Neural Bandit with Arm Group Graph

Yunzhe Qi  
University of Illinois at Urbana-Champaign  
yunzheq2@illinois.edu

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

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

ABSTRACT
Contextual bandits aim to identify among a set of arms the optimal one with the highest reward based on their contextual information. Motivated by the fact that the arms usually exhibit group behaviors and the mutual impacts exist among groups, we introduce a new model, Arm Group Graph (AGG), where the nodes represent the groups of arms and the weighted edges formulate the correlations among groups. To leverage the rich information in AGG, we propose a bandit algorithm, AGG-UCB, where the neural networks are designed to estimate rewards, and we propose to utilize graph neural networks (GNN) to learn the representations of arm groups with correlations. To solve the exploitation-exploration dilemma in bandits, we derive a new upper confidence bound (UCB) built on neural networks (exploitation) for exploration. Furthermore, we prove that AGG-UCB can achieve a near-optimal regret bound with over-parameterized neural networks (GNN) to learn the representations of arm groups with correlations. To solve the exploitation-exploration dilemma in bandits, we derive a new upper confidence bound (UCB) built on neural networks (exploitation) for exploration. Furthermore, we prove that AGG-UCB can achieve a near-optimal regret bound with over-parameterized neural networks, and provide the convergence analysis of GNN with fully-connected layers which may be of independent interest. In the end, we conduct extensive experiments against state-of-the-art baselines on multiple public data sets, showing the effectiveness of the proposed algorithm.

CCS CONCEPTS
• Information systems → Personalization; • Theory of computation → Online learning algorithms.

KEYWORDS
Contextual Bandits; Online Learning; Graph Neural Networks

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.

ACM Reference Format:
Yunzhe Qi, Yikun Ban, and Jingrui He. 2022. Neural Bandit with Arm Group Graph. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’22), August 14–18, 2022, Washington, DC, USA. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3534678.3539312

1 INTRODUCTION
Contextual bandits are a specific type of multi-armed bandit (MAB) problem where the learner has access to the contextual information (contexts) related to arms at each round, and the learner is required to make recommendations based on past contexts and received rewards. A variety of models and algorithms have been proposed and successfully applied on real-world problems, such as online content and advertising recommendation [28, 40], clinical trials [15, 36] and virtual support agents [31].

In this paper, we focus on exploiting the accessible arm information to improve the performance of bandit algorithms. Among different types of contextual bandit algorithms, upper confidence bound (UCB) algorithms have been proposed to balance between exploitation and exploration [3, 11, 34]. For conventional UCB algorithms, they are either under the “pooling setting” [11] where one single UCB model is applied for all candidate arms, or the “disjoint setting” [28] where each arm is given its own estimator without the collaboration across different arms. Both settings have their limitations: applying only one single model may lead to unanticipated estimation error when some arms exhibit distinct behaviors [40, 41]; the other hand, assigning each arm its own estimator neglects the mutual impacts among arms and usually suffers from limited user feedback [5, 18].

To deal with this challenge, adaptively assigning UCB models to arms based on their group information can be an ideal strategy, i.e., each group of arms has one estimator to represent its behavior. This modeling strategy is linked to “arm groups” existing in real-world applications. For example, regarding the online movie recommendation scenario, the movies (arms) with the same genre can be assigned to one (arm) group. Another scenario is the drug development, where given a new cancer treatment and a patient pool, we need to select the best patient on whom the treatment is most effective. Here, the patients are the arms, and they can be naturally grouped by their non-numerical attributes, such as the cancer types. Such group information is easily accessible, and can significantly improve the performance of bandit algorithms. Although some works [12, 30] have been proposed to leverage the arm correlations, they can suffer from two common limitations. First, they rely on the assumption of parametric (linear / kernel-based) reward functions, which may not hold in real-world applications [46]. Second, they both neglect the correlations among arm groups. We emphasize that the correlations among arm groups also play indispensable roles in many decision-making scenarios. For instance, in online movie recommendation, with each genre being a group of movies, the users who like “adventure movies” may also appreciate “action movies”. Regarding drug development, since the alternation of some genes may lead to multiple kinds of tumors [32], different types of cancer can also be correlated to some extent.

To address these limitations, we first introduce a novel model, AGG (Arm Group Graph), to formulate non-linear reward assumptions and arm groups with correlations. In this model, as arm attributes are easily accessible (e.g., movie’s genres and patient’s cancer types), the arms with the same attribute are assigned into one group, and represented as one node in the graph. The weighted edge between two nodes represents the correlation between these...
two groups. In this paper, we assume the arms from the same group are drawn from one unknown distribution. This also provides us with an opportunity to model the correlation of two arm groups by modeling the statistical distance between their associated distributions. Meantime, the unknown non-parametric reward mapping function can be either linear or non-linear.

Then, with the arm group graph, we propose the AGG-UCB framework for contextual bandits. It applies graph neural networks (GNNs) to learn the representations of arm groups with correlations, and neural networks to estimate the reward functions (exploitation). In particular, with the collaboration across arm groups, each arm will be assigned with the group-aware arm representation learned by GNN, which will be fed into a fully-connected (FC) network for the estimation of arm rewards. To deal with the exploitation-exploration dilemma, we also derive a new upper confidence bound based on network-gradients for exploration. By leveraging the arm group information and modeling arm group correlations, our proposed framework provides a novel arm selection strategy for dealing with the aforementioned challenges and limitations. Our main contributions can be summarized as follows:

- First, motivated by real-world applications, we introduce a new graph-based model in contextual bandits to leverage the available group information of arms and exploit the potential correlations among arm groups.
- Second, we propose a novel UCB-based neural framework called AGG-UCB for the graph-based model. To exploit the relationship of arm groups, AGG-UCB estimates the arm group graph with received contexts on the fly, and utilizes GNN to learn group-aware arm representations.
- Third, we prove that AGG-UCB can achieve a near-optimal regret bound in the over-parameterized neural works, and provide convergence analysis of GNN with fully-connected layers, which may be of independent interest.
- Finally, we conduct experiments on publicly available real data sets, and demonstrate that our framework outperforms state-of-the-art techniques. Additional studies are conducted to understand the properties of the proposed framework.

The rest of this paper is organized as following. In Section 2, we briefly review related works. Section 3 introduces the new problem settings, and details of our proposed framework AGG-UCB will be presented in Section 4. Then, we provide theoretical analysis for AGG-UCB in Section 5. After presenting experimental results in Section 6, we finally conclude the paper in Section 7. Due to the page limit, readers may refer to our arXiv version of the paper for the supplementary contents (https://arxiv.org/abs/2206.03644).

2 RELATED WORKS

In this section, we briefly review the related work on contextual bandits. Lin-UCB [11] first formulates the reward estimation through a linear regression with the received context and builds a confidence bound accordingly. Kernel-UCB [34] further extends the reward mapping to the Reproducing Kernel Hilbert Space (RKHS) for the reward and confidence bound estimation under non-linear settings. Besides, there are algorithms under the non-linear settings. Similarly, CGP-UCB [27] models the reward function through a Gaussian process. GCN-UCB [33] applies the GNN model to learn each context an embedding for the linear regression. Then, Neural-UCB [46] proposes to apply FC neural network for reward estimations and derive a confidence bound with the network gradient, which is proved to be a success, and similar ideas has been applied to some other models [4, 6, 45]. [8] assigns another FC neural network to learn the confidence ellipsoid for exploration. Yet, as these works consider no collaboration among estimators, they may suffer from the performance bottleneck in the introduction.

To collaborate with different estimators for contextual bandits, various approaches are proposed from different perspectives. User clustering algorithms [5, 7, 19, 29] try to cluster user with alike preferences into user groups for information sharing while COFIBA [30] additionally models the relationship of arms. Then, KMTL-UCB [12] extends Kernel-UCB to multi-task learning settings for a refined reward and confidence bound estimation. However, these works may encounter with performance bottlenecks as they incline to make additional assumptions on the reward mapping by applying parametric models and neglect the available arm group information.

GNNs [17, 25, 26, 39] are a kind of neural models that deal with tasks on graphs, such as community detection [44], recommender systems [38] and modeling protein interactions [16]. GNNs can learn from the topological graph structure information and the input graph signal simultaneously, which enables AGG-UCB to cooperate with different arm groups by sharing information over the arm group neighborhood.

3 PROBLEM DEFINITION AND NOTATION

We suppose a fixed pool \( C = \{1, \ldots, N_c\} \) for arm groups with the number of arm groups being \(|C| = N_c\), and assume each arm group \( c \in C \) is associated with an unknown fixed context distribution \( D_c \).

At each time step \( t \), we will receive a subset of groups \( C_t \subseteq C \). For each group \( c \in C_t \), we will have the set of sampled arms \( X_{c,t} = \{x_{c,t}^{(1)}, \ldots, x_{c,t}^{(n_{c,t})}\} \) with the size of \(|X_{c,t}| = n_{c,t}\). Then, \( V_t \in [n_{c,t}] = \{1, \ldots, n_{c,t}\} \), we suppose \( x_{c,t}^{(i)} \sim D_c \) with the dimensionality \( x_{c,t}^{(i)} \in \mathbb{R}^{d_c} \). Therefore, in the \( t \)-th round, we receive

\[
\{X_{c,t}|c \in C_t\} \text{ and } X_{c,t} = \{x_{c,t}^{(1)}, \ldots, x_{c,t}^{(n_{c,t})}\}, \forall c \in C_t. 
\]

With \( W^* \in \mathbb{R}^{N_c \times N_c} \) being the unknown affinity matrix encoding the true arm group correlations, the true reward \( r_{c,t}^{(i)} \) for arm \( x_{c,t}^{(i)} \) is defined as

\[
r_{c,t}^{(i)} = h(W^*, x_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)} \tag{2}
\]

where \( h(\cdot) \) represents the unknown reward mapping function, and \( \epsilon \) is the zero-mean Gaussian noise. For brevity, let \( x_t \) be the arm we select in round \( t \) and \( r_t \) be the corresponding received reward.

Our goal is recommending arm \( x_t \) (with reward \( r_t \)) at each time step to minimize the cumulative pseudo-regret \( R(T) = \sum_{t=1}^{T} \mathbb{E}[(r^*_t - r_t)] \), where \( \mathbb{E}[r^*_t] = \max_{c \in C_t, i \in [n_{c,t}]} h(W^*, x_{c,t}^{(i)}) \). At each time step \( t \), the overall set of received contexts is defined as \( X_t = \{x_{c,t}^{(i)}: c \in C_t, i \in [n_{c,t}]\} \). Note that one arm is possibly associated with multiple arm groups, such as a movie with multiple genres. In other words, for some \( c, c' \in C_t \), we may have \( X_{c,t} \cap X_{c',t} \neq \emptyset \).

In order to model the arm group correlations, we maintain an undirected graph \( G \). At each time step \( t \), each arm group from \( C_t \) is mapped to a corresponding node in node set \( V \).
ALGORITHM 1: AGG-UCB

1. **Input**: Number of rounds $T$, exploration parameter $\gamma$, regularization parameter $\lambda$, network width $m$, network depth $L$, neighborhood size $k$.

2. **Output**: Arm recommendation $x_t$ for each time step $t$.

3. **Initialization**: Initialize the arm group graph as a connected graph $G_t = (V, E, W_t)$. Initialize gradient matrix $Z_0 = L$. Initialize parameter $\Theta_0$ for the model $f(G, X; \Theta_0)$.

4. **for** $t = 1, 2, ..., T$ **do**
   
   5. Receive a set of input arm contexts $X_t = \{x^{(i)}_{t,c} \mid c \in C, i \in \{n_{c,t}\}\}$.
   
   6. Embed the arm set $X_t$ into $\tilde{X}_t$ w.r.t. Eq. 4.

   7. **for** each embedded arm $\tilde{x}^{(j)}_{c,t} \in \tilde{X}_t$ **do**

   8. Obtain the point estimate $\hat{r}^{(j)}_{c,t} = f(G_t, \tilde{x}^{(j)}_{c,t}; \Theta_t)$.

   9. Obtain network gradient $g^{(j)}_{c,t} = \nabla_{\Theta_t} f(G_t, \tilde{x}^{(j)}_{c,t}; \Theta_t)$.

10. Calculate confidence bound as $\hat{\gamma}^{(j)}_{c,t} = \sqrt{g^{(j)}_{c,t} \cdot Z_{t-1} g^{(j)}_{c,t} / m}$. 

11. **end**

12. Recommend $\tilde{x}_t = \text{argmax}_{\tilde{x}^{(j)}_{c,t} \in \tilde{X}_t} (\hat{r}^{(j)}_{c,t} + \gamma^{(j)}_{c,t})$ with the received reward represented as $r_t$.

13. Calculate arm group distances w.r.t. Eq. 3, and update the arm group graph $G_t$ to $G_{t+1}$.

14. Update the model parameter $\Theta_{t+1}$ to $\Theta_t$ according to Algorithm 2.

15. Retrieve the $\tilde{x}_t$’s gradient vector $g_t$, and update gradient matrix $Z_t = Z_{t-1} + g_t \cdot g_t^T$.

16. **end**

Then, $E = \{e(c, c')\}_{c, c' \in C}$ is the set of edges, and $W_t$ represents the set of edge weights. Note that by definition, $G_t$ will stay as a fully-connected graph, and the estimated arm group correlations are modeled by the edge weights connecting pairs of nodes. For a node $v \in V$, we denote the augmented $k$-hop neighborhood $\tilde{N}_k(v) = N_k(v) \cup \{v\}$ as the union node set of its $k$-hop neighborhood $N_k(v)$ and node $v$ itself.

4.2 PROPOSED AGG-UCB FRAMEWORK

In this section, we start with an overview to introduce our proposed AGG-UCB framework. Then, we will show our estimation method of arm group graph before mentioning the related group-aware arm embedding. Afterwards, the two components of our proposed framework, namely the aggregation module and the reward-estimation module, will be presented.

4.1 Overview of AGG-UCB Framework

In Algorithm 1, we present the pseudo-code of proposed AGG-UCB framework. At each time step $t \in [T]$, AGG-UCB would receive a set of input arm contexts $X_t = \{x^{(i)}_{c,t} \mid c \in C, i \in \{n_{c,t}\}\}$ (line 5).
4.3 Group-Aware Arm Embedding

To conduct the aggregation of GNN, we reconstruct a matrix for each arm context vector. Recall that for an arm group \( c \in C \), if \( c \in C_j \), we receive the contexts \( x_{c,t}^{(i)} \in \mathbb{R}^{d_c} \), \( i \in [n_c,t] \) at time step \( t \). Then, the reconstructed matrix for \( x_{c,t}^{(i)} \) is defined as

\[
\tilde{X}_{c,t}^{(i)} = \begin{pmatrix}
(x_{c,t}^{(i)})^\top & 0 & \cdots & 0 \\
0 & (x_{c,t}^{(i)})^\top & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & (x_{c,t}^{(i)})^\top
\end{pmatrix} \in \mathbb{R}^{N_c \times d_c}
\]  

(4)

where \( d_c = d_x \cdot N_c \) is the column dimension of \( \tilde{X}_{c,t}^{(i)} \). Here, for the \( c \)'-th row in matrix \( \tilde{X}_{c,t}^{(i)} \), the \((c-1) \cdot d_x + 1 \)-th to the \((c \cdot d_x \cdot d_c)\)-th entries are the transposed original arm context \((x_{c,t}^{(i)})^\top \), while the other entries are zeros. Receiving a set of arm contexts \( \tilde{X}_t \), we derive the corresponding embedded arm set as \( \hat{X}_c = (\tilde{X}_{c,t}^{(i)})_{c \in C_t, i \in [n_c,t]} \).

4.3.1 Aggregation of arm group representations. To leverage the estimated arm group graph for downstream reward estimations, we propose to aggregate over the arm group neighborhood for a more comprehensive arm representation through the GNN-based module, named as group-aware arm representation. It has been proven that the local averaging operation on the graph neighborhood can be deemed as applying the low-pass filter on the corresponding node features [23, 39], which would give locally smooth node features within the same neighborhood. Inspired by the SGC model [39], we propose to aggregate over the \( k \)-hop arm group neighborhood \( \tilde{N}_{k}(\cdot) \) for incorporating arm group correlations to obtain the aggregated group-aware embedding for an embedded arm \( \hat{X}_{c,t}^{(i)} \), denoted by

\[
H_{gnn} = \frac{1}{m} \cdot \sigma(S_l \cdot \tilde{X}_{c,t}^{(i)} \Theta_{gnn}) \in \mathbb{R}^{N_c \times m}
\]  

(5)

where \( S_l = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \) is the symmetrically normalized adjacency matrix, and we have

\[
\Theta_{gnn} = \begin{pmatrix}
\Theta_{gnn}^1 \in \mathbb{R}^{d_x \times m} \\
\vdots \\
\Theta_{gnn}^c \in \mathbb{R}^{d_x \times m} \\
\vdots \\
\Theta_{gnn}^{N_c} \in \mathbb{R}^{d_x \times m}
\end{pmatrix}
\in \mathbb{R}^{d_x \times m}.
\]

being the trainable weight matrix with width \( m \). Here, \( \sigma(\cdot) \) denotes the non-linear activation function, which is added after the aggregation operation to alleviate potential concerns when the contexts are not linearly separable [23]. Note that the \( c \)'-th row of \( \hat{X}_{c,t}^{(i)} \cdot \Theta_{gnn} \), denoted by \( \hat{X}_{c,t}^{(i)} \cdot \Theta_{gnn} \), is the hidden representation of arm \( x \) in terms of \( c \)'-th arm group in \( C \). Then, these hidden representations will then be aggregated over \( \hat{N}_c(\cdot), c \in C \) by multiplying with \( S_l^T \) to derive the aggregated arm representation for \( x \), i.e., \( H_{gnn}(x) \).

4.3.2 Incorporating initial embedded contexts. Moreover, solely aggregating information from neighbors through the GNN-based models can lead to “over-smoothing” problems [42, 43]. Aggregating from the node neighborhood will end up with identical representations for all the nodes if they form an isolated complete sub-graph, which may not correctly reflect the relationship among these nodes in real-world applications. Therefore, we propose to apply skip-connections to address this potential problem by combining the initial contexts with the aggregated hidden features. Similar ideas have been applied to boost the performance of neural models. For instance, JK-Net [42] and GraphSAGE [20] concatenate hidden features from different levels of node neighborhoods; and ResNet [21] adopts additive residual connections.

Putting these two parts together and setting \( d' = m + d_c \), we then have \( H_0 \in \mathbb{R}^{N_c \times d'} \) as the output group-aware arm representation for \( \hat{X}_{c,t}^{(i)} \), represented by

\[
H_0 = f_{gnn}(G_t, \hat{X}_{c,t}^{(i)} \Theta_{gnn}) = [\sigma(S_l \cdot \hat{X}_{c,t}^{(i)} \Theta_{gnn}); \hat{X}_{c,t}^{(i)}] \in \mathbb{R}^{N_c \times d'}
\]

(6)

where \([\cdot; \cdot] \) refers to the column-wise concatenation of matrices.

4.4 Reward Estimation Module

In this subsection, we estimate the rewards with a FC network of \( L \) layers and width \( m \), based on group-aware arm representation \( H_0 \).

4.4.1 Reward and confidence bound estimation. Here, let \( \Theta_{fc} = \{\Theta_i\}_{i \in [L]} \) be the set of trainable weight matrices of a fully-connected network, where the specifications are: \( \Theta_i \in \mathbb{R}^{d_x \times m}, \Theta_i \in \mathbb{R}^{m} \) and \( \Theta_i \in \mathbb{R}^{m \times m}, \forall i \in \{2, \ldots, L-1\} \). Then, given the group-aware representation \( H_0 \), we have the reward estimation module as follows

\[
H_l = \sqrt{\frac{1}{m} \cdot \sigma(H_{l-1} \cdot \Theta)}, \quad l \in \{1, \ldots, L-1\}
\]

(7)

\[
\hat{r}_{all} = f_{fc}(H_0: \Theta_{fc}) = \sqrt{\frac{1}{m} \cdot H_{L-1} \cdot \Theta_L}
\]

where \( \hat{r}_{all} \in \mathbb{R}^{N_c} \) represents the point-estimation vector for the received contexts embedding \( H_0 \) with respect to all the arms groups. Given that the arm \( \hat{X}_{c,t}^{(i)} \) belonging to \( c \)-th group, we will then have the reward estimation \( \hat{r}_{c,t}^{(i)} = [\hat{r}_{all}]_c \in \mathbb{R} \) for the embedded context matrix \( \hat{X}_{c,t}^{(i)} \), which is the \( c \)-th element of \( \hat{r}_{all} \).

Finally, combining the aggregation module with the reward estimation module, given arm group graph \( G_t \) at time step \( t \), the reward estimation for the embedded arm \( \hat{X}_{c,t}^{(i)} \) (i.e., the reward estimation given its arm group) can be represented as

\[
r_{c,t}^{(i)} = f(G_t, \hat{X}_{c,t}^{(i)}; \Theta) = [f_{fc}(\cdot; \Theta_{fc}) \circ f_{gnn}(\cdot; \Theta_{gnn})](G_t, \hat{X}_{c,t}^{(i)})_c
\]

(8)

Setting \( p = (2N_c \cdot d + (L-1) \cdot m^2 + m) \), we have \( \Theta \in \mathbb{R}^{p} \) being the set of all the parameters from these two modules.

4.4.2 Arm pulling mechanism. We obtain confidence bounds for the point estimation with the network gradients as \( \hat{r} = \sqrt{g^\top Z_{t-1} \cdot g/m} \) where \( g = \nabla_{\Theta} f(G_t, \hat{X}_{c,t}^{(i)}; \Theta) \in \mathbb{R}^{p} \) is the gradient vector, and \( Z_{t-1} = I + \sum_{\tau=1}^{t-1} g_{\tau} \cdot g_{\tau}^\top \) with \( g_{\tau} \) being the gradient vector of the embedded arm which is selected at step \( \tau \in \{1, \ldots, t-1\} \). After obtaining the reward and confidence bound estimations for all embedded arm in set \( \hat{X}_t \), we choose the best arm as \( \hat{x}_t = \arg\max_{x \in \hat{X}_t} \hat{r}_{c,t}^{(i)} + y \cdot \hat{q}_{c,t}^{(i)} \) where \( y \) is the exploration parameter, and the theoretical upper confidence bound will be given in
Section 5. Note that based on our problem definition (Section 3), one arm may associate with multiple arm groups. Here, we will separately estimate rewards and confidence bounds of each arm group it belongs to, and consider them as different arms for selection.

4.5 Model Initialization

For the aggregation module weight matrix $\Theta_{gmn}$, each of its entries is sampled from the Gaussian distribution $N(0, 1)$. Similarly, the parameters from the first $L-1$ reward estimation module layers ([$\Theta_1, \ldots, \Theta_{L-1}$]) are also sampled from $N(0, 1)$. For the final (L-th) layer, its weight matrix $\Theta_L$ is initialized by drawing the entry values from the Gaussian distribution $N(0, 1/m)$.

5 THEORETICAL ANALYSIS

In this section, we provide the theoretical analysis for our proposed framework. For the sake of analysis, at each time step $t$, we assume each arm group $c \in C$ would receive one arm $x_{ct}$, which makes $|X'_t| = \cdots = |X'_{Nc}| = t$. We also apply the adjacency matrix $A_t$ instead of $S_t$ for aggregation, and set its elements $|A_t|_{ij} = \frac{1}{Nc} \sum_{c=1}^{Nc} \phi_{kc}(x_{ci})^T \phi_{kc}(x_{cj})$ for arm group similarity between group $c, c_j \in C$. Here, $\phi_{kc}()$ is the kernel mapping given an RBF kernel $k_g$. With $g^*$ being the unknown true arm group graph, its adjacency matrix elements are $|A^*|_{ij} = \frac{1}{Nc} \sum_{c=1}^{Nc} \phi_{kc}(x_i)^T \phi_{kc}(x_j)$. Note that the norm of adjacency matrices $\|A^*\|_2, \|A_t\|_2$ satisfying $\|x\|_2 = 1$ and its embedded context matrix $\tilde{X}$, there exists $\Theta^*_{t-1} \in \mathbb{R}^p$ at time step $t$, and a constant $S > 0$, such that

$$h(g^*, \tilde{X}) = \langle g(g^*, \tilde{X}; \Theta_{t-1}), \Theta^*_{t-1} - \Theta_0 \rangle$$

where $\|\Theta^*_{t-1} - \Theta_0\|_2 \leq S/\sqrt{m}, \forall t \in [T]$, and $g^*$ stands for the unknown true underlying arm group graph.

Note that with sufficient network width $m$, we will have $\Theta^*_{t-1} \approx \Theta_0^*$, $\forall t \in [T]$, and we will include more details in the full version of the paper. Following the analogous ideas from previous works [6, 46], this lemma formulates the expected reward as a linear function parameterized by the difference between randomly initialized network parameter $\Theta_0$ and the parameter $\Theta^*_{t-1}$, which lies in the confidence set with the high probability [1]. Then, regarding the activation function $\sigma()$, we have the following assumption on its continuity and smoothness.

Assumption 5.2 ($\zeta$-Lipschitz continuity and Smoothness [4, 13]). For non-linear activation function $\sigma()$, there exists a positive constant $\zeta > 0$, such that $\forall x, x' \in \mathbb{R}$, we have

$$|\sigma(x) - \sigma(x')| \leq \zeta \cdot \|x - x'\|$$

$$|\sigma'(x) - \sigma'(x')| \leq \zeta \cdot \|x - x'\|$$

with $\sigma'(\cdot)$ being the derivative of activation function $\sigma()$.

Note that Assumption 5.2 is mild and applicable on many activation functions, such as Sigmoid. Then, we proceed to bound the regret for a single time step $t$.

5.1 Upper Confidence Bound

Recall that at time step $t$, given an embedded arm matrix $\tilde{X}$, the output of our proposed framework is $f = f(G_t, \tilde{X}; \Theta_{t-1})$ with $G_t, \Theta_{t-1}$ as the estimated arm group graph and trained parameters respectively. The true function $h(g^*, \tilde{X})$ is given in Lemma 5.1. Supposing there exists the true arm group graph $g^*$, the confidence bound for a single round $t$ will be

$$C_B(t) = |f(G_t, \tilde{X}; \Theta_{t-1}) - h(g^*, \tilde{X})|$$

$$\leq |f(G_t, \tilde{X}; \Theta_{t-1}) - h(G_t, \tilde{X})| + |h(G_t, \tilde{X}) - h(g^*, \tilde{X})|$$

where $R_1$ denotes the error induced by network parameter estimations, and $R_2$ refers to the error from arm group graph estimations. We will then proceed to bound them separately.

5.1.1 Bounding network parameter error $R_1$. For simplicity, the $G_t$ notation is omitted for this subsection. To bridge the network parameters after GD with those at random initialization, we define the gradient-based regression estimator $\hat{\Theta}_t = Z_t^* - b_t$ where $Z_t = \mathcal{L} + \frac{1}{m} \sum_{r=1}^{t} g(x_{r}, \Theta_{r}) \cdot g(x_{r}, \Theta_{r})^T$, $b_t = \sum_{r=1}^{t} g(x_{r}, \Theta_{r})/\sqrt{m}$. Then, we derive the bound for $R_1$ with the following lemma.

Lemma 5.3. Assume there are constants $\beta > 0, 1 < \beta_1, \beta_2, \beta_3, \beta_4 < 2$, $\beta_5 = \max\{\beta_1, \beta_2, \beta_3, \beta_4\}$, and

$$\beta_6 = \max\{\beta_1, \beta_2 + \beta_3, \beta_1 + 1, (\beta_4)^{L-2}(\beta_2^2 + \beta_3^2)\}.$$

With a constant $\delta \in (0, 1)$, and $L$ as the layer number for the FC network, let the network width $m \geq Poly(t, L, \frac{1}{\beta_4}, \frac{1}{\beta_6}, log(1/\delta))$, and learning rate $\eta \leq O((t \cdot L \beta_8^2 \sqrt{2\beta_4 L^2})^{-1})$. Denoting the terms

$$\tilde{Y} = \frac{2\sqrt{t}}{\beta_8} (\beta_6 + \Lambda)^4 (\beta_6 + 1)^L - \Lambda \cdot \Lambda \cdot \frac{2\beta_6/L - 1}{m}$$

$$\tilde{T}_1 = \sqrt{t \cdot L \cdot \beta_6^4} (\beta_6 + 1)^L + \Lambda \cdot \Lambda \cdot \frac{2\beta_6/L - 1}{m}$$

$$\tilde{T}_2 = \Lambda \cdot \Lambda \cdot \sqrt{L + \tilde{T}_1}$$

at time step $t$, given the received contexts and rewards, with probability at least $1 - \delta$ and the embedded arm group $\tilde{X}$, we have

$$|h(\tilde{X}) - f(\tilde{X}; \Theta_{t-1})| \leq B_1(|g(\tilde{X}; \Theta_{t-1})| / \sqrt{m}Z_{t-1}^2 + B_2 + B_3$$

with the terms

$$B_1 = \sqrt{\log\left(\frac{\det(Z_{t-1})}{\det(\Lambda I)}\right) - 2 \log(\delta) + \Lambda \cdot \Lambda \cdot \frac{S}{m}}$$

$$B_2 = \left(\frac{\tilde{T}_1}{m^2} + \frac{\tilde{T}_2}{m^2} + \frac{\tilde{T}_3}{m^2}\right)^{\frac{1}{2}}$$

$$B_3 = m^{-0.5}(\beta_5 + \beta_6) + L \cdot \Lambda \cdot (\Lambda + \beta_6) + 1)$$

Proof. Given the embedded context $\tilde{X}$, and following the statement in Lemma 5.1, we have

$$|h(\tilde{X}) - f(\tilde{X}; \Theta_{t-1})|$$

$$\leq |g(\tilde{X}; \Theta_{t-1}) / \sqrt{m}, \sqrt{m}Z_{t-1}^2 - \Theta_0| + |g(\tilde{X}; \Theta_{t-1}) / \sqrt{m}, \hat{\Theta}_{t-1}|$$

$$+ |g(\tilde{X}; \Theta_{t-1}) / \sqrt{m}, \Theta_{t-1}| - f(\tilde{X}; \Theta_{t-1})| = B_3 + B_4.$$
With Theorem 2 from [1], we have $R_3 \leq B_1|g(\widehat{X}, \Theta_{t-1})/\sqrt{m}|Z_{t-1}^2$. Then, for $R_4$, we have $|f(\widehat{X}, \Theta_{t-1}) - (g(\widehat{X}, \Theta_{t-1})/\sqrt{m}, \Theta_{t-1})|$

$$\leq R_5 + R_6 = [f(\widehat{X}, \Theta_{t-1}) - (g(\widehat{X}, \Theta_{t-1}), \Theta_{t-1} - \Theta_0)]$$

$$+ |(g(\widehat{X}, \Theta_{t-1}), \Theta_{t-1} - \Theta_0 - \widehat{\Theta}_{t-1}/\sqrt{m})|$$

where $R_5$ can be bounded by $B_3$ with Lemma A.6. Then, with conclusions from Lemma B.3 and Lemma A.5, we have

$$R_6 \leq \|\Theta_{t-1} - \Theta_0 - \widehat{\Theta}_{t-1}/\sqrt{m}\|_2 \cdot |g(\widehat{X}, \Theta_{t-1})|_2$$

$$\leq B_2 = \{(\tilde{T}_1 \cdot \sqrt{V(B_3 + m \cdot \tilde{T}_2)}(m \lambda) + \sqrt{T}/(m \lambda)) \cdot (\Lambda \cdot \sqrt[6]{4L + m^{-1}} + m^{-1}) \beta \cdot \sqrt{L \cdot \beta^2_3 \cdot (\beta_2 \zeta)^{2t} + m},$$

which completes the proof. 

5.1.2 Bounding graph estimation error $R_2$. Regarding the regret term $R_2$ and for the aggregation module, we have

$$H_{nn} = \frac{1}{m} \cdot \sigma(A^i_1 \cdot \bar{X}_\Theta_{nn}) \in \mathbb{R}^{N_i \times m}$$

as the output where $\Theta_{nn}$ refers to the trainable weight matrix. Then, we use the following lemma to bound $R_2$.

**Lemma 5.4. At this time step $t + 1$, given any two arm groups $c_i, c_j \in C$ and their sampled arm contexts $A^i_{t+1} = \{x_{c_i, r}^t \}_{r=1}^\tau, A^j_{t+1} = \{x_{c_j, r}^t \}_{r=1}^\tau$, with the notation from Lemma 5.3 and the probability at least $1 - \delta$, we have**

$$||A^* - A_t||_{max} \leq \frac{1}{N_n} \cdot \sqrt{1 + \frac{1}{2t}} \cdot \frac{\text{log}(N^2_n - N_i)}{\delta} ||\frac{\text{det}(I - \lambda S)}{\lambda S}|| + \frac{1}{\beta} \cdot \frac{\beta \zeta}{\lambda S}$$

where $||\cdot||_{max}$ refers to the greatest entry of a matrix. Then, we will have $R_2 \leq B_4 \sqrt{T}/t$ with

$$B_4 = \frac{S \sqrt{T} \cdot \beta \zeta}{\sqrt{m}} \cdot (\beta \zeta \cdot \frac{Y_m}{m})^{O(1)} \cdot \left(\frac{1}{\sqrt{T}} \cdot \text{log}(N^2_n - N_i) \right) \cdot \frac{1}{\beta} \cdot \frac{\beta \zeta}{\lambda S}$$

and $N_n = |C|$ is the number of arm groups.

**Proof.** Recall that for $c_i, c_j \in C$, the element of matrix $[A^*_i]_{ij} = \frac{1}{N_n} \sum_{r=1}^{\tau} \sum_{x_{c_i, e}^t \in \mathcal{D}_{c_i}} (\phi_{x_{c_i}}(x_{c_i}^t) \Phi_{x_{c_j}}(x_{c_j}^t)), i, j \in [N_n], \Gamma A_{1, i} = 1/N_n \cdot \sum_{x_{c_i, e}^t \in \mathcal{D}_{c_i}} (\phi_{x_{c_i}}(x_{c_i}^t) \Phi_{x_{c_j}}(x_{c_j}^t))$. Here, suppose a distribution $D_{ij}$ where $E[D_{ij}] = \frac{1}{N_n} \sum_{x_{c_i, e}^t \in \mathcal{D}_{c_i}} (\phi_{x_{c_i}}(x_{c_i}^t) \Phi_{x_{c_j}}(x_{c_j}^t))$. Given $N_n$ arm groups, we have $N_n(N_n - 1)/2$ different group pairs. For group pair $c_i, c_j \in C$, each $\phi_{x_{c_i}}(x_{c_i}^t) \Phi_{x_{c_j}}(x_{c_j}^t)$ is a sample drawn from $D_{ij}$, and the element distance $||[A^*_i]_{ij} - [A^*_i]_{ij}||$ can be regarded as the difference between the mean value of samples and the expectation. Applying the Hoeffding’s inequality and the union bound would complete the proof. As $||\cdot||_2 \leq n ||\cdot||_{max}$ for an $n \times n$ square matrix, we have the bound for matrix differences.

Then, consider the power of adjacency matrix $A^T$ (for graph $G$) as input and fix $X$. Analogous to the idea that the activation function with the Lipschitz continuity and smoothness property will lead to Lipschitz neural networks [2], applying Assumption 5.2 and with Lemma A.2, Lemma A.3, we simply have the gradient $g(G, X, \Theta_{t-1})$ being Lipschitz continuous w.r.t. the input graph as

$$R_2 \leq \|g(G, X, \Theta_{t-1}) - g(G_t, X_t, \Theta_{t-1})\|_2 \cdot \|\Theta_{t-1} - \Theta_0\|_2$$

$$\leq \frac{S \sqrt{T} \cdot \beta_2 \zeta}{\sqrt{m}} \cdot (\beta \zeta \cdot \frac{Y_m}{m})^{O(1)} \cdot \left(\frac{1}{\sqrt{T}} \cdot \text{log}(N^2_n - N_i) \right) \cdot \frac{1}{\beta} \cdot \frac{\beta \zeta}{\lambda S}$$

where $(i)$ is because $A_t, A^*$ are symmetric and bounded polynomial functions are Lipschitz continuous. Combining the two parts will lead to the conclusion. 

5.1.3 Combining $R_2$ with $R_1$. At time step $t$, with the notation and conclusions from Lemma 5.3 and Lemma 5.4, re-scaling the constant $\delta$, we have the confidence bound given embedded arm $X$ as

$$CB_t(X) \leq B_1||g(G_t, X, \Theta_{t-1})/\sqrt{m}|Z_{t-1}^2 + B_2 + B_3 + B_4 \sqrt{T}/t \cdot \text{log}(T))$$

5.2 Regret Bound

With the UCB shown in Eq. 10, we provide the following regret upper bound $R(T)$, for a total of $T$ time steps.

**Theorem 5.5.** Given the received contexts and rewards, with the notation from Lemma 5.3, Lemma 5.4, and probability at least $1 - \delta$, if $m, n$ satisfy conditions in Lemma 5.3, we will have the regret $R(T) \leq 2 \cdot (2B_4 \sqrt{T} + B_4) + 2 \sqrt{2dT \log (1 + T/\lambda) + 2T} \cdot (\sqrt{\Lambda S} + \sqrt{1 - 2dT\log(\lambda/2) + (\lambda d \log (1 + T/\lambda)))}$

where the effective dimension $d = \text{log(det}(I + G_0))/\text{log}(\sqrt{T + \delta}))$ with $G_0 = G_0 G_0^T$ and $G_0 = (g(\bar{X}_1; \Theta_0)^T, \ldots, g(\bar{X}_t; \Theta_0)^T)$.

**Proof.** By definition, we have the regret $R_t$ for time step $t$ as

$$R_t = h(G_t, X_t) - h(G^*, X_t)$$

$$\leq \text{CB}_t(X_t) + h(f(G_t, X_t; \Theta_{t-1}) - h(G^*, X_t)$$

$$\leq \text{CB}_t(X_t) + h(f(G_t, X_t; \Theta_{t-1}) - h(G^*, X_t) \leq 2 \cdot \text{CB}_t(X_t)$$

where the second inequality is due to our arm pulling mechanism. Then, based on Lemma 5.4, Lemma 5.3, and Eq. 10, we have $R(T) = \sum_{t=1}^{T} R_t \leq \sum_{t=1}^{T} \left(B_1||g(G_t, X_t, \Theta_{t-1})/\sqrt{m}|Z_{t-1}^2 + B_2 + B_3 + B_4 \sqrt{1/T} \cdot \text{log}(T)) \right)$

$$\leq 2 \cdot (2B_4 \sqrt{T} + B_4) + 2 \sum_{t=1}^{T} \left(B_1||g(G_t, X_t, \Theta_{t-1})/\sqrt{m}|Z_{t-1}^2 \cdot \text{log}(\text{det}(Z_{t})) \text{det}(I)) \right)$$

where $(i)$ is based on Lemma 6.3 in [4] and Lemma 5.4 in [46].
Here, the effective dimension \( \tilde{d} \) measures the vanishing speed of \( G(0) \)'s eigenvalues, and it is analogous to that of existing works on neural contextual bandits algorithms [4, 6, 46]. As \( \tilde{d} \) is smaller than the dimension of the gradient matrix \( G(0) \), it is applied to prevent the dimension explosion. Our result matches the state-of-the-art regret complexity [4, 45, 46] under the worst-case scenario.

5.3 Model Convergence after GD

For model convergence, we first give an assumption of the gradient matrix after \( j \) iterations of GD. First, we define \( G^{(j)}(\Theta_{L-1}) = (g(\tilde{X}_1; \Theta_{(j)}^{(1)}), \ldots, g(\tilde{X}_m; \Theta_{(j)}^{(m)}))^T \) and for \( j \in [J] \), we have the minimal eigenvalue of \( G^{(j)} \) as

\[
\lambda_{\text{min}}(G^{(j)}(\Theta_{L-1})) \geq \lambda_0 / 2
\]

where \( \lambda_0 \) is the minimal eigenvalue of the neural tangent kernel (NTK) [24] matrix induced by AGG-UCB.

Assumption 5.6. With width \( m \geq \text{Poly}(T, L, \frac{1}{\beta_F}, \frac{1}{\lambda_0}, \frac{1}{\beta_F} L^2, \log \left( \frac{1}{\delta} \right)) \) and for \( j \in [J] \), we have the minimal eigenvalue of \( G^{(j)} \) as

\[
\lambda_{\text{min}}(G^{(j)}(\Theta_{L-1})) \geq \lambda_0 / 2
\]

where \( \lambda_0 \) is the minimal eigenvalue of the neural tangent kernel (NTK) [24] matrix induced by AGG-UCB.

Note that Assumption 5.6 is mild and has been proved for various neural architectures in [13]. The NTK for AGG-UCB can be derived following a comparable approach as in [14, 24]. Then, we apply the following lemma and theorem to prove the convergence of AGG-UCB. The proof of Lemma 5.7 is given in the appendix.

Lemma 5.7. After \( T \) time steps, assume the network are trained with the \( J \)-iterations GD on the past contexts and rewards. Then, with \( \beta_F > 0 \) and \( \beta_F \cdot \eta < 1 \), for any \( j \in [J] \),

\[
\|F^{(j)} - Y_T\|_2^2 \leq \frac{4}{\epsilon_2^2 \beta_F^2} \cdot \|F^{(j)} - Y_T\|_2^2
\]

with network width \( m \) defined in Lemma 5.3.

The Lemma 5.7 shows that we are able to bound the difference in network outputs after one step of GD. Then, we proceed to prove the convergence with the theorem below.

Theorem 5.8. After \( T \) time steps, the model with width \( m \) defined in Lemma 5.7 is trained with the \( J \)-iterations GD on the contexts \( \tilde{X}_r \)'s and rewards \( r_r \)'s. For the learning rate \( \eta \) such that \( \beta_F \cdot \eta < 1 \), where \( \lambda_0 \) is the minimal eigenvalue of the neural tangent kernel (NTK) [24] matrix induced by AGG-UCB.

\[ \|F^{(j)} - Y_T\|_2^2 \leq \frac{1}{\epsilon_2^2 \beta_F^2} \cdot \|F^{(j)} - Y_T\|_2^2 \]

where \( \eta \) is the gradient vector w.r.t. \( \Theta_{L-1} \).

Proof. Following an approach analogous to [13], we apply and induction based method for the proof. The hypothesis is that \( \|F^{(j)} - Y_T\|_2^2 \leq \frac{1}{\epsilon_2^2 \beta_F^2} \cdot \|F^{(j)} - Y_T\|_2^2 \) for every \( j \in [J] \). With a similar procedure in Condition A.1 of [13], we have

\[
\|F^{(j+1)} - Y_T\|_2^2 \leq \|F^{(j)} - Y_T\|_2^2 - 2\eta \|F^{(j)} - Y_T\|_2^2 \left( \|G^{(j)}\| + 2\|F^{(j)} - Y_T\|_2^2 \right)^2
\]

where \( \|G^{(j)}\| \) is the gradient vector w.r.t. \( \Theta_{L-1} \).
dimension \( d = 20 \). Then, since the genome-scores of user-specified tags are provided for each movie, we select 20 tags with the highest variance to construct the movie features \( v_t \in \mathbb{R}^d \) with their scores on these tags. Then, these movies are allocated into 19 groups based on their genres (\(|C| = 19\)). Receiving a user \( u_t \) at each time step \( t \), we follow the idea of Generalized Matrix Factorization (GMF) [22, 47, 48] to encode user information into the contexts as \( x_{c,t}^{(i)} = [v_t \odot v_i] \in \mathbb{R}^d, c \in C, i \in [n_{c,t}] \), and let \( |X_t| = 20 \). Finally, we concatenate a constant 0.01 to each \( x_{c,t}^{(i)} \) to obtain \( x_{c,t}^{20} \in \mathbb{R}^{d+1} \), which makes \( d_c = 21 \), before normalizing \( x_{c,t}^{20} \). Rewards \( r_{c,t}^{(i)} \) are user ratings normalized into range \([0, 1]\).

Then, for the Yelp data set (https://www.yelp.com/dataset), we choose 4000 users with most reviews and restaurants from 20 different categories as arms (\(|C| = 20\)). Both user features and arm features are obtained through SVD with the dimension \( d = 20 \). Analogous to the MovieLens data set, we follow the GMF based approach and the fore-mentioned constant concatenation to get the arm context \( x_{c,t}^{20} (d_c = 21, |X_t| = 20) \) to encode the user information, and the rewards are the normalized user ratings.

**MNIST data set with augmented classes (MNIST-Aug).** MNIST is a well-known classification data set with 10 original classes where each sample is labeled as a digit from 0 to 9. Here, we further divide the samples from each class into 5 sub-divisions through \( K \)-means clustering, which gives us a total of 50 augmented sub-classes (i.e., arm groups) for the whole data set. Given a sample \( x_t \), the reward would be \( r_t = 1 \) if the learner accurately predicts its sub-class; or the learner will receive the partial reward \( r_t = 0.5 \) when it chooses the wrong sub-class, but this sub-class and the correct one belong to the same digit (original class). Otherwise, the reward \( r_t = 0 \).

**XRMB data set.** XRMB data set [37] is a multi-view classification data set with 40 different labels. Here, we only apply samples from the first 38 classes as there are insufficient samples for the last two classes. The arm contexts \( x_r \) are the first-view features of the samples. Then, learner will receive a reward of \( r_t = 1 \) when they predict the right label, and \( r_t = 0 \) otherwise.

### 6.2 Experimental Results

Figure 1 shows the cumulative regret results on the two real recommendation data sets where our proposed AGG-UCB outperforms all strong baselines. In particular, we can find that algorithms with group-aware arm embedding tend to perform better than those without the arm group information (Kernel-Pool, Neural-Pool). This confirms the necessity of exploiting arm group information. Nevertheless, these baselines fed with group-aware are outperformed by AGG-UCB, which implies the advantages of our new graph-based model. Meantime, it can be observed that neural algorithms (AGG-UCB, Neural-Ind, Neural-TS) generally perform better compared with other baselines due to the representation power of neural networks. Note that since the user features and arm features of the Yelp data set are directly extracted with SVD, the reward estimation on the Yelp data set is comparably easy compared with others data sets. Therefore, the performances of benchmarks do not differ dramatically with AGG-UCB. In opposite, MovieLens data set with true arm features tends to be a more challenging task where a more complex mapping from arms to their rewards can be involved. This can be reason for AGG-UCB’s superiority over the competitors.

Then, Figure 2 shows the cumulative regret results on the two classification data sets where our AGG-UCB achieves the best performance compared with other baselines. In particular, since sub-classes from each digit are highly correlated in the MNIST-Aug data set, our proposed AGG-UCB tends to perform significantly better due to its ability of leveraging arm group correlations compared with other neural methods. Thus, these two aspects verify our claim that associating the neural models with arm group relationship modeling can lead to better performance.

### 6.3 Parameter Study

In this section, we conduct our parameter study for the neighborhood parameter \( k \) on the MovieLens data set and MNIST-Aug data set with augmented labels, and the results are presented in Figure 3. For the MovieLens data set, we can observe that setting \( k = 1 \) would give the best result. Although increasing \( k \) can enable the aggregation module to propagate the hidden representations for multiple hops, it can potentially fail to focus on local arm group neighbors with high correlations, which is comparable to the aforementioned “over-smoothing” problem. In addition, since the arm group graph of MovieLens data set only has 19 nodes, \( k = 1 \) would be enough.
Meantime, setting $k = 1$ also achieves the best performance on the MNIST data set. The reason can be that the 1-hop neighborhood of each sub-class can already include all the other sub-classes from the same digit with heavy edge weights within the neighborhood for arm group collaboration. Therefore, unless setting $k$ to considerably large values, the AGG-UCB can maintain robust performances, which reduces the workload for hyperparameter tuning.

7 CONCLUSION

In this paper, motivated by real applications where the arm group information is available, we propose a new graph-based model to characterize the relationship among arm groups. Based on this model, we propose a novel UCB-based algorithm named AGG-UCB, which uses GNN to exploit the arm group relationship and share the information across similar arm groups. Compared with existing methods, AGG-UCB provides a new way of collaborating multiple neural contextual bandit estimators for obtaining the rewards. In addition to the theoretical analysis of AGG-UCB, we empirically demonstrate its superiority on real data sets in comparison with state-of-the-art baselines.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, IIS-2137468, and IIS-2002540. 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 Abbasia-Malikdari, David Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. NeurIPS 24 (2011), 2312–2320.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Quanquan Gu. 2019. Generalizing from several contexts. arXiv preprint arXiv:1910.08977 (2019).

[3] Simon Du, Jason Lee, Haochuan Li, and Liwei Wang. 2019. Neural context bandits with linear payoff functions. In NeurIPS. PMLR, 67–82.

[4] Yi-Kai Liu, Rong Ge, and Yanjun Han. 2018. A neural network perspective. arXiv preprint arXiv:1808.00134 (2018).

[5] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In WWW. 2178–2186.

[6] Edward Chlebus. 2009. An approximate formula for a partial sum of the divergent p-series. Applied Mathematics Letters 22, 5 (2009), 732–737.

[7] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In AISTATS. 208–214.

[8] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. 2017. Multi-task learning to characterize the relationship among arm groups. In Proceedings of the Web Conference 2021.

[9] Thomas Sripataporn and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).

[10] Andreas Krause and Cheng Soon Ong. 2011. Contextual Gaussian Process Bandit Optimization. In NeurIPS. 2447–2455.

[11] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Contextual Bandits with Arm Group Graph. KDD ’22, August 14–18, 2022, Washington, DC, USA.
[48] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabad, Evren Körpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In KDD ’21: 2409–2419.
A Lemmas for Intermediate Variables and Weight Matrices

Due to page limit, we will give the proof sketch for lemmas at the end of each corresponding appendix section. Recall that each input context $x_{c_t}^{(i)}$, $i \in [n_{c_t}]$ is embedded to $X_{c_t}^{(i)}$ (represented by $X$ for brevity). Supposing $\tilde{X}$ belongs to the arm group $c$, denote $h_k = [\tilde{A}_k^T \tilde{X}]$, as the corresponding row in matrix $\tilde{A}_k^T \tilde{X}$ based on index of group $c$ in $C$ (if group $c$ is the $c'$-th group in $C$, then $h_k$ is the $c'$-th row in $\tilde{A}_k^T \tilde{X}$). Similarly, we have $h_{gnn} = [H_{gnn}]_c$, and $h_l = [H_l]_c$ respectively. Given received contexts $(\tilde{X}_t)_{t=1}^T$ and rewards $(r_t)_{t=1}^T$, the gradient w.r.t. weight matrix $\Theta$, $\forall l \in \{1, \ldots, L - 1\}$ will be

$$
\frac{\partial L(\Theta)}{\partial \Theta_l} = m \sum_{t=1}^T f(\tilde{X}_t; \Theta) - r_t \langle h_{l-1} \Theta_l \rangle \left( \sum_{q=1}^{L-1} \Gamma_q \Theta_q^T \right) \Gamma_l
$$

where $\Gamma_q = \text{diag}([\sigma'(h_{q-1} \Theta_q)])$ is the diagonal matrix whose entries are the elements from $\sigma'(h_{q-1} \Theta_q)$. The coefficient $\frac{1}{2}$ of the cost function is omitted for simplicity. Then, for $\Theta_{gnn}$, we have

$$
\frac{\partial L(\Theta)}{\partial \Theta_{gnn}} = m \sum_{t=1}^T f(\tilde{X}_t; \Theta) - r_t \langle h_A \Theta_{gnn}^T \rangle \left( \sum_{q=1}^{L-1} \Gamma_q \Theta_q^T \right) \Gamma_{gnn}
$$

where $\Gamma_{gnn} = \text{diag}([\sigma'(h_A \Theta_{gnn})])$, $Q = \left( \begin{array}{c} I \in \mathbb{R}^{m \times m} \\ 0 \in \mathbb{R}^{(d - m) \times m} \end{array} \right) \in \mathbb{R}^{d \times m}$. Given the same $\Gamma_l$, we provide lemmas to bound the $R_1$ term of Eq. 9. For brevity, the subscript $t \in [T]$ and notation $\Gamma_l$ are omitted below by default.

**Lemma A.1.** Given the randomly initialized parameters $\Theta^{(0)} = (\Theta_{gnn}^{(0)}, \Theta_1^{(0)}, \Theta_2^{(0)}, \ldots, \Theta_L^{(0)})$, with the probability at least $1 - O(TL) \cdot e^{-\Omega(m)}$ and constants $1 < \beta_1, \beta_2, \beta_3, \beta_4 < 2$, we have

$$
\|\Theta^{(0)}\|_2 \leq \beta_1 \sqrt{m}, \|\Theta_l^{(0)}\|_2 \leq \beta_2, \|\Theta_{gnn}^{(0)}\|_2 \leq \beta_3, \|\Theta_{l-1}^{(0)}\|_2 \leq \beta_4,
$$

$$
\|f(\tilde{X}; \Theta^{(0)})\|_2 \leq \xi \cdot \beta_1, \|f(\tilde{X}; \Theta^{(0)})\|_2 \leq \xi \cdot \beta_2, \|f(\tilde{X}; \Theta^{(0)})\|_2 \leq \xi \cdot \beta_3, \|f(\tilde{X}; \Theta^{(0)})\|_2 \leq \xi \cdot \beta_4
$$

where $\beta_i \geq \sqrt{\frac{d_c}{m} + 1}$ with $m > d_c$. Applying the analogous approach for the other randomly initialized matrices would give similar bounds. Regarding the nature of $A$, we can easily have $\|A\|_2 \leq 1$. Then,

$$
\|h_{gnn}^{(0)}\|_2 = m^{-\frac{1}{2}} \|\sigma(h_A \cdot \Theta_{gnn})\|_2 \leq \xi m^{-\frac{1}{2}} \cdot \|h_A\|_2 \cdot \|\Theta_{gnn}\|_2 \leq \xi \cdot \beta_1
$$

due to the assumed $\xi$-Lipschitz continuity. Denoting the concatenated input for reward estimation module as $x' = [h_{gnn}^{(0)}; \tilde{X}]_c \in \mathbb{R}^{1 \times (d_c + m)}$, we can easily derive that $\|x'\|_2 \leq \xi \cdot \beta_1 + 1$. Thus,

$$
\|h_1^{(0)}\|_2 = m^{-\frac{1}{2}} \|\sigma(x'; \Theta_1)\|_2 \leq \xi m^{-\frac{1}{2}} \cdot \|x'\|_2 \cdot \|\Theta_1\|_2 \\
\leq \xi \cdot \beta_2 (\xi \cdot \beta_1 + 1) = \xi \cdot \beta_2 + \xi^2 \cdot \beta_1 \beta_2.
$$

Following the same procedure recursively for other intermediate outputs and applying the union bound would complete the proof.

**Lemma A.2.** After $T$ time steps, run GD for $J$-iterations on the network with the received contexts and rewards. Suppose $\|h_{l}^{(j)} - h_l^{(0)}\|_2 \leq \Lambda(L), \forall j \in [J]$. With the probability of at least $1 - O(TL) \cdot e^{-\Omega(m)}$ and $\Theta \in \{\Theta_{gnn}, \Theta_1, \ldots, \Theta_L\}$, we have

$$
\|\Theta^{(j)} - \Theta^{(0)}\|_F \leq \sqrt{m}/\sqrt{\xi}.
$$

**Proof.** We prove this Lemma following an induction-based procedure [4, 13]. The hypothesis is $\|\Theta^{(j)} - \Theta^{(j+1)}\|_F \leq 1/\sqrt{m}, \forall \Theta \in \{\Theta_{gnn}, \Theta_1, \ldots, \Theta_L\}$, and let $\beta_4 = \max(\beta_1, \beta_2, \beta_3, \beta_4)$. According to Algorithm 2, we have for the $j + 1$-th iteration and $l \in [L]$, $\|\Theta_l^{(j+1)} - \Theta_l^{(j)}\|_F = m^{-\frac{1}{2}} \eta \cdot \sum_{t=1}^T \left( \langle h_{l-1} \Theta_l \rangle - r_t \langle h_{l-1} \Theta_l \rangle \right) \cdot \left( \sum_{q=1}^{L-1} \Gamma_q \Theta_q^T \right) \Gamma_l
$

$$
\leq m^{-\frac{1}{2}} \eta \sqrt{m} \cdot \|F_l^{(j)} - Y_l\|_2 \|h_{l-1} \Theta_l^{(j)}\|_F \leq \left( \sum_{q=1}^{L-1} \Gamma_q \Theta_q^T \right) \Gamma_l
$$

$$
\leq m^{-\frac{1}{2}} \eta \sqrt{m} \cdot \|F_l^{(j)} - Y_l\|_2 \|h_{l-1} \Theta_l^{(j)}\|_F \leq \left( \sum_{q=1}^{L-1} \Gamma_q \Theta_q^T \right) \Gamma_l
$$

by Cauchy inequality. For $\|\Theta_q^{(j)}\|_2$, we have

$$
\|\Theta_q^{(j)}\|_2 \leq \left( \sum_{q=1}^{L-1} \langle \Theta_q^{(j)} \rangle_2 \|\Theta_q^{(j)} - \Theta_q^{(0)}\|_2 \right) \leq (\beta_4 \sqrt{m} + \sqrt{m}) \xi L^{-1}.
$$

while for $\|\Theta_l^{(j)}\|_2$, we have $\|\Theta_l^{(j)}\|_2 \leq \sqrt{m} \xi L^{-1} \xi L^{-1}$. Combining all the results above and based on Lemma 5.8, it means that for $l \in [L]$, $\|\Theta_l^{(j+1)} - \Theta_l^{(j)}\|_F \leq m^{-\frac{1}{2}} \eta \sqrt{m} \cdot \|F_l^{(j)} - Y_l\|_2 \|\Theta_l^{(j)}\|_F \leq \sqrt{m} \xi L^{-1} \xi L^{-1}$

$$
\leq m^{-\frac{1}{2}} (\beta_4 \eta) \cdot \|F_l^{(j)} - Y_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2
$$

where the last inequality is due to Lemma A.3. Then, since we have $\|\Theta_l^{(j+1)} - \Theta_l^{(j)}\|_F \leq \|\Theta_l^{(j+1)} - \Theta_l^{(j)}\|_F + \|\Theta_l^{(j)} - \Theta_l^{(0)}\|_F$, it leads to

$$
\|\Theta_l^{(j+1)} - \Theta_l^{(0)}\|_F \leq \frac{2 \sqrt{m}}{\beta_4 \sqrt{m}} \|F_l^{(0)} - Y_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2 \|h_{l-1} \Theta_l\|_2
$$

Neural Bandit with Arm Group Graph

KDD ’22, August 14–18, 2022, Washington, DC, USA
For the last layer $\Theta_L$, the conclusion can be verified through a similar procedure. Analogously, for $\Theta_{g_{nn}}$, we have

$$\|\Theta^{(j+1)}_{g_{nn}} - \Theta^{(j)}_{g_{nn}}\|_F = m^{-\frac{1}{2}} \eta \sum_{t=1}^{T} \|f(\bar{x}_t; \Theta) - r_t\|^2 \cdot \left( h_A \cdot \Theta^2_L \cdot ( L_{q=2}^{l=1} (\Gamma_q \cdot \Theta^1_q) \cdot \Gamma_1^T) \right) \cdot \text{Q} \cdot \Gamma_{g_{nn}} \|_F$$

$$\leq m^{-\frac{1}{2}} \eta \sqrt{T}(1 - \beta_F \cdot \eta)^{1/2} \|F(0) - Y_t\|_2 \cdot \|h_A\|_2 \cdot \prod_{q=1}^{L} \|\Theta_q\|_2^{2\zeta} \cdot \|Q\|_2$$

$$\leq \sqrt{\text{Vim}(1 - \beta_F \cdot \eta)^{1/2} \cdot \|F(0) - Y_t\|_2 \cdot \zeta \cdot \beta_l \cdot (\beta_L + Y/m)^L + \text{L}}$$

which leads to

$$\|\Theta^{(j+1)}_{g_{nn}} - \Theta^{(j)}_{g_{nn}}\|_F \leq \frac{2}{\beta_F} \sqrt{\text{Vim}(1 - \beta_F \cdot \eta)^{1/2} \cdot \|F(0) - Y_t\|_2 \cdot \zeta \cdot (\beta_L + Y/m)^L}.$$  

Since $\|F(0) - Y_t\|_2 \leq \sqrt{2T}$ (Lemma 5.8) and $Y/m \leq 1$ with sufficiently large $m$, combining all the results above would give the conclusion.

**Lemma A.3.** After $T$ time steps, with the probability of at least $1 - O(TL) \cdot e^{-\Omega(m)}$ and running GD of $J$-iterations on the contexts and rewards, we have $\delta^*_j = \max\{\xi, \beta_1, \xi \beta_2 + \xi^2, \beta_2, (\xi^2 \beta_2 + \xi^2), \beta_3, \beta_4\}$ and $\beta_h = \max\{\xi, \beta_1, 1, \beta_h\}$. With $h \in \{h_{g_{nn}}, h_1, \ldots, h_{L-1}\}$, we have

$$\|h^{(j)} - h^{(0)}\|_2 \leq \frac{\sqrt{T}}{m} \cdot \beta_h \cdot \frac{(2\beta_L \beta_1 - 1)}{2\beta_L - 1} = \Lambda^{(j)} \cdot \|h^{(j)}\|_2 \leq \beta_h + \Lambda^{(j)}$$

**Proof.** Similar to the proof of Lemma A.2, we adopt an induction-based approach. For $l \in [L - 1]$, we have

$$\|h_l^{(j)} - h_l^{(0)}\|_2 \leq \frac{\sqrt{T}}{m} \|\sigma(h_{l-1}^{(j)} \cdot \Theta_l^{(j)} - \sigma(h_{l-1}^{(0)} \cdot \Theta_l^{(0)}))\|_2$$

$$\leq \frac{\sqrt{T}}{m} \cdot (\|h_{l-1}^{(0)} \cdot \Theta_l^{(0)} - h_{l-1}^{(0)} \cdot \Theta_l^{(0)}\|_2 + \|h_{l-1}^{(0)} \cdot \Theta_l^{(0)} - h_{l-1}^{(0)} \cdot \Theta_l^{(0)}\|_2)$$

$$\leq \frac{\sqrt{T}}{m} \cdot (\|\Theta_l^{(0)}\|_2 \cdot \|f_l^{(0)}\|_F \cdot \|h_{l-1}^{(0)} - h_{l-1}^{(0)}\|_2 \leq \frac{\sqrt{T}}{m} \cdot (\|h_{l-1}^{(0)} - h_{l-1}^{(0)}\|_2 \cdot \|\Theta_l^{(0)}\|_2 + \|f_l^{(0)}\|_F \cdot \|h_{l-1}^{(0)} - h_{l-1}^{(0)}\|_2 \leq \xi \cdot \beta_L \cdot \|h_{l-1}^{(0)} - h_{l-1}^{(0)}\|_2$$

where the last two inequalities are derived by applying Lemma A.2 and the hypothesis. For the aggregation module output $h_{g_{nn}}^{(0)}$, we have

$$\|h_{g_{nn}}^{(j)} - h_{g_{nn}}^{(0)}\|_2 \leq \frac{\sqrt{T}}{m} \|\sigma(h_{g_{nn}}^{(j)} \cdot h_S) - \sigma(h_{g_{nn}}^{(0)} \cdot h_S)\|_2$$

$$\leq \frac{\sqrt{T}}{m} \|\Theta_{g_{nn}}^{(j)} - \Theta_{g_{nn}}^{(0)}\|_F \cdot \|h_S\|_2 \leq \frac{\sqrt{T}}{m} \cdot \beta_h.$$
To continue the proof, we need to bound the term $\|\nabla_v^{(0)}\|_F$ as

$$\|\nabla_v^{(0)}\|_F = m^{-0.5}\|\Gamma_{\phi_t^0}^{(0)} \cdot \nabla_v^{(0)}\|_F \leq \zeta \beta_L : \|\nabla_v^{(0)}\|_F.$$  

Since for $l = L - 1$ we have

$$\|\nabla_{v_{L-1}}^{(0)}\|_F \leq \frac{\zeta \beta_L}{m},$$

we can derive

$$\|\nabla_v^{(0)}\|_F \leq \frac{\beta_L}{m} \cdot (\zeta \beta_L)^L \leq 1$$

with sufficiently large $m$, and this bound also applies to $\|\nabla_{g_{nn}}^{(0)}\|_F$. For

$$\|\nabla_v^{(0)} - \nabla_v^{(j)}\|_F = m^{-0.5}\|\nabla_{v_{L-1}}^{(0)} - \nabla_{v_{L-1}}^{(j)}\|_F \leq \Lambda(j) \sqrt{m},$$

Therefore, we have

$$\|\nabla_v^{(0)} - \nabla_v^{(j)}\|_F \leq \frac{\zeta \beta_L}{m} \cdot (\zeta \beta_L)^L \leq \frac{\Lambda(j)}{\beta_L}.$$

By following a similar approach as in Lemma A.3, we will have

$$\|\nabla_v^{(0)} - \nabla_v^{(j)}\|_F \leq \frac{\Lambda(j)}{\beta_L} \cdot (2 \beta_L + 1) \leq 3 \cdot \Lambda(j)$$

with sufficiently large $m$. This bound can also be derived for $\|\nabla_{g_{nn}}^{(j)}\|_F$, with a similar procedure. For $l$-th layer, we have

$$\|\nabla_{v_{L-1}}^{(0)} - \nabla_{v_{L-1}}^{(j)}\|_F \leq m^{-0.5}\|\nabla_{v_{L-1}}^{(0)} - \nabla_{v_{L-1}}^{(j)}\|_F \leq \Lambda(j) \sqrt{m},$$

which completes the proof.

**Lemma A.5.** With the probability of at least $1 - O(TL) \cdot e^{-\Omega(m)}$, we have the gradient for all the network as

$$\|g(\tilde{X}; \theta^{(0)})\|_2 \leq m^{-1} \beta_h \cdot \sqrt{L \cdot \beta_L^5 \cdot (\beta_L \zeta)^{2L} + m},$$

$$\|g(\tilde{X}; \theta^{(j)})\|_2 \leq \Lambda(j) \cdot \sqrt{L} + m^{-1} \beta_h \cdot \sqrt{L \cdot \beta_L^5 \cdot (\beta_L \zeta)^{2L} + m},$$

$$\|g(\tilde{X}; \theta^{(0)}) - g(\tilde{X}; \theta^{(j)})\|_2 \leq \Lambda(j) \cdot \sqrt{L} + m^{-1}.$$

**Proof.** First, for the gradient before GD, we have

$$\|g(\tilde{X}; \theta^{(0)})\|_2 = \sqrt{\|\nabla_{g_{nn}}^{(j)} f(\tilde{X}; \theta^{(0)})\|_2^2 + \sum_{l=1}^L \|\nabla_{v_{L-1}}^{(j)} f(\tilde{X}; \theta^{(0)})\|_2^2} \leq m^{-1} \beta_h \cdot \sqrt{L \cdot \beta_L^5 \cdot (\beta_L \zeta)^{2L} + m}.$$

Then, for the norm of gradients, $\Theta \in \{\theta_{g_{nn}}, \theta_1, \ldots, \theta_{L-1}\}$, we have

$$\|g(\tilde{X}; \theta^{(0)}) - g(\tilde{X}; \theta^{(j)})\|_2 \leq \sqrt{L \cdot \Lambda(j)^2 + \Lambda(j)^2 / m} = \Lambda(j) \cdot \sqrt{L + m^{-1}}.$$
B LEMMAS FOR GRADIENT MATRICES

Inspired by [4, 46] and with sufficiently large network width \(m\), the trained network parameter can be related to ridge regression estimator where the context is embedded by network gradients. With the received contexts and rewards up to time step \(t\), we have the estimated parameter \(\hat{\Theta}\) as \(\hat{\Theta} = (Z_0)^{-1}b_0\) where \(Z_0 = \lambda I + \frac{1}{m} \Sigma_{t=1}^g g(\overline{X};\Theta_0) g(X;\Theta_0)\). We also define the gradient matrix w.r.t. the network parameters as

\[
G^{(j)} = (g(\overline{X};\Theta^{(j)}), \ldots, g(\overline{X};\Theta^{(j)}))
\]

\[
f^{(j)} = (f(\overline{X};\Theta^{(j)}), \ldots, f(\overline{X};\Theta^{(j)})), \quad r = (r_1, \ldots, r_t)
\]

\[
\Theta^{(j+1)} = \Theta^{(j)} - \eta \cdot (G^{(j)})^\top (f^{(j)} - r).
\]

where the \(t\) notation is omitted by default. Then, we use the following Lemma to bound the above matrices.

**Lemma 1.** After \(j\) iterations, with the probability of at least \(1 - O(L) \cdot e^{-\Omega(m)}\), we have

\[
\|G^{(0)}\|_F \leq G_2 = m^{-1} \beta_h \cdot \sqrt{L \cdot \beta_2^2 \cdot (\lambda \beta_3^2) + m},
\]

\[
\|G^{(0)} - G^{(j)}\|_F \leq \Lambda^{(j)} \cdot \sqrt{L \cdot (9L + m - 1)},
\]

\[
\|G^{(j)}\|_F \leq \tilde{I} = \sqrt{L \cdot \beta_2^2 \cdot (\lambda \beta_3^2)L + m} + \Lambda^{(j)} \sqrt{L \cdot (9L + m - 1)}
\]

\[
\|f^{(j)} - (G^{(j)})^\top (\Theta^{(j)} - \hat{\Theta}^{(0)})\|_2 \leq \sqrt{t} \cdot \Lambda^{(j)} \cdot \sqrt{L \cdot \beta_2^2 \cdot (\lambda \beta_3^2)L + m} + \Lambda^{(j)} \cdot \sqrt{L \cdot (9L + m - 1)}
\]

**Proof.** For the gradient matrix after random initialization, we have

\[
\|G^{(0)}\|_F = \sum_{t=1}^T \|g(\overline{X};\Theta^{(0)})\|_2^2 \leq m^{-1} \beta_h \cdot \sqrt{L \cdot \beta_2^2 \cdot (\lambda \beta_3^2)L + m}
\]

with the conclusion from Lemma A.5. Then,

\[
\|G^{(0)} - G^{(j)}\|_F = \sum_{t=1}^T \|g(\overline{X};\Theta^{(0)}) - g(\overline{X};\Theta^{(j)})\|_2^2 \leq \Lambda^{(j)} \cdot \sqrt{L \cdot (9L + m - 1)}.
\]

For the third inequality in this Lemma, we have

\[
\|f^{(j)} - (G^{(j)})^\top (\hat{\Theta}^{(j)} - \Theta^{(0)})\|_2 \leq \sqrt{t} \cdot m^{-0.5} [\beta_3 (\Lambda^{(j)} + \beta_h) + \Lambda^{(j)} \cdot \beta_3 + \beta_h (\Lambda^{(j)} / \beta + 1)]
\]

based on Lemma A.6.

Analogous to [4, 46], we define another auxiliary sequence to bound the parameter difference. With \(\hat{\Theta} = \hat{\Theta}^{(0)}\), we have \(\hat{\Theta}^{(j+1)} = \hat{\Theta}^{(j)} - \eta \cdot (G^{(j)})^\top (\hat{\Theta}^{(j)} - \hat{\Theta}^{(0)})\).

**Lemma 2.** After \(j\) iterations, with the probability of at least \(1 - O(L) \cdot e^{-\Omega(m)}\), we have

\[
\|\hat{\Theta}^{(j)} - \Theta^{(0)}\|_{\sqrt{m}} \leq \sqrt{t} / (m \lambda)
\]

**Proof.** The proof is analogous to Lemma 10.2 in [4] and Lemma C.4 in [46]. Switching \(G_0\) to \(G_j\) would give the result.

Then, we can have the following lemma to bridge the difference between the regression estimator \(\hat{\Theta}\) and the network parameter \(\Theta\).

**Lemma 3.** At this time step \(t\), with the notation defined in Lemma 5.3 and the probability at least \(1 - O(L) \cdot e^{-\Omega(m)}\), we will have

\[
\|\Theta - \Theta_0 - \Theta_t / \sqrt{m}\|_2 \leq (\tilde{I} \cdot \sqrt{t} / (m \lambda) + \tilde{I} / (\sqrt{t} / (m \lambda))
\]

with proper \(m, \eta\) as in Lemma 5.3

**Proof.** With an analogous approach from Lemma 6.2 in [4], we can have

\[
\|\Theta^{(j+1)} - \Theta^{(j+1)}\|_2 \leq \eta \|G^{(j)}\|_2 \|f^{(j)} - (G^{(j)})^\top (\Theta^{(j)} - \Theta^{(0)})\|_2 + \eta m \lambda \|\Theta^{(0)} - \Theta^{(j)}\|_2 + \|I - \eta \cdot (m \lambda I + G^{(j)} (G^{(j)})^\top)\|_2 \|\Theta^{(j)} - \Theta^{(j)}\|_2 = I_1 + I_2 + I_3.
\]

With Lemma 1, we can bound them as

\[
I_1 \leq \eta \cdot \tilde{I} \cdot \sqrt{t} B_3
\]

\[
I_2 \leq \eta m \lambda \sum_{t=0}^{\tilde{I}} \|\Theta^{(0)} - \Theta^{(j)}\|_2 \leq \eta m \lambda \sqrt{L} + 1 \cdot \sqrt{\tilde{I}}
\]

based on the conclusion from Lemma A.2. For \(I_3\), we have

\[
\eta \cdot (m \lambda I + G^{(j)} (G^{(j)})^\top) \leq I
\]

\[
= \sqrt{t} \cdot m^{-0.5} [\beta_3 (\Lambda^{(j)} + \beta_h) + \Lambda^{(j)} \cdot \beta_3 + \beta_h (\Lambda^{(j)} / \beta + 1)]
\]

\[
= \sqrt{t} \cdot m^{-0.5} [\beta_3 (\Lambda^{(j)} + \beta_h) + \Lambda^{(j)} \cdot \beta_3 + \beta_h (\Lambda^{(j)} / \beta + 1)]
\]

which leads to

\[
\|\hat{\Theta}^{(j)} - \Theta^{(j)}\|_2 \leq (\tilde{I} \cdot \sqrt{t} B_3 + m \cdot \tilde{I} / (m \lambda))
\]

lengthwise

\[
\|\hat{\Theta}^{(j)} - \Theta^{(j)}\|_2 \leq \sqrt{t} / (m \lambda)
\]

Finally,

\[
\|\Theta_t - \Theta_0 - \Theta_t / \sqrt{m}\|_2 \leq \|\Theta^{(0)} - \Theta^{(j)}\|_2 + \|\Theta_t - \Theta_0 - \Theta_t / \sqrt{m}\|_2 \leq (\tilde{I} \cdot \sqrt{t} B_3 + m \cdot \tilde{I} / (m \lambda))
\]

which completes the proof.

**Lemma 4.** At this time step \(t\), with the probability at least \(1 - O(L) \cdot e^{-\Omega(m)}\), we will have

\[
\|Z_t\|_2 \leq \lambda + \eta + \frac{t (L + 1)}{m} (\Lambda^{(j)} \cdot \sqrt{9L + m - 1} \cdot \beta_3 \cdot \sqrt{L \cdot \beta_3^2 \cdot (\lambda \beta_3^2) + m} + \Lambda^{(j)} \cdot \sqrt{9L + m - 1} \cdot \beta_3 \cdot \sqrt{L \cdot \beta_3^2 \cdot (\lambda \beta_3^2) + m})
\]

\[
\|G_t^T G_t - G_0^T G_0\|_F \leq 2 t \cdot m^{-1} (\Lambda^{(j)} \cdot \sqrt{9L + m - 1} \cdot \beta_3 \cdot \sqrt{L \cdot \beta_3^2 \cdot (\lambda \beta_3^2) + m})
\]

\[
\|\hat{\Theta} - \Theta_0 - \Theta_t / \sqrt{m}\|_2 \leq \sqrt{t} / (m \lambda)
\]

with proper \(m, \eta\) as in Lemma 5.3

**Proof.** For the gradient matrix of ridge regression, we have

\[
\|Z_t\|_2 \leq \lambda + \eta + \frac{t (L + 1)}{m} (\Lambda^{(j)} \cdot \sqrt{9L + m - 1} \cdot \beta_3 \cdot \sqrt{L \cdot \beta_3^2 \cdot (\lambda \beta_3^2) + m})
\]
with the results from Lemma A.5. Then, 
\[ \|G_t^T G_t - G_t^T G_0\|_F \leq m^{-1}. \]
\[ \left[ \sum_{t=1}^{T} \|g(\tilde{X}_t; \Theta_t) + g(\tilde{X}_t; \Theta_0)\|_2^2 + \|g(\tilde{X}_t; \Theta_t) - g(\tilde{X}_t; \Theta_0)\|_2^2 \right] \]
\[ \leq 2t \cdot m^{-1} \cdot (\Lambda(t) \sqrt{\Omega} + m^{-1} + m^{-1} \beta_h \cdot \sqrt{\Omega} \cdot \beta_h \cdot (\beta_h \xi)^{2L} + m) \]
\[ (\Lambda(t) \cdot \sqrt{\Omega} + m^{-1}) \leq B_G/m. \]
The proof is then completed.

**Proof sketch for Lemmas B.1-B.4.** Analogous to lemmas in Section A, Lemma B.1 is proved by Lemmas A.5, A.6 by breaking the target into the product of norms. The proof of Lemma B.2 is analogous to Lemma 10.2 in [4] and Lemma C.4 in [46], then replacing \(G_0\) with \(G_j\) would give the result. Then, based on Lemma B.2 results, Lemma B.3 will be proved with after bounding \(\|G(t+1) - G(j+1)\|_2\) by induction. Finally, Lemma B.4 is proved by decomposing the norm into sum of individual terms, and bounding these terms with bounds on gradients in Lemma A.5.

**C LEMMAS FOR MODEL CONVERGENCE**

Lemma C.1. After \(T\) time steps, assume the model with width \(m\) defined in Lemma 5.3 are trained with the \(J\)-iterations GD on the past contexts and rewards. Then, there exists a constant \(\beta_F\), such that \(\beta_F \cdot \eta \leq 1\), for any \(j \in [J]\):
\[ \|V(j)\|_2 \leq \frac{1}{4} \eta \beta_F \cdot \|F(j) - Y_T\|_2 \]
where \(F(j) = [f(G_r, \tilde{X}_r; \Theta(j))]_{r=1}^{T}, \) and \(Y_T = [r_T]_{r=1}^{T}\).

**Proof.** We prove this lemma following an analogous approach as Lemma B.6 in [13]. Given \(X\), we denote \(\nabla L(\Theta(j)) = \frac{\partial L(\Theta(j))}{\partial \Theta}\), and \(\nabla f(\Theta(j)) = \frac{\partial f(G_r, X; \Theta(j))}{\partial \Theta}\), where \(\Theta \in \{\Theta_{\text{gmn}}, \Theta_1, \ldots, \Theta_L\}\).

By the definition of \(\|V(j)\|_2\), we have its element \(|V(j)(\tilde{X})|\)
\[ \leq \eta \cdot \max_{0 \leq s \leq T} \left[ \sum_{\Theta} \|\nabla L(\Theta(j))\|_F \|\nabla f(\Theta(j)) - \nabla f(\Theta(j), s)\|_F \right] \]

With the notation and conclusion from Lemma A.2, we have
\[ \|\nabla L(\Theta(j))\|_F \leq m^{-\frac{1}{2}} \sqrt{T} \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

Meantime, \(\|V(f)(\Theta(j)) - \nabla f(\Theta(j), s)\|_F = m^{-\frac{1}{2}} \frac{1}{4} \eta \beta_F \cdot \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \)
\[ \|\Gamma(j)\|_2 \leq \frac{2}{m} \eta \beta_F \cdot \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

With \(Y \sqrt{m} \leq 1\), and \(\Lambda(j) \leq \beta_h\) and a similar procedure as in Lemma A.3 and Lemma A.2, we have
\[ \|\Theta(j+1) - \Theta(j)\|_F \leq \eta \frac{\gamma(j)}{\sqrt{m}}, \quad \|\Theta(j)\|_F \leq 2\beta_h \sqrt{m} \]
\[ \|\Gamma(j+1) - \Gamma(j)\|_F \leq 2\eta \beta_F \cdot \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]
\[ \|\Gamma(j+1) - \Gamma(j)\|_F \leq 2\eta \beta_F \cdot \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

With Lemma G.1 from [13], for \(\Theta \in \{\Theta_{\text{gmn}}, \Theta_1, \ldots, \Theta_L\}\),
\[ \|\nabla f(\Theta(j)) - \nabla f(\Theta(j), s)\|_F \leq \frac{4\gamma}{\sqrt{m}} \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

Combining with \(\|\nabla L(\Theta(j))\|_F\), we have
\[ \|V(j)(\tilde{X})\| \leq \eta \eta \frac{4T}{m} \cdot \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

Since this inequality holds for an arbitrary \(\tilde{X} \in \{|X_r|_{r=1}^T\} \) and \(\|F(j) - Y_T\|_2 \leq C(\sqrt{T})\), given network width \(m\), we finally have
\[ \|V(j)\| \leq \frac{1}{4} \eta \beta_F \cdot \|F(j) - Y_T\|_2 \]

with the choice of learning rate \(\eta \leq O(T^{-1} L^{-1} \beta_h^{-2} (2\xi)^{2L}).\)

**Proof of Lemma 5.7.** We prove this lemma following an analogous approach as Lemma B.7 in [13]. By the model definition and substituting \(Y(j) \cdot \sqrt{m} \) with \(m^{-\frac{1}{2}} \cdot \sqrt{T} \|F(j) - Y_T\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \) as the upper bound based on Lemma A.2, with \(\Lambda(j) \leq \beta_h\), we have
\[ \|F(j) - F(j+1)\|_2 \leq \frac{1}{m} \sum_{r=1}^{T} \left| \sum_{\{j=1\} \cup \{j+1\}}^T \right| \|\Gamma(j)\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]
\[ \leq \frac{1}{m} \|\Gamma(j)\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

\[ \|J(j+1) - J(j)\|_2 \leq \frac{1}{m} \|\Gamma(j)\|_2 \cdot \xi^L \cdot (2\beta_h)^L \beta_h \]

where the last inequality is due to sufficiently large \(m\) and the choice of learning rate \(\eta\).