leq C_2\min\{L^{-6}[\log m]^{-3/2}, (m(\lambda\eta)^2L^{-6}t^{-1}(\log m)^{-1})^{3/8}\}, \\ &\eta \leq C_3(m\lambda + tmL)^{-1}, \\ &m^{1/6} \geq C_4\sqrt{\log m}L^{7/2}t^{7/6}\lambda^{-7/6}(1 + \sqrt{t/\lambda}),\end{aligned}$$

then, with probability at least $1-\delta$, it has

$$\begin{aligned}&\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\| \leq 2\sqrt{t/(m\lambda)} \\ &\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \boldsymbol{\theta}'\| \leq \\ &(1-\eta m\lambda)^{J/2}\sqrt{t/(m\lambda)} + C_5m^{-2/3}\sqrt{\log m}L^{7/2}t^{5/3}\lambda^{-5/3}(1 + \sqrt{t/\lambda}).\end{aligned}$$

Lemma 8.8 (Theorem 5 in [3]). *With probability at least $1-\delta$, there exist constants C_1, C_2 such that if $\tau \leq C_1L^{-9/2}\log^{-3}m$, for $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2 \leq \tau$, we have*

$$\|g(\mathbf{x}_t;\boldsymbol{\theta}_t) - g(\mathbf{x}_t;\boldsymbol{\theta}_0)\|_2 \leq C_2\sqrt{\log m}\tau^{1/3}L^3\|g(\mathbf{x}_t;\boldsymbol{\theta}_0)\|_2.$$