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, 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

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]; on 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

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539312

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 discuss 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 $C = \{1, \dots, N_c\}$ 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 $\mathcal{D}_c$. 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 $\mathcal{X}_{c,t} = \{\boldsymbol{x}_{c,t}^{(1)}, \cdots \boldsymbol{x}_{c,t}^{(n_{c,t})}\}$ with the size of $|\mathcal{X}_{c,t}| = n_{c,t}$. Then, $\forall i \in [n_{c,t}] = \{1, \dots, n_{c,t}\}$, we suppose $\boldsymbol{x}_{c,t}^{(i)} \sim \mathcal{D}_c$ with the dimensionality $\boldsymbol{x}_{c,t}^{(i)} \in \mathbb{R}^{d_x}$. Therefore, in the t-th round, we receive

$$\{\mathcal{X}_{c,t} | c \in C_t\} \text{ and } \mathcal{X}_{c,t} = \{\boldsymbol{x}_{c,t}^{(1)}, \cdots \boldsymbol{x}_{c,t}^{(n_{c,t})}\}, \forall c \in C_t. \tag{1}$$

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

$$r_{c,t}^{(i)} = h(\boldsymbol{W}^*, \boldsymbol{x}_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)} \tag{2}$$

where $h(\cdot)$ represents the unknown reward mapping function, and ϵ is the zero-mean Gaussian noise. For brevity, let $\boldsymbol{x}_t$ be the arm we select in round t and r_t be the corresponding received reward.

Our goal is recommending arm $\boldsymbol{x}_t$ (with reward r_t) at each time step t to minimize the cumulative pseudo-regret $R(T) = \sum_{t=1}^{T} \mathbb{E}\left[(r_t^* - r_t)\right]$ where $\mathbb{E}[r_t^*] = \max_{(c \in C_t, i \in [n_{c,t}])} \left[h(\boldsymbol{W}^*, \boldsymbol{x}_{c,t}^{(i)})\right]$. At each time step t, the overall set of received contexts is defined as $\mathcal{X}_t = \{\boldsymbol{x}_{c,t}^{(i)}\}_{c \in C_t, i \in [n_{c,t}]}$. 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 $\mathcal{X}_{c,t} \cap \mathcal{X}_{c',t} \neq \emptyset$.

In order to model the arm group correlations, we maintain an undirected graph $\mathcal{G}_t = (V, E, W_t)$ at each time step t, where each arm group from C is mapped to a corresponding node in node set V.

ALGORITHM 1: AGG-UCB

Input: Number of rounds T, exploration parameter γ, regularization parameter λ, network width m, network depth L, neighborhood size k.
Output: Arm recommendation $\boldsymbol{x}_t$ for each time step t.
Initialization: Initialize the arm group graph as a connected graph $\mathcal{G}_1 = (V, E, W_1)$. Initialize gradient matrix $Z_0 = \lambda \boldsymbol{I}$. Initialize parameter $\boldsymbol{\Theta_0}$ for the model $f(\mathcal{G}, X; \boldsymbol{\Theta_0})$.
for $t = 1, 2, ..., T$ **do**
  Receive a set of arm contexts $\mathcal{X}_t = \{\boldsymbol{x}_{c,t}^{(i)}\}_{c \in C_t, i \in [n_{c,t}]}$.
  Embed the arm set $\mathcal{X}_t$ into $\widetilde{\mathcal{X}}_t$ w.r.t. **Eq.**4.
  for *each embedded arm* $\widetilde{\boldsymbol{X}}_{c,t}^{(i)} \in \widetilde{\mathcal{X}}_t$ **do**
    Obtain the point estimate $\widehat{r}_{c,t}^{(i)} = f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}; \boldsymbol{\Theta_{t-1}})$.
    Obtain network gradient $\boldsymbol{g}_{c,t}^{(i)} \leftarrow \nabla_\Theta f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}; \boldsymbol{\Theta_{t-1}})$.
    Calculate confidence bound as $\widehat{i}_{c,t}^{(i)} = \sqrt{\boldsymbol{g}_{c,t}^{(i)\top} Z_{t-1} \boldsymbol{g}_{c,t}^{(i)} / m}$.
  end
  Recommend $\widetilde{\boldsymbol{X}}_t = \text{argmax}_{\widetilde{\boldsymbol{X}}_{c,t}^{(i)} \in \widetilde{\mathcal{X}}_t} (\widehat{r}_{c,t}^{(i)} + \gamma \cdot \widehat{i}_{c,t}^{(i)})$ with the received reward represented as r_t.
  Calculate arm group distances w.r.t. **Eq.**3, and update the arm group graph $\mathcal{G}_t$ to $\mathcal{G}_{t+1}$.
  Update the model parameter $\boldsymbol{\Theta_{t-1}}$ to $\boldsymbol{\Theta_t}$ according to **Algorithm 2**.
  Retrieve the $\widetilde{\boldsymbol{X}}_t$'s gradient vector $\boldsymbol{g}_t$, and update gradient matrix $Z_t = Z_{t-1} + \boldsymbol{g}_t \cdot \boldsymbol{g}_t^\top$.
end

Then, $E = \{e(c_i, c_j)\}_{c_i, c_j \in C}$ is the set of edges, and W_t represents the set of edge weights. Note that by definition, $\mathcal{G}_t$ 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 $\widetilde{\mathcal{N}}_k(v) = \mathcal{N}_k(v) \cup \{v\}$ as the union node set of its k-hop neighborhood $\mathcal{N}_k(v)$ and node v itself. For the arm group graph $\mathcal{G}_t$, we denote $\boldsymbol{A}_t \in \mathbb{R}^{N_c \times N_c}$ as the adjacency matrix (with added self-loops) of the given arm group graph and $\boldsymbol{D}_t \in \mathbb{R}^{N_c \times N_c}$ as its degree matrix. For the notation consistency, we will apply a true arm group graph $\mathcal{G}^*$ instead of $\boldsymbol{W}^*$ in **Eq.** 2 to represent the true arm group correlation.

4 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 $\mathcal{X}_t = \{\boldsymbol{x}_{c,t}^{(i)}\}_{c \in C_t, i \in [n_{c,t}]}$ (line 5).

ALGORITHM 2: Model Training

Input: Initial parameter $\boldsymbol{\Theta_0}$, step size η, training steps J, network width m. Updated arm group graph $\mathcal{G}_{t+1}$. Selected embedded contexts $\{\widetilde{\boldsymbol{X}}_\tau\}_{\tau=1}^t$.
Output: Updated model parameter $\boldsymbol{\Theta_t}$.
$\boldsymbol{\Theta}_t^0 \leftarrow \boldsymbol{\Theta}_0$.
Let $\mathcal{L}(\boldsymbol{\Theta}) = \frac{1}{2} \sum_{\tau=1}^t |f(\mathcal{G}_{t+1}, \widetilde{\boldsymbol{X}}_\tau; \boldsymbol{\Theta}) - r_\tau|^2$
for $j = 1, 2, \ldots, J$ **do**
  $\boldsymbol{\Theta}_t^j = \boldsymbol{\Theta}_t^{j-1} - \eta \cdot \nabla_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\Theta}_t^{j-1})$
end
Return new parameter $\boldsymbol{\Theta}_t^J$.

Then, we embed the arm set $\mathcal{X}_t$ to $\widetilde{\mathcal{X}}_t$ based on **Eq.**4 from Subsection 4.3 (line 6). For each embedded arm $\widetilde{\boldsymbol{X}} \in \widetilde{\mathcal{X}}_t$, its estimated reward $\widehat{r}$ and confidence bound $\widehat{i}$ would be calculated (line 8-10) with the model $f(\cdot)$ in Subsection 4.4. After recommending the best arm $\widetilde{\boldsymbol{X}}_t$ (line 12) and receiving its true reward r_t, we update the current arm group graph $\mathcal{G}_t$ based on Subsection 4.2 (line 13). Then, the model parameters $\boldsymbol{\Theta_{t-1}}$ will be trained based on **Algorithm 2** (line 14), and we incrementally update the gradient matrix to $Z_t = Z_{t-1} + \boldsymbol{g}_t \cdot \boldsymbol{g}_t^\top$ with the gradient vector $\boldsymbol{g}_t$ of model $f(\cdot)$ given the selected arm $\widetilde{\boldsymbol{X}}_t$ (line 15).

The steps from **Algorithm 2** demonstrate our training process for AGG-UCB parameters. With the updated arm group graph $\mathcal{G}_{t+1}$ and the past embedded arm contexts $\{\widetilde{\boldsymbol{X}}_\tau\}_{\tau=1}^t$ until current time step t, we define the loss function as the straightforward quadratic loss function (line 4). Finally, we run gradient descent (GD) for J steps to derive the new model parameters Θ_t (lines 5-7) based on the initial parameters $\boldsymbol{\Theta}_0$ (initialized in Subsection 4.5). Next, we proceed to introduce the detail of framework components.

4.2 Arm Group Graph Estimation

Recall that at time step t, we model the similar arms into an arm group graph $\mathcal{G}_t = (V, E, W_t)$ where the nodes V are corresponding to the arm groups from C and edges weights W_t formulate the correlations among arm groups. Given two nodes $\forall c, c' \in C$, to measure the similarity between them, inspired by the kernel mean embedding in the multi-task learning settings [9, 12], we define edge weight between c and c' as:

$$w^*(c, c') = \exp(-\|\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_c}\left[\phi_{k_\mathcal{G}}(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}_{c'}}\left[\phi_{k_\mathcal{G}}(\boldsymbol{x}')\right]\|^2 / \sigma_s)$$

where $\phi_{k_\mathcal{G}}(\cdot)$ is the induced feature mapping of a given kernel $k_\mathcal{G}$, e.g., a radial basis function (RBF) kernel. Unfortunately, $\forall c \in C$, $\mathcal{D}_c$ is unknown. Therefore, we update the edge weight based on the empirical estimation of arm group correlations. Here, let $\mathcal{X}_c^t = \{\boldsymbol{x}_{c,\tau}^{(i)}\}_{\tau \in [t], i \in [n_{c,\tau}]}$ represent the set of all arm contexts from group $c \in C$ up to time step t. We define the arm similarity measurement between arms $c, c' \in C$ through a Gaussian-like kernel as

$$w_t(c, c') = \exp(-\|\Psi_t(\mathcal{D}_c) - \Psi_t(\mathcal{D}_{c'})\|^2 / \sigma_s) \tag{3}$$

where $\Psi_t(\mathcal{D}_c) = \frac{1}{|\mathcal{X}_c^t|} \sum_{x \in \mathcal{X}_c^t} k_\mathcal{G}(\cdot, x)$ denotes the kernel mean estimation of $\mathcal{D}_c$ with a given kernel $k_\mathcal{G}$; and σ_s refers to the bandwidth. Then, at time step t and $\forall c, c' \in C$, we update the corresponding weight of edge $e(c, c')$ in the weight set W_t with $w_t(c, c')$.

4.3 Group-Aware Arm Embedding

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

$$\widetilde{\boldsymbol{X}}_{c,t}^{(i)} = \begin{pmatrix} (\boldsymbol{x}_{c,t}^{(i)})^\intercal & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{x}_{c,t}^{(i)})^\intercal & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\boldsymbol{x}_{c,t}^{(i)})^\intercal \end{pmatrix} \in \mathbb{R}^{N_c \times d_{\widetilde{x}}} \quad (4)$$

where $d_{\widetilde{x}} = d_x \cdot N_c$ is the column dimension of $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$. Here, for the c'-th row in matrix $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$, the $((c'-1)\cdot d_x + 1)$-th to the $(c' \cdot d_x)$-th entries are the transposed original arm context $(\boldsymbol{x}_{c,t}^{(i)})^\intercal$, while the other entries are zeros. Receiving a set of arm contexts $\mathcal{X}_t$, we derive the corresponding embedded arm set as $\widetilde{\mathcal{X}}_t = \{\widetilde{\boldsymbol{X}}_{c,t}^{(i)}\}_{c \in C_t, i \in [n_{c,t}]}$.

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 $\widetilde{\mathcal{N}}_k(\cdot)$ for incorporating arm group correlations to obtain the aggregated group-aware embedding for an embedded arm $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$, denoted by

$$\boldsymbol{H}_{gnn} = \sqrt{\frac{1}{m}} \cdot \sigma(\boldsymbol{S}_t^k \cdot \widetilde{\boldsymbol{X}}_{c,t}^{(i)} \boldsymbol{\Theta}_{gnn}) \in \mathbb{R}^{N_c \times m} \quad (5)$$

where $\boldsymbol{S}_t = \boldsymbol{D}_t^{-\frac{1}{2}} \boldsymbol{A}_t \boldsymbol{D}_t^{-\frac{1}{2}}$ is the symmetrically normalized adjacency matrix, and we have

$$\boldsymbol{\Theta}_{gnn} = \begin{pmatrix} \boldsymbol{\Theta}_{gnn}^1 \in \mathbb{R}^{d_x \times m} \\ \vdots \\ \boldsymbol{\Theta}_{gnn}^{c'} \in \mathbb{R}^{d_x \times m} \\ \vdots \\ \boldsymbol{\Theta}_{gnn}^{N_c} \in \mathbb{R}^{d_x \times m} \end{pmatrix} \in \mathbb{R}^{d_{\widetilde{x}} \times m}.$$

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 $(\widetilde{\boldsymbol{X}}_{c,t}^{(i)} \cdot \boldsymbol{\Theta}_{gnn})$, denoted by $[\widetilde{\boldsymbol{X}}_{c,t}^{(i)} \cdot \boldsymbol{\Theta}_{gnn}]_{c',:}$, is the hidden representation of arm $\boldsymbol{x}$ in terms of c'-th arm group in C. Then, these hidden representations will then be aggregated over $\widetilde{\mathcal{N}}_k(c), c \in C$ by multiplying with $\boldsymbol{S}_t^k$ to derive the aggregated arm representation for $\boldsymbol{x}$, i.e., $\boldsymbol{H}_{gnn}(x)$.

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 $d' = m + d_{\widetilde{x}}$, we then have $\boldsymbol{H}_0 \in \mathbb{R}^{N_c \times d'}$ as the output group-aware arm representation for $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$, represented by

$$\boldsymbol{H}_0 = f_{gnn}(\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}; \boldsymbol{\Theta}_{gnn}) = [\sigma(\boldsymbol{S}_t^k \cdot \widetilde{\boldsymbol{X}}_{c,t}^{(i)} \boldsymbol{\Theta}_{gnn}); \widetilde{\boldsymbol{X}}_{c,t}^{(i)}] \quad (6)$$

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 $\boldsymbol{H}_0$.

4.4.1 Reward and confidence bound estimation. Here, let $\boldsymbol{\Theta}_{fc} = \{\boldsymbol{\Theta}_l\}_{l \in [L]}$ be the set of trainable weight matrices of a fully-connected network, where the specifications are: $\boldsymbol{\Theta}_1 \in \mathbb{R}^{d' \times m}$, $\boldsymbol{\Theta}_L \in \mathbb{R}^m$ and $\boldsymbol{\Theta}_l \in \mathbb{R}^{m \times m}, \forall l \in \{2, \ldots, L-1\}$. Then, given the group-aware representation $\boldsymbol{H}_0$, we have the reward estimation module as follows

$$\begin{aligned} \boldsymbol{H}_l &= \sqrt{\frac{1}{m}} \cdot \sigma(\boldsymbol{H}_{l-1} \cdot \boldsymbol{\Theta}_l),\ l \in \{1, \ldots, L-1\}, \\ \widehat{\boldsymbol{r}}_{all} &= f_{fc}(\boldsymbol{H}_0; \boldsymbol{\Theta}_{fc}) = \sqrt{\frac{1}{m}} \cdot \boldsymbol{H}_{L-1} \cdot \boldsymbol{\Theta}_L \end{aligned} \quad (7)$$

where $\widehat{\boldsymbol{r}}_{all} \in \mathbb{R}^{N_c}$ represents the point-estimation vector for the received contexts embedding $\boldsymbol{H}_0$ with respect to all the arms groups. Given that the arm $\widetilde{\boldsymbol{x}}_{c,t}^{(i)}$ belonging to c-th group, we will then have the reward estimation $\widehat{r}_{c,t}^{(i)} = [\widehat{\boldsymbol{r}}_{all}]_c \in \mathbb{R}$ for the embedded context matrix $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$, which is the c-th element of $\widehat{\boldsymbol{r}}_{all}$.

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

$$\widehat{r}_{c,t}^{(i)} = f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}; \boldsymbol{\Theta}) = \left[\left(f_{fc}(\cdot; \boldsymbol{\Theta}_{fc}) \circ f_{gnn}(\cdot; \boldsymbol{\Theta}_{gnn}) \right) (\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}) \right]_c .$$

Setting $p = (2N_c \cdot d) \cdot m + (L-1) \cdot m^2 + m$, we have $\boldsymbol{\Theta} \in \mathbb{R}^p$ 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 $\widehat{\imath} = \sqrt{\boldsymbol{g}^\intercal \cdot \boldsymbol{Z}_{t-1} \cdot \boldsymbol{g}/m}$ where $\boldsymbol{g} = \nabla_{\boldsymbol{\Theta}} f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}_{c,t}^{(i)}; \boldsymbol{\Theta}) \in \mathbb{R}^p$ is the gradient vector, and $\boldsymbol{Z}_{t-1} = \boldsymbol{I} + \sum_{\tau=1}^{t-1} \boldsymbol{g}_\tau \cdot \boldsymbol{g}_\tau^\intercal$ with $\boldsymbol{g}_\tau$ 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 $\widetilde{\mathcal{X}}_t$, we choose the best arm as $\widetilde{\boldsymbol{X}}_t = \operatorname{argmax}_{\widetilde{\boldsymbol{X}}_{c,t}^{(i)} \in \widetilde{\mathcal{X}}_t} (\widehat{r}_{c,t}^{(i)} + \gamma \cdot \widehat{\imath}_{c,t}^{(i)})$ where γ 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 $\mathbf{\Theta}_{gnn}$, 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 ($[\mathbf{\Theta}_1,\ldots,\mathbf{\Theta}_{L-1}]$) are also sampled from $N(0,1)$. For the final (L-th) layer, its weight matrix $\mathbf{\Theta}_L$ 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 $\boldsymbol{x}_{c,t}$, which makes $|\mathcal{X}_1^t| = \cdots = |\mathcal{X}_{N_c}^t| = t$. We also apply the adjacency matrix $\boldsymbol{A}_t$ instead of $\boldsymbol{S}_t$ for aggregation, and set its elements $[\boldsymbol{A}_t]_{ij} = \frac{1}{t\cdot N_c}\sum_{\tau=1}^{t}\phi_{k_{\mathcal{G}}}(x_{c_i,\tau})^\intercal\phi_{k_{\mathcal{G}}}(x_{c_j,\tau})$ for arm group similarity between group $c_i, c_j \in C$. Here, $\phi_{k_{\mathcal{G}}}(\cdot)$ is the kernel mapping given an RBF kernel $k_{\mathcal{G}}$. With $\mathcal{G}^*$ being the unknown true arm group graph, its adjacency matrix elements are $[\boldsymbol{A}^*]_{ij} = \frac{1}{N_c}\mathbb{E}_{x_i\sim\mathcal{D}_{c_i},x_j\sim\mathcal{D}_{c_j}}(\phi_{k_{\mathcal{G}}}(x_i)^\intercal\phi_{k_{\mathcal{G}}}(x_j))$. Note that the norm of adjacency matrices $\|\boldsymbol{A}^*\|_2, \|\boldsymbol{A}_t\|_2 \le 1$ since $\langle\phi_{k_{\mathcal{G}}}(x),\phi_{k_{\mathcal{G}}}(x')\rangle \le 1$ for any $x, x' \in \mathbb{R}^{d_x}$, which makes it feasible to aggregate over k-hop neighborhood without the explosion of eigenvalues. Before presenting the main results, we first introduce the following background.

Lemma 5.1 ([4, 46]). *For any $t \in [T]$, given arm $\boldsymbol{x} \in \mathbb{R}^{d_x}$ satisfying $\|\boldsymbol{x}\|_2 = 1$ and its embedded context matrix $\widetilde{\boldsymbol{X}}$, there exists $\mathbf{\Theta}_{t-1}^* \in \mathbb{R}^p$ at time step t, and a constant $S > 0$, such that*

$$h(\mathcal{G}^*, \widetilde{\boldsymbol{X}}) = \langle g(\mathcal{G}^*, \widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1}), \mathbf{\Theta}_{t-1}^* - \mathbf{\Theta}_0\rangle \tag{8}$$

where $\|\mathbf{\Theta}_{t-1}^* - \mathbf{\Theta}_0\|_2 \le S/\sqrt{m}$, $\forall t \in [T]$, and $\mathcal{G}^*$ stands for the unknown true underlying arm group graph.

Note that with sufficient network width m, we will have $\mathbf{\Theta}_{t-1}^* \approx \mathbf{\Theta}_0^*, \forall t \in [T]$, 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 $\mathbf{\Theta}_0$ and the parameter $\mathbf{\Theta}_{t-1}^*$, which lies in the confidence set with the high probability [1]. Then, regarding the activation function $\sigma(\cdot)$, we have the following assumption on its continuity and smoothness.

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

$$|\sigma(x) - \sigma(x')| \le \zeta\cdot\|x - x'\|, \quad |\sigma'(x) - \sigma'(x')| \le \zeta\cdot\|x - x'\|$$

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

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 $\widetilde{\boldsymbol{X}}$, the output of our proposed framework is $\widehat{r} = f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})$ with $\mathcal{G}_t$, $\mathbf{\Theta}_{t-1}$ as the estimated arm group graph and trained parameters respectively. The true function $h(\mathcal{G}^*, \widetilde{\boldsymbol{X}})$ is given in Lemma 5.1. Supposing there exists the true arm group graph $\mathcal{G}^*$, the confidence bound for a single round t will be

$$\begin{aligned}\mathrm{CB}_t(\widetilde{\boldsymbol{X}}) &= |f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1}) - h(\mathcal{G}^*, \widetilde{\boldsymbol{X}})| \\ &\le \underbrace{|f(\mathcal{G}_t, \widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1}) - h(\mathcal{G}_t, \widetilde{\boldsymbol{X}})|}_{R_1} + \underbrace{|h(\mathcal{G}_t, \widetilde{\boldsymbol{X}}) - h(\mathcal{G}^*, \widetilde{\boldsymbol{X}})|}_{R_2}\end{aligned} \tag{9}$$

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 $\mathcal{G}_t$ 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 $\widehat{\mathbf{\Theta}}_t = \boldsymbol{Z}_t^{-1}\boldsymbol{b}_t$ where $\boldsymbol{Z}_t = \lambda\boldsymbol{I} + \frac{1}{m}\sum_{\tau=1}^{t} g(\widetilde{\boldsymbol{X}}_\tau; \mathbf{\Theta}_\tau)\cdot g(\widetilde{\boldsymbol{X}}_\tau; \mathbf{\Theta}_\tau)^\intercal, \boldsymbol{b}_t = \sum_{\tau=1}^{t} r_\tau\cdot g(\widetilde{\boldsymbol{X}}_\tau; \mathbf{\Theta}_\tau)/\sqrt{m}$. Then, we derive the bound for R_1 with the following lemma.

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

$$\beta_h = \max\{\zeta\beta_1,\ \zeta\beta_2 + \zeta^2\beta_1\beta_2,\ \zeta\beta_L + 1,\ (\zeta\beta_4)^{L-2}(\zeta\beta_2 + \zeta^2\beta_1\beta_2)\}.$$

With a constant $\delta \in (0,1)$, and L as the layer number for the FC network, let the network width $m \ge \mathrm{Poly}\big(t, L, \frac{1}{\beta_F}, \frac{1}{\lambda}, (\zeta\beta_L)^L, \log(\frac{1}{\delta})\big)$, and learning rate $\eta \le O\big((t\cdot L\beta_h^2(2\zeta\beta_L)^{2L})^{-1}\big)$. Denoting the terms

$$\begin{aligned}\Upsilon &= \frac{2\sqrt{2}t}{\beta_F}(\beta_h + \Lambda)(\beta_L + 1)^L\zeta^L, \quad \Lambda = \frac{\zeta\Upsilon\beta_h}{m}\cdot\frac{(2\zeta\beta_L)^L - 1}{2\zeta\beta_L - 1} \\ \widetilde{I}_1 &= \sqrt{t\cdot(L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L} + m)} + \Lambda\cdot\sqrt{t\cdot(9L + m^{-1})}, \\ \widetilde{I}_2 &= \lambda\sqrt{L+1}\cdot\Upsilon/\sqrt{m},\end{aligned}$$

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

$$|h(\widetilde{\boldsymbol{X}}) - f(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})| \le B_1\|g(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})/\sqrt{m}\|_{\boldsymbol{Z}_{t-1}^{-1}} + B_2 + B_3$$

with the terms

$$\begin{aligned}B_1 &= \sqrt{\log(\frac{\det(\boldsymbol{Z}_{t-1})}{\det(\lambda\boldsymbol{I})}) - 2\log(\delta)} + \lambda^{\frac{1}{2}}S, \\ B_2 &= \Big(\frac{\widetilde{I}_1\cdot\sqrt{t}B_3 + m\cdot\widetilde{I}_2}{m\lambda} + \sqrt{\frac{t}{m\lambda}}\Big) \\ &\quad\cdot\big(\Lambda\cdot\sqrt{9L + m^{-1}} + m^{-1}\beta_h\cdot\sqrt{L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L} + m}\big) \\ B_3 &= m^{-0.5}\big(\beta_3(\Lambda + \beta_h) + L\cdot\Upsilon\cdot(\Lambda + \beta_h)(\Lambda/\beta_h + 1)\big).\end{aligned}$$

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

$$\begin{aligned}&|h(\widetilde{\boldsymbol{X}}) - f(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})| \\ &\le |\langle g(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})/\sqrt{m}, \sqrt{m}(\mathbf{\Theta}_{t-1}^* - \mathbf{\Theta}_0)\rangle - \langle g(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})/\sqrt{m}, \widehat{\mathbf{\Theta}}_{t-1}\rangle| \\ &\quad + |\langle g(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})/\sqrt{m}, \widehat{\mathbf{\Theta}}_{t-1}\rangle - f(\widetilde{\boldsymbol{X}}; \mathbf{\Theta}_{t-1})| = R_3 + R_4.\end{aligned}$$

With Theorem 2 from [1], we have $R_3 \leq B_1\|g(\widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}}$. Then, for R_4, we have $|f(\widetilde{X};\Theta_{t-1}) - \langle g(\widetilde{X};\Theta_{t-1}/\sqrt{m}), \widehat{\Theta}_{t-1}\rangle|$

$$\leq R_5 + R_6 = |f(\widetilde{X};\Theta_{t-1}) - \langle g(\widetilde{X};\Theta_{t-1}), \Theta_{t-1} - \Theta_0\rangle| + |\langle g(\widetilde{X};\Theta_{t-1}), \Theta_{t-1} - \Theta_0 - \widehat{\Theta}_{t-1}/\sqrt{m}\rangle|$$

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

$$\begin{aligned} R_6 &\leq \|\Theta_{t-1} - \Theta_0 - \widehat{\Theta}_{t-1}/\sqrt{m}\|_2 \cdot \|g(\widetilde{X};\Theta_{t-1})\|_2 \\ &\leq B_2 = \big((\widetilde{I}_1 \cdot \sqrt{t}B_3 + m \cdot \widetilde{I}_2)/(m\lambda) + \sqrt{t/(m\lambda)}\big) \\ &\quad \cdot \big(\Lambda \cdot \sqrt{9L + m^{-1}} + m^{-1}\beta_h \cdot \sqrt{L \cdot \beta_3^2 \cdot (\beta_L\zeta)^{2L} + m}\big), \end{aligned}$$

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

$$\boldsymbol{H}_{gnn} = \sqrt{\frac{1}{m}} \cdot \sigma(\boldsymbol{A}_t^k \cdot \widetilde{\boldsymbol{X}}\boldsymbol{\Theta}_{gnn}) \in \mathbb{R}^{N_c \times m}$$

as the output where Θ_{gnn} 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 $\mathcal{X}_{c_i}^t = \{\boldsymbol{x}_{c_i,\tau}\}_{\tau=1}^t$, $\mathcal{X}_{c_j}^t = \{\boldsymbol{x}_{c_j,\tau}\}_{\tau=1}^t$, with the notation from Lemma 5.3 and the probability at least $1-\delta$, we have*

$$\|\boldsymbol{A}^* - \boldsymbol{A}_t\|_{\max} \leq \frac{1}{N_c} \cdot \sqrt{\frac{1}{2t}\log(\frac{N_c^2 - N_c}{\delta})}$$

where $\|\cdot\|_{\max}$ refers to the greatest entry of a matrix. Then, we will have $R_2 \leq B_4\sqrt{1/t}$ with

$$B_4 = \frac{S\sqrt{Lk}}{\sqrt{m}}(\beta_h + \Lambda)(\zeta\beta_L + \frac{\Upsilon\zeta}{m})^{O(L)}\sqrt{\frac{1}{2}\log(\frac{N_c^2 - N_c}{\delta})},$$

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

Proof. Recall that for $c_i, c_j \in C$, the element of matrix $[\boldsymbol{A}^*]_{ij} = \frac{1}{N_c}\mathbb{E}_{x_i\sim\mathcal{D}_{c_i}, x_j\sim\mathcal{D}_{c_j}}(\phi_{k_\mathcal{G}}(x_i)^\intercal \phi_{k_\mathcal{G}}(x_j)), \forall i, j \in [N_c]$, and $[\boldsymbol{A}_t]_{ij} = \frac{1}{t\cdot N_c}\sum_{\tau=1}^t \phi_{k_\mathcal{G}}(x_{c_i,\tau})^\intercal \phi_{k_\mathcal{G}}(x_{c_j,\tau})$. Here, suppose a distribution $\mathcal{D}_{ij}$ where $\mathbb{E}[\mathcal{D}_{ij}] = \frac{1}{N_c}\mathbb{E}_{x_i\sim\mathcal{D}_{c_i}, x_j\sim\mathcal{D}_{c_j}}(\phi_{k_\mathcal{G}}(x_i)^\intercal \phi_{k_\mathcal{G}}(x_j))$. Given N_c arm groups, we have $N_c(N_c-1)/2$ different group pairs. For group pair $c_i, c_j \in C$, each $\phi_{k_\mathcal{G}}(x_{c_i,\tau})^\intercal \phi_{k_\mathcal{G}}(x_{c_j,\tau}), \tau \in [t]$ is a sample drawn from $\mathcal{D}_{ij}$, and the element distance $|[\boldsymbol{A}_t]_{ij} - [\boldsymbol{A}^*]_{ij}|$ 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 $\boldsymbol{A}^k$ (for graph $\mathcal{G}$) as input and fix $\widetilde{\boldsymbol{X}}$. 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 $g(\mathcal{G}, \widetilde{X};\Theta_{t-1})$ being Lipschitz continuous w.r.t. the input graph as

$$\begin{aligned} R_2 &\leq \|g(\mathcal{G}^*, \widetilde{X};\Theta_{t-1}) - g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})\|_2 \cdot \|\Theta_{t-1}^* - \Theta_0\|_2 \\ &\leq \frac{S\sqrt{L}}{\sqrt{m}}(\beta_h + \Lambda)(\zeta\beta_L + \frac{\Upsilon\zeta}{\sqrt{m}})^{O(L)} \cdot |\|(\boldsymbol{A}_t)^k\|_2 - \|(\boldsymbol{A}^*)^k\|_2| \\ &\underset{(i)}{\leq} \frac{S\sqrt{Lk}}{\sqrt{m}}(\beta_h + \Lambda)(\zeta\beta_L + \frac{\Upsilon\zeta}{\sqrt{m}})^{O(L)} \cdot \|\boldsymbol{A}_t - \boldsymbol{A}^*\|_2 \end{aligned}$$

where (i) is because $\boldsymbol{A}_t, \boldsymbol{A}^*$ 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 δ, we have the confidence bound given embedded arm $\widetilde{\boldsymbol{X}}$ as

$$\mathrm{CB}_t(\widetilde{X}) \leq B_1\|g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}} + B_2 + B_3 + B_4\sqrt{\frac{1}{t}}. \quad (10)$$

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, η satisfy conditions in Lemma 5.3, we will have the regret*

$$\begin{aligned} R(T) &\leq 2 \cdot (2B_4\sqrt{T} + 2 - B_4) + 2\sqrt{2\widetilde{d}T\log(1 + T/\lambda) + 2T} \\ &\quad \cdot \big(\sqrt{\lambda}S + \sqrt{1 - 2\log(\delta/2) + (\widetilde{d}\log(1 + T/\lambda))}\big) \end{aligned}$$

where the effective dimension $\widetilde{d} = \frac{\log\det(\boldsymbol{I}+\boldsymbol{G}(0)/\lambda)}{\log(1+T/\lambda)}$ with $\boldsymbol{G}(0) = \boldsymbol{G}_0\boldsymbol{G}_0^\intercal$ and $\boldsymbol{G}_0 = \big(g(\widetilde{\boldsymbol{X}}_1;\Theta_0)^\intercal, \ldots, g(\widetilde{\boldsymbol{X}}_t;\Theta_0)^\intercal\big)$.

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

$$\begin{aligned} R_t &= h(\mathcal{G}^*, \widetilde{X}_t^*) - h(\mathcal{G}^*, \widetilde{X}_t) \\ &\leq \mathrm{CB}_t(\widetilde{X}_t^*) + f(\mathcal{G}_t, \widetilde{X}_t^*;\Theta_{t-1}) - h(\mathcal{G}^*, \widetilde{X}_t) \\ &\leq \mathrm{CB}_t(\widetilde{X}_t) + f(\mathcal{G}_t, \widetilde{X}_t;\Theta_{t-1}) - h(\mathcal{G}^*, \widetilde{X}_t) \leq 2 \cdot \mathrm{CB}_t(\widetilde{X}_t) \end{aligned}$$

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 $R(T) =$

$$\begin{aligned} \sum_{t=1}^T R_t &\leq 2\sum_{t=1}^T \Big(B_1\|g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}} + B_2 + B_3 + B_4\sqrt{\frac{1}{t}}\Big) \\ &\leq 2 \cdot (2B_4\sqrt{T} + 2 - B_4) + 2\sum_{t=1}^T (B_1\|g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}}) \end{aligned}$$

with the choice of m for bounding the summation of B_2, B_3, and the bound of $\sum_{i=1}^T [t^{-i/2}]$ in [10]. Then, with Lemma 11 from [1],

$$\begin{aligned} &\sum_{t=1}^T (B_1\|g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}}) \\ &\leq B_1\sqrt{T\sum_{t=1}^T \|g(\mathcal{G}_t, \widetilde{X};\Theta_{t-1})/\sqrt{m}\|_{Z_{t-1}^{-1}}^2} \leq \sqrt{T}B_1\sqrt{2\log(\frac{\det(Z_T)}{\det(\lambda\boldsymbol{I})})} \\ &\underset{(i)}{\leq} \sqrt{2\widetilde{d}T\log(1 + T/\lambda) + 2T}\big(\sqrt{\lambda}S + \sqrt{1 - 2\log(\delta/2) + (\widetilde{d}\log(1 + T/\lambda))}\big) \end{aligned}$$

where (i) is based on Lemma 6.3 in [4] and Lemma 5.4 in [46]. ■

Here, the effective dimension $\widetilde{d}$ measures the vanishing speed of $\boldsymbol{G}(0)$'s eigenvalues, and it is analogous to that of existing works on neural contextual bandits algorithms [4, 6, 46]. As $\widetilde{d}$ is smaller than the dimension of the gradient matrix $\boldsymbol{G}(0)$, 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 $\boldsymbol{G}^{(j)}(\boldsymbol{\Theta}_{L-1}) = \big(g(\widetilde{\boldsymbol{X}}_1;\boldsymbol{\Theta}_{L-1}^{(j)}),\ldots,g(\widetilde{\boldsymbol{X}}_T;\boldsymbol{\Theta}_{L-1}^{(j)})\big)^\intercal\big(g(\widetilde{\boldsymbol{X}}_1;\boldsymbol{\Theta}_{L-1}^{(j)}),\ldots,g(\widetilde{\boldsymbol{X}}_T;\boldsymbol{\Theta}_{L-1}^{(j)})\big)$ where $g(\widetilde{\boldsymbol{X}};\boldsymbol{\Theta}_{L-1})$ is the gradient vector w.r.t. $\boldsymbol{\Theta}_{L-1}$.

Assumption 5.6. *With width $m \geq Poly(T, L, \frac{1}{\beta_F}, \frac{1}{\lambda}, (\zeta\beta_L)^L, \log(\frac{1}{\delta}))$ and for $j \in [J]$, we have the minimal eigenvalue of $\boldsymbol{G}^{(j)}$ as*

$$\lambda_{\min}(\boldsymbol{G}^{(j)}(\boldsymbol{\Theta}_{L-1})) \geq \lambda_0/2$$

where λ_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 $j \in [J]$:*

$$\|\boldsymbol{F}_T^{(j)} - \boldsymbol{F}_T^{(j+1)}\|_2^2 \leq \frac{1}{4}\eta\beta_F \cdot \|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2$$

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, assume the model with width m defined in Lemma 5.3 is trained with the J-iterations GD on the contexts $\{\widetilde{\boldsymbol{X}}_\tau\}_{\tau=1}^T$ and rewards $\{r_\tau\}_{\tau=1}^T$. With probability at least $1-\delta$, a constant β_F such that $\beta_F \cdot \eta < 1$, set the network width $m \geq Poly(T, L, \frac{1}{\beta_F}, \frac{1}{\lambda}, (\zeta\beta_L)^L, \log(\frac{1}{\delta}))$ and the learning rate $\eta \leq O(T^{-1}L^{-1}\beta_h^{-2}(2\zeta\beta_L)^{-2L})$. Then, for any $j \in [J]$, we have*

$$\|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2 \leq (1-\beta_F\cdot\eta)^j \cdot \|\boldsymbol{F}_T^{(0)} - \boldsymbol{Y}_T\|_2^2$$

where the vector $\boldsymbol{F}^{(j)} = [f(\mathcal{G}_T, \widetilde{\boldsymbol{X}}_\tau; \boldsymbol{\Theta}^{(j)})]_{\tau=1}^T$, and $\boldsymbol{Y}_T = [r_\tau]_{\tau=1}^T$.

Proof. Following an approach analogous to [13], we apply and induction based method for the proof. The hypothesis is that $\|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2 \leq (1-\beta_F\cdot\eta)^j\cdot\|\boldsymbol{F}_T^{(0)} - \boldsymbol{Y}_T\|_2^2, j \in [J]$. With a similar procedure in Condition A.1 of [13], we have

$$\|\boldsymbol{F}_T^{(j+1)} - \boldsymbol{Y}_T\|_2^2 \leq \|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2 - 2\eta\|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_{\boldsymbol{G}^{(j)}}^2 - 2(\boldsymbol{Y}_T - \boldsymbol{F}_T^{(j)})^\intercal\boldsymbol{V}^{(j)} + \|\boldsymbol{F}_T^{(j+1)} - \boldsymbol{F}_T^{(j)}\|_2^2$$

with $\boldsymbol{V}^{(j)} = (\boldsymbol{V}^{(j)}(\widetilde{\boldsymbol{X}}_1),\ldots,\boldsymbol{V}^{(j)}(\widetilde{\boldsymbol{X}}_T))^\intercal$. For $\boldsymbol{\Theta}' \in \{\boldsymbol{\Theta}_{gnn},\ldots,\boldsymbol{\Theta}_{L-1}\}$,

$$|\boldsymbol{V}^{(j)}(\widetilde{\boldsymbol{X}})| = \eta \max_{0\leq s\leq\eta}\left[\sum_{\boldsymbol{\Theta}'}\|\nabla\mathcal{L}(\boldsymbol{\Theta}'^{(j)})\|_F\|\nabla f(\boldsymbol{\Theta}'^{(j)}) - \nabla f(\boldsymbol{\Theta}'^{(j)}, s)\|_F\right]$$

where $\nabla f(\boldsymbol{\Theta}'^{(j)}, s) = \nabla f\big(\boldsymbol{\Theta}'^{(j)} - s\cdot\nabla\mathcal{L}(\boldsymbol{\Theta}'^{(j)})\big)$. The notation $\mathcal{G}, \widetilde{\boldsymbol{X}}$ is omitted for simplicity. Then, based on the conclusions from Lemma C.1, Lemma 5.7 and Assumption 5.6, we can have

$$\|\boldsymbol{F}_T^{(j+1)} - \boldsymbol{Y}_T\|_2^2 \leq (1-\eta\lambda_0)\|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2 - 2(\boldsymbol{Y}_T - \boldsymbol{F}_T^{(j)})^\intercal\boldsymbol{V}^{(j)} + \|\boldsymbol{F}_T^{(j+1)} - \boldsymbol{F}_T^{(j)}\|_2^2 \leq (1-\frac{\eta\lambda_0}{2})\|\boldsymbol{F}_T^{(j)} - \boldsymbol{Y}_T\|_2^2$$

by setting $\beta_F = \lambda_0/2$. ■

This theorem shows that with sufficiently large m and proper η, the GD will converge to the global minimum at a linear rate, which is essential for proving the regret bound.

6 EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed framework by comparing its performances with state-of-the-art baselines through experiments on four real data sets. As linear algorithms have been outperformed in previous works [12, 45, 46], we will not include these linear methods in the experiments below. Our six baseline algorithms are:

- KMTL-UCB [12] estimates the "task similarities" with received contextual information. The estimations are based on a variant of kernel ridge regression.
- Kernel-Ind is Kernel-UCB [34] under the *"disjoint setting"* [28] where it learns individual estimators for each arm group.
- Kernel-Pool represents Kernel-UCB under the *"pooling setting"* where it applies a single estimator for all arm groups.
- Neural-TS stands for Neural Thompson Sampling [45] with group-aware embedding, which enables it to leverage the group information. It applies a neural network for exploitation and Thompson sampling strategy for exploration.
- Neural-Pool is for Neural-UCB [46] with a single neural network to evaluate the reward, and calculate the upper confidence bounds with the network gradients.
- Neural-Ind represents Neural-UCB with group-aware embedding for utilizing the group information.

Note that COFIBA [30] is naturally Kernel-Ind (with linear kernel) given the arm group information and one single user to serve, so we do not include it in our benchmarks. To find the best exploration parameter, we perform grid searches over the range $\{10^{-1}, 10^{-2}, 10^{-3}\}$ for all algorithms. Similarly, the learning rate for neural algorithms are chosen from $\{10^{-2}, 10^{-3}, 10^{-4}\}$. For Neural-UCB, Neural-TS and our reward estimation module, we apply a two-layer FC network with $m = 500$. RBF kernels are applied for KMTL-UCB and Kernel-UCB as well as our graph estimation module. Kernel-Pool and Neural-Pool will not fit into the multi-class classification setting, as we only receive one arm (context) at each time step without the arm group information.

6.1 Real Data Sets

Here, we compare our proposed model with baseline algorithms on four real data sets with different specifications.

MovieLens and Yelp data sets. The first real data set is the "MovieLens 20M rating data set" (grouplens.org/datasets/movielens/20m/) . To obtain the user features, we first choose 100 movies and 4000 users with **most reviews** to form the user-movie matrix where the entries are user ratings, and the user features $\boldsymbol{v}_u \in \mathbb{R}^d$ are obtained through singular value decomposition (SVD) where the

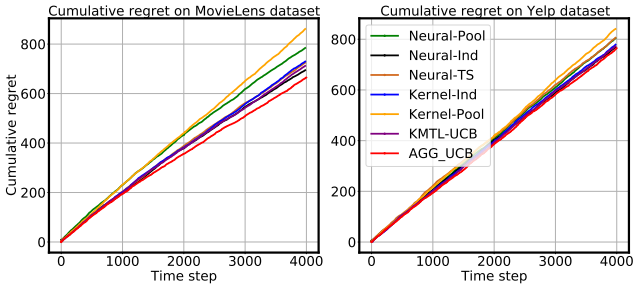

Figure 1: Cumulative regrets for recommendation data sets.

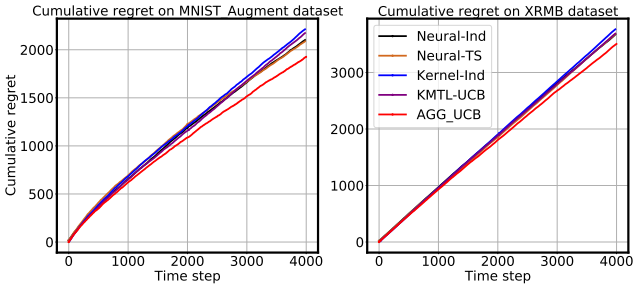

Figure 2: Cumulative regrets for classification data sets.

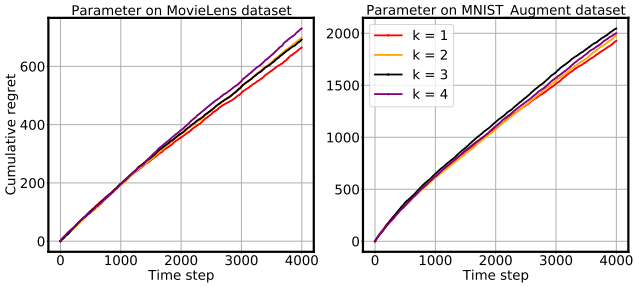

Figure 3: Cumulative regrets on MovieLens and MNIST-Aug data sets with different neighborhood parameter k.

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 $\boldsymbol{v}_i \in \mathbb{R}^d$ 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 $\widetilde{\boldsymbol{x}}_{c,t}^{(i)} = [\boldsymbol{v}_{u_t} \odot \boldsymbol{v}_i] \in \mathbb{R}^d, c \in C_t, i \in [n_{c,t}]$, and let $|\mathcal{X}_t| = 20$. Finally, we concatenate a constant 0.01 to each $\widetilde{\boldsymbol{x}}_{c,t}^{(i)}$ to obtain $\boldsymbol{x}_{c,t}^{(i)} \in \mathbb{R}^{d_x}$, which makes $d_x = 21$, before normalizing $\boldsymbol{x}_{c,t}^{(i)}$. 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 $\boldsymbol{x}_{c,t}^{(i)}$ ($d_x = 21, |\mathcal{X}_t| = 20$) 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 $\boldsymbol{x}_t$, 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 $\boldsymbol{x}_t$ 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 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 subclasses 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. Base 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 Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. *NeurIPS* 24 (2011), 2312–2320.
[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *ICML*. PMLR, 242–252.
[3] 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.
[4] Yikun Ban and Jingrui He. 2021. Convolutional neural bandit: Provable algorithm for visual-aware advertising. *arXiv preprint arXiv:2107.07438* (2021).
[5] Yikun Ban and Jingrui He. 2021. Local clustering in contextual multi-armed bandits. In *Proceedings of the Web Conference 2021*. 2335–2346.
[6] Yikun Ban, Jingrui He, and Curtiss B Cook. 2021. Multi-facet Contextual Bandits: A Neural Network Perspective. *arXiv preprint arXiv:2106.03039* (2021).
[7] Yikun Ban, Yunzhe Qi, Tianxin Wei, and Jingrui He. 2022. Neural Collaborative Filtering Bandits via Meta Learning. *ArXiv* abs/2201.13395 (2022).
[8] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. 2022. EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits. In *ICLR*.
[9] Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*. 2178–2186.
[10] Edward Chlebus. 2009. An approximate formula for a partial sum of the divergent p-series. *Applied Mathematics Letters* 22, 5 (2009), 732–737.
[11] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *AISTATS*. 208–214.
[12] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. 2017. Multi-task learning for contextual bandits. In *NeurIPS*. 4848–4856.
[13] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. 2019. Gradient descent finds global minima of deep neural networks. In *ICML*. PMLR, 1675–1685.
[14] Simon S Du, Kangcheng Hou, Barnabás Póczos, Ruslan Salakhutdinov, Ruosong Wang, and Keyulu Xu. 2019. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *arXiv preprint arXiv:1905.13192* (2019).
[15] 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*. PMLR, 67–82.
[16] Dongqi Fu and Jingrui He. 2021. DPPIN: A biological repository of dynamic protein-protein interaction network data. *arXiv preprint arXiv:2107.02168* (2021).
[17] Dongqi Fu and Jingrui He. 2021. SDG: A Simplified and Dynamic Graph Neural Network. In *SIGIR '21*. 2273–2277.
[18] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *ICML*. 1253–1262.
[19] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online clustering of bandits. In *ICML*. 757–765.
[20] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
[22] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[23] NT Hoang, Takanori Maehara, and Tsuyoshi Murata. 2021. Revisiting Graph Neural Networks: Graph Filtering Perspective. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 8376–8383.
[24] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572* (2018).
[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[26] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).
[27] Andreas Krause and Cheng Soon Ong. 2011. Contextual Gaussian Process Bandit Optimization.. In *NeurIPS*. 2447–2455.
[28] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*. 661–670.
[29] Shuai Li, Wei Chen, Shuai Li, and Kwong-Sak Leung. 2019. Improved Algorithm on Online Clustering of Bandits. In *IJCAI*. 2923–2929.
[30] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *SIGIR*. 539–548.
[31] Sandra Sajeev, Jade Huang, Nikos Karampatziakis, Matthew Hall, Sebastian Kochman, and Weizhu Chen. 2021. Contextual Bandit Applications in a Customer Support Bot. In *KDD '21*. 3522–3530.
[32] Xin Shao, Ning Lv, Jie Liao, Jinbo Long, Rui Xue, Ni Ai, Donghang Xu, and Xiaohui Fan. 2019. Copy number variation is highly correlated with differential gene expression: a pan-cancer study. *BMC medical genetics* 20, 1 (2019), 1–14.
[33] Sohini Upadhyay, Mikhail Yurochkin, Mayank Agarwal, Yasaman Khazaeni, et al. 2020. Online Semi-Supervised Learning with Bandit Feedback. *arXiv preprint arXiv:2010.12574* (2020).
[34] 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).
[35] Roman Vershynin. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027* (2010).
[36] Sofía S Villar, Jack Bowden, and James Wason. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics* 30, 2 (2015), 199.
[37] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. 2015. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE ICASSP*. IEEE.
[38] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. 2020. Fast adaptation for cold-start collaborative filtering with meta-learning. In *ICDM*. IEEE, 661–670.
[39] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *ICML*. PMLR, 6861–6871.
[40] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. 2016. Contextual bandits in a collaborative environment. In *SIGIR*. 529–538.
[41] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. 2019. Dynamic Ensemble of Contextual Bandits to Satisfy Users' Changing Interests. In *WWW*. 2080–2090.
[42] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*. PMLR, 5453–5462.
[43] Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. 2021. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *ICML*. PMLR, 11592–11602.
[44] Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *ICML*. PMLR, 7134–7143.
[45] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural thompson sampling. *arXiv preprint arXiv:2010.00827* (2020).
[46] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Contextual Bandits with UCB-based Exploration. arXiv:1911.04462 [cs.LG]
[47] Yao Zhou, Haonan Wang, Jingrui He, and Haixun Wang. 2021. From Intrinsic to Counterfactual: On the Explainability of Contextualized Recommender Systems. *ArXiv* (2021). arXiv:2110.14844

[48] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabadi, 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 $\widetilde{\boldsymbol{X}}_{c,t}^{(i)}$ (represented by $\widetilde{\boldsymbol{X}}$ for brevity). Supposing $\widetilde{\boldsymbol{X}}$ belongs to the arm group c, denote $\boldsymbol{h_A} = [\boldsymbol{A}_t^k \widetilde{\boldsymbol{X}}]_c$ as the corresponding row in matrix $\boldsymbol{A}_t^k \widetilde{\boldsymbol{X}}$ based on index of group c in C (if group c is the c'-th group in C, then $\boldsymbol{h_A}$ is the c'-th row in $\boldsymbol{A}_t^k \widetilde{\boldsymbol{X}}$). Similarly, we have $\boldsymbol{h}_{gnn} = [\boldsymbol{H}_{gnn}]_c$ and $\boldsymbol{h}_l = [\boldsymbol{H}_l]_c$ respectively. Given received contexts $\{\widetilde{\boldsymbol{X}}_\tau\}_{\tau=1}^T$ and rewards $\{r_\tau\}_{\tau=1}^T$, the gradient w.r.t. weight matrix $\boldsymbol{\Theta}_l, \forall l \in \{1, \ldots, L-1\}$ will be

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \boldsymbol{\Theta}_l} = m^{-\frac{L-l+1}{2}} \sum_{\tau=1}^{T} |f(\widetilde{\boldsymbol{X}}_\tau; \Theta) - r_\tau|^2 \left(\boldsymbol{h}_{l-1} \boldsymbol{\Theta}_L^\intercal \Big(\prod_{q=l+1}^{L-1} \boldsymbol{\Gamma}_q \boldsymbol{\Theta}_q^\intercal \Big) \boldsymbol{\Gamma}_l \right)$$

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

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \boldsymbol{\Theta}_{gnn}} = m^{-\frac{L+1}{2}} \sum_{\tau=1}^{T} |f(\widetilde{\boldsymbol{X}}_\tau; \Theta) - r_\tau|^2 \left(\boldsymbol{h_A} \boldsymbol{\Theta}_L^\intercal \Big(\prod_{q=2}^{L-1} \boldsymbol{\Gamma}_q \boldsymbol{\Theta}_q^\intercal \Big) \boldsymbol{\Gamma}_1 \boldsymbol{\Theta}_1^\intercal \boldsymbol{Q} \boldsymbol{\Gamma}_{gnn} \right)$$

where $\boldsymbol{\Gamma}_{gnn} = diag([\sigma'(\boldsymbol{h_A}\boldsymbol{\Theta}_{gnn})])$. $\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{I} \in \mathbb{R}^{m \times m} \\ \boldsymbol{0} \in \mathbb{R}^{(d'-m)\times m} \end{pmatrix} \in \mathbb{R}^{d' \times m}$. Given the same $\mathcal{G}_t$, we provide lemmas to bound the R_1 term of **Eq.** 9. For brevity, the subscript $\tau \in [T]$ and notation $\mathcal{G}_t$ are omitted below by default.

Lemma A.1. *Given the randomly initialized parameters* $\boldsymbol{\Theta}^{(0)} = \{\boldsymbol{\Theta}_{gnn}^{(0)}, \boldsymbol{\Theta}_1^{(0)}, \boldsymbol{\Theta}_2^{(0)}, \ldots, \boldsymbol{\Theta}_L^{(0)}\}$, *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*

$$\begin{aligned}
&\|\boldsymbol{\Theta}_{gnn}^{(0)}\|_2 \le \beta_1\sqrt{m},\ \|\boldsymbol{\Theta}_1^{(0)}\|_2 \le \beta_2\sqrt{m},\ \|\boldsymbol{\Theta}_L^{(0)}\|_2 \le \beta_3,\\
&\|\boldsymbol{h}_{gnn}^{(0)}\|_2 \le \zeta \cdot \beta_1, \quad \|\boldsymbol{h}_1^{(0)}\|_2 \le \zeta \cdot \beta_2 + \zeta^2 \cdot \beta_1\beta_2,\\
&|f(\widetilde{\boldsymbol{X}}; \boldsymbol{\Theta}^{(0)})| \le \zeta \cdot \beta_3 \cdot (\zeta \cdot \beta_4)^{L-2}(\zeta \cdot \beta_2 + \zeta^2 \cdot \beta_1\beta_2)/\sqrt{m},\\
&\|\boldsymbol{\Theta}_l^{(0)}\|_2 \le \beta_4\sqrt{m},\ \|\boldsymbol{h}_l^{(0)}\|_2 \le (\zeta \cdot \beta_4)^{l-1}(\zeta \cdot \beta_2 + \zeta^2 \cdot \beta_1\beta_2),\\
&\forall l \in \{2, \ldots, L-1\}.
\end{aligned}$$

Proof. Based on the properties of random Gaussian matrices [4, 13, 35], with the probability of at least $1 - e^{-\frac{(\beta_1 - \sqrt{d_{\widetilde{x}}/m}-1)^2 \cdot m}{2}} = 1 - e^{-\Omega(m)}$, we have

$$\|\boldsymbol{\Theta}_{gnn}^{(0)}\|_2 \le \beta_1\sqrt{m}$$

where $\beta_1 \ge \sqrt{d_{\widetilde{x}}/m} + 1$ with $m > d_{\widetilde{x}}$. Applying the analogous approach for the other randomly initialized matrices would give similar bounds. Regarding the nature of $\boldsymbol{A}$, we can easily have $\|\boldsymbol{h_A}\|_2 \le 1$. Then,

$$\|\boldsymbol{h}_{gnn}^{(0)}\|_2 = m^{-\frac{1}{2}}\|\sigma(\boldsymbol{h_A} \cdot \boldsymbol{\Theta}_{gnn})\|_2 \le \zeta m^{-\frac{1}{2}} \cdot \|\boldsymbol{h_A}\|_2 \|\boldsymbol{\Theta}_{gnn}\|_2 \le \zeta \cdot \beta_1$$

due to the assumed ζ-Lipschitz continuity. Denoting the concatenated input for reward estimation module as $\boldsymbol{x'} = [\boldsymbol{h}_{gnn}^{(0)}; \widetilde{\boldsymbol{X}}]_c \in \mathbb{R}^{1 \times (d_{\widetilde{x}}+m)}$, we can easily derive that $\|\boldsymbol{x'}\|_2 \le \zeta \cdot \beta_1 + 1$. Thus,

$$\begin{aligned}
\|\boldsymbol{h}_1^{(0)}\|_2 &= m^{-\frac{1}{2}}\|\sigma(\boldsymbol{x'} \cdot \boldsymbol{\Theta}_1)\|_2 \le \zeta m^{-\frac{1}{2}} \cdot \|\boldsymbol{x'}\|_2 \|\boldsymbol{\Theta}_1\|_2\\
&\le \zeta \cdot \beta_2(\zeta \cdot \beta_1 + 1) = \zeta \cdot \beta_2 + \zeta^2 \cdot \beta_1\beta_2.
\end{aligned}$$

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* $\|\boldsymbol{h}_l^{(j)} - \boldsymbol{h}_l^{(0)}\|_2 \le \Lambda^{(j)}, \forall j \in [J]$. *With the probability of at least* $1 - O(TL) \cdot e^{-\Omega(m)}$ *and* $\boldsymbol{\Theta} \in \{\boldsymbol{\Theta}_{gnn}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_L\}$, *we have*

$$\|\boldsymbol{\Theta}^{(j)} - \boldsymbol{\Theta}^{(0)}\|_F \le \Upsilon/\sqrt{m}$$

where $\Upsilon = \frac{2\sqrt{2}t}{\beta_F}(\beta_h + \Lambda^{(j)})(\beta_L + 1)^L\zeta^L$.

Proof. We prove this Lemma following an induction-based procedure [4, 13]. The hypothesis is $\|\boldsymbol{\Theta}^{(0)} - \boldsymbol{\Theta}^{(j)}\|_F \le \Upsilon/\sqrt{m}, \forall \boldsymbol{\Theta} \in \{\boldsymbol{\Theta}_{gnn}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_L\}$, and let $\beta_L = \max\{\beta_1, \beta_2, \beta_3, \beta_4\}$. According to **Algorithm** 2, we have for the $j+1$-th iteration and $l \in [L]$,

$$\begin{aligned}
&\|\boldsymbol{\Theta}_l^{(j+1)} - \boldsymbol{\Theta}_l^{(j)}\|_F = m^{-\frac{L-l+1}{2}}\eta \cdot \|\sum_{\tau=1}^{T} |f(\widetilde{\boldsymbol{X}}_\tau; \boldsymbol{\Theta}^{(j)}) - r_\tau|^2 \cdot \boldsymbol{h}_{l-1}^{(j)} (\boldsymbol{\Theta}_L^{(j)})^\intercal\\
&\quad \cdot \Big(\prod_{q=l+1}^{L-1} \boldsymbol{\Gamma}_q^{(j)} \cdot (\boldsymbol{\Theta}_q^{(j)})^\intercal \Big) \cdot \boldsymbol{\Gamma}_l^{(j)}\|_F\\
&\le m^{-\frac{L-l+1}{2}}\eta\sqrt{t} \cdot \|\boldsymbol{F}_t^{(j)} - \boldsymbol{Y}_t\|_2 \|\boldsymbol{h}_{l-1}^{(j)}\boldsymbol{\Theta}_L^{(j)}\|_F \prod_{q=l+1}^{L-1} \|\boldsymbol{\Theta}_q^{(j)}\|_2 \prod_{q=l}^{L-1} \|\boldsymbol{\Gamma}_q^{(j)}\|_2\\
&\le m^{-\frac{L-l+1}{2}}\eta\sqrt{t} \cdot \|\boldsymbol{F}_t^{(j)} - \boldsymbol{Y}_t\|_2 \|\boldsymbol{h}_{l-1}^{(j)}\|_2 \|\boldsymbol{\Theta}_L^{(j)}\|_2 \prod_{q=l+1}^{L-1} \|\boldsymbol{\Theta}_q^{(j)}\|_2 \prod_{q=l}^{L-1} \|\boldsymbol{\Gamma}_q^{(j)}\|_2
\end{aligned}$$

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

$$\begin{aligned}
\prod_{q=l+1}^{L-1} \|\boldsymbol{\Theta}_q^{(j)}\|_2 &\le \prod_{q=l+1}^{L-1} \Big(\|\boldsymbol{\Theta}_q^{(0)}\|_2 + \|\boldsymbol{\Theta}_q^{(j)} - \boldsymbol{\Theta}_q^{(0)}\|_2 \Big)\\
&\le (\beta_L\sqrt{m} + \Upsilon/\sqrt{m})^{L-l-1};
\end{aligned}$$

while for $\|\boldsymbol{\Gamma}_q^{(j)}\|_2$, we have $\prod_{q=l}^{L-1} \|\boldsymbol{\Gamma}_q^{(j)}\|_2 \le \zeta^{L-l}$. Combining all the results above and based on Lemma 5.8, it means that for $l \in [L]$,

$$\begin{aligned}
&\|\boldsymbol{\Theta}_l^{(j+1)} - \boldsymbol{\Theta}_l^{(j)}\|_F \le m^{-\frac{L-l+1}{2}}\eta\sqrt{t} \cdot (1 - \beta_F \cdot \eta)^{j/2} \cdot \|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2\\
&\quad \cdot \|\boldsymbol{h}_{l-1}^{(j)}\|_2 \cdot \|\boldsymbol{\Theta}_L^{(j)}\|_2 \cdot (\beta_L\sqrt{m} + \Upsilon/\sqrt{m})^{L-l-1} \cdot \zeta^{L-l+1}\\
&\le m^{-\frac{1}{2}}(1 - \beta_F\eta)^{j/2}\eta\sqrt{t}\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2((\beta_h + \Lambda^{(j)})(\beta_L + \Upsilon/m)^{L-l}\zeta^{L-l}
\end{aligned}$$

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

$$\|\boldsymbol{\Theta}_l^{(j+1)} - \boldsymbol{\Theta}_l^{(0)}\|_F \le \frac{2\sqrt{t}}{\beta_F\sqrt{m}}\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2(\beta_h + \Lambda^{(j)})(\beta_L + \Upsilon/m)^{L-l}\zeta^{L-l}.$$

For the last layer Θ_L, the conclusion can be verified through a similar procedure. Analogously, for Θ_{gnn}, we have

$$\begin{aligned}
&\|\Theta_{gnn}^{(j+1)} - \Theta_{gnn}^{(j)}\|_F \\
&= m^{-\frac{L+1}{2}}\eta\|\sum_{\tau=1}^{T}|f(\widetilde{X}_\tau;\Theta) - r_\tau|^2 \cdot \left(\boldsymbol{h}_{\boldsymbol{A}} \cdot \Theta_L^\top \cdot (\prod_{q=2}^{L-1}\Gamma_q \cdot \Theta_q^\top)\cdot \Gamma_1\Theta_1^\top\right) \\
&\qquad \cdot \boldsymbol{Q}\cdot\Gamma_{gnn}\|_F \\
&\le m^{-\frac{L+1}{2}}\eta\sqrt{t}(1-\beta_F\cdot\eta)^{j/2}\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2\|\boldsymbol{h}_{\boldsymbol{A}}\|_2\|\prod_{q=1}^{L}\Theta_q\|_2\zeta^L\|\boldsymbol{Q}\|_2 \\
&\le \sqrt{t}m^{-\frac{1}{2}}(1-\beta_F\cdot\eta)^{j/2}\eta\cdot\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2\cdot\zeta^L\cdot(\beta_L+\Upsilon/m)^L,
\end{aligned}$$

which leads to

$$\|\Theta_{gnn}^{(j+1)} - \Theta_{gnn}^{(0)}\|_F \le \frac{2}{\beta_F}\sqrt{t}m^{-\frac{1}{2}}\cdot\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2\cdot\zeta^L\cdot(\beta_L+\Upsilon/m)^L.$$

Since $\|\boldsymbol{F}_t^{(0)} - \boldsymbol{Y}_t\|_2 \le \sqrt{2t}$ (Lemma 5.8) and $\Upsilon/m \le 1$ 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 $\beta_h' = \max\{\zeta\cdot\beta_1, \zeta\cdot\beta_2+\zeta^2\cdot\beta_1\beta_2, (\zeta\cdot\beta_4)^{L-2}(\zeta\cdot\beta_2+\zeta^2\cdot\beta_1\beta_2)\}$ and $\beta_h = \max\{\zeta\cdot\beta_L+1, \beta_h'\}$. With $\boldsymbol{h}\in\{\boldsymbol{h}_{gnn}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{L-1}\}$, we have*

$$\|\boldsymbol{h}^{(j)} - \boldsymbol{h}^{(0)}\|_2 \le \frac{\zeta\Upsilon}{m}\cdot\beta_h\cdot\frac{(2\zeta\beta_L)^L - 1}{2\zeta\beta_L - 1} = \Lambda^{(j)},\ \|\boldsymbol{h}^{(j)}\|_2 \le \beta_h + \Lambda^{(j)}$$

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

$$\begin{aligned}
&\|\boldsymbol{h}_l^{(j)} - \boldsymbol{h}_l^{(0)}\|_2 = \sqrt{\frac{1}{m}}\|\sigma(\boldsymbol{h}_{l-1}^{(j)}\cdot\Theta_l^{(j)}) - \sigma(\boldsymbol{h}_{l-1}^{(0)}\cdot\Theta_l^{(0)})\|_2 \\
&\le \sqrt{\frac{1}{m}}\zeta\cdot\left(\|\boldsymbol{h}_{l-1}^{(j)}\cdot\Theta_l^{(j)} - \boldsymbol{h}_{l-1}^{(0)}\cdot\Theta_l^{(j)}\|_2 + \|\boldsymbol{h}_{l-1}^{(0)}\cdot\Theta_l^{(j)} - \boldsymbol{h}_{l-1}^{(0)}\cdot\Theta_l^{(0)}\|_2\right) \\
&\le \sqrt{\frac{1}{m}}\zeta\cdot(\|\Theta_l^{(0)}\|_2 + \|\Theta_l^{(j)} - \Theta_l^{(0)}\|_F)\cdot\|\boldsymbol{h}_{l-1}^{(j)} - \boldsymbol{h}_{l-1}^{(0)}\|_2 + \\
&\quad \sqrt{\frac{1}{m}}\zeta\cdot\|\boldsymbol{h}_{l-1}^{(0)}\|_2\cdot\|\Theta_l^{(j)} - \Theta_l^{(0)}\|_F \\
&\le \sqrt{\frac{1}{m}}\zeta\cdot(\beta_L\sqrt{m} + \Upsilon/\sqrt{m})\cdot\|\boldsymbol{h}_{l-1}^{(j)} - \boldsymbol{h}_{l-1}^{(0)}\|_2 + \zeta\cdot\beta_h'\cdot\Upsilon/m \\
&\le \zeta\cdot(\beta_L+\Upsilon/m)\cdot\zeta\frac{\Upsilon}{m}\cdot\Lambda_{l-1}^{(j)} + \zeta\cdot\beta_h'\cdot\Upsilon/m \\
&\le \zeta\frac{\Upsilon}{m}\cdot(\beta_h + 2\zeta\beta_L\cdot\Lambda_{l-1}^{(j)}) = \zeta\frac{\Upsilon}{m}\cdot\Lambda_l^{(j)}
\end{aligned}$$

where the last two inequalities are derived by applying **Lemma** A.2 and the hypothesis. For the aggregation module output $\boldsymbol{h}_{gnn}^{(0)}$,

$$\begin{aligned}
&\|\boldsymbol{h}_{gnn}^{(j)} - \boldsymbol{h}_{gnn}^{(0)}\|_2 = \sqrt{\frac{1}{m}}\|\sigma(\Theta_{gnn}^{(j)}\cdot\boldsymbol{h}_S) - \sigma(\Theta_{gnn}^{(0)}\cdot\boldsymbol{h}_S)\|_2 \\
&\le \frac{\zeta}{\sqrt{m}}\|\Theta_{gnn}^{(j)} - \Theta_{gnn}^{(0)}\|_F\cdot\|\boldsymbol{h}_S\|_2 \le \frac{\zeta\Upsilon}{m}\beta_h.
\end{aligned}$$

Then, for the first layer $l=1$, we have

$$\begin{aligned}
&\|\boldsymbol{h}_1^{(j)} - \boldsymbol{h}_1^{(0)}\|_2 = \sqrt{\frac{1}{m}}\|\sigma(\boldsymbol{x}'\cdot\Theta_1^{(j)}) - \sigma(\boldsymbol{x}'\cdot\Theta_1^{(0)})\|_2 \\
&\le \frac{\zeta}{\sqrt{m}}\|\Theta_{gnn}^{(j)} - \Theta_1^{(0)}\|_F\cdot\|\boldsymbol{x}'\|_2 \le \frac{\zeta\Upsilon}{m}\cdot(\zeta\cdot\beta_L+1) \le \frac{\zeta\Upsilon}{m}\cdot\beta_h.
\end{aligned}$$

Combining all the results, for $\boldsymbol{h}\in\{\boldsymbol{h}_{gnn}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{L-1}\}$, it has

$$\|\boldsymbol{h}^{(j)} - \boldsymbol{h}^{(0)}\|_2 \le \frac{\zeta\Upsilon}{m}\cdot\beta_h\cdot\frac{(2\zeta\beta_L)^L - 1}{2\zeta\beta_L - 1} = \Lambda^{(j)},$$

which completes the proof.

Lemma A.4. *With initialized network parameters Θ and the probability of at least $1-O(TL)\cdot e^{-\Omega(m)}$, we have*

$$\|\nabla_{\Theta}f(\widetilde{X};\Theta^{(0)})\|_F \le \beta_h\beta_3\cdot(\beta_L\zeta)^L/m, \|\nabla_{\Theta_L}f(\widetilde{X};\Theta^{(0)})\|_F \le \beta_h/\sqrt{m},$$

and the norm of gradient difference

$$\begin{aligned}
&\|\nabla_{\Theta}f(\widetilde{X};\Theta^{(0)}) - \nabla_{\Theta}f(\widetilde{X};\Theta^{(j)})\|_F \le 3\cdot\Lambda^{(j)}, \\
&\|\nabla_{\Theta_L}f(\widetilde{X};\Theta^{(0)}) - \nabla_{\Theta_L}f(\widetilde{X};\Theta^{(j)})\|_F \le \Lambda^{(j)}/\sqrt{m}.
\end{aligned}$$

with $\Theta\in\{\Theta_{gnn}, \Theta_1, \ldots, \Theta_{L-1}\}$.

Proof. First, for $l\in[L-1]$, we have

$$\begin{aligned}
&\|\nabla_{\Theta_l}f(\widetilde{X};\Theta^{(0)})\|_F = m^{-\frac{L-l+1}{2}}\|\boldsymbol{h}_{l-1}^{(0)}(\Theta_L^{(0)}(\prod_{q=l+1}^{L-1}\Gamma_q\cdot\Theta_q^{(0)})\cdot\Gamma_l)\|_F \\
&\le m^{-\frac{L-l+1}{2}}\cdot\|\boldsymbol{h}_{l-1}^{(0)}\Theta_L^{(0)}\|_F\cdot\|\prod_{q=l}^{L-1}\Gamma_q\|_2\cdot\|\prod_{q=l+1}^{L-1}\Theta_q^{(0)}\|_2 \\
&\le m^{-\frac{L-l+1}{2}}\cdot\|\boldsymbol{h}_{l-1}^{(0)}\|_2\cdot\|\Theta_L^{(0)}\|_2\cdot\|\prod_{q=l}^{L-1}\Gamma_q\|_2\cdot\|\prod_{q=l+1}^{L-1}\Theta_q^{(0)}\|_2 \\
&\le m^{-\frac{L-l+1}{2}}\cdot\beta_h\beta_3\cdot\zeta^{L-l}\cdot(\beta_L\sqrt{m})^{L-l-1} \le \beta_h\beta_3\cdot(\beta_L\zeta)^L/m.
\end{aligned}$$

For Θ_{gnn}, we can also derive similar results. For Θ_L,

$$\|\nabla_{\Theta_L}f(\widetilde{X};\Theta^{(0)})\|_F = m^{-0.5}\cdot\|\boldsymbol{h}_{L-1}^{(0)}\|_2 \le \beta_h/\sqrt{m}$$

Then, with $\nabla_l^{(j)} = m^{-\frac{L-l+1}{2}}\cdot(\Theta_L^{(j)})^\top\cdot\left(\prod_{q=l+1}^{L-1}\Gamma_q\cdot(\Theta_q^{(j)})^\top\right)\cdot\Gamma_l$, we have the norm of gradient difference

$$\begin{aligned}
&\|\nabla_{\Theta_l}f(\widetilde{X};\Theta^{(j)}) - \nabla_{\Theta_l}f(\widetilde{X};\Theta^{(0)})\|_F = \|\boldsymbol{h}_{l-1}^{(0)}\cdot\nabla_l^{(0)} - \boldsymbol{h}_{l-1}^{(j)}\cdot\nabla_l^{(j)}\|_F \\
&\le \|\boldsymbol{h}_{l-1}^{(0)}\cdot\nabla_l^{(0)} - \boldsymbol{h}_{l-1}^{(j)}\cdot\nabla_l^{(0)}\|_F + \|\boldsymbol{h}_{l-1}^{(j)}\cdot\nabla_l^{(0)} - \boldsymbol{h}_{l-1}^{(j)}\cdot\nabla_l^{(j)}\|_F \\
&\le \|\boldsymbol{h}_{l-1}^{(j)}\|_F\cdot\|\nabla_l^{(0)} - \nabla_l^{(j)}\|_F + \|\nabla_l^{(0)}\|_F\cdot\|\boldsymbol{h}_{l-1}^{(0)} - \boldsymbol{h}_{l-1}^{(j)}\|_F \\
&\le (\beta_h + \Lambda^{(j)})\cdot\|\nabla_l^{(0)} - \nabla_l^{(j)}\|_F + \Lambda^{(j)}\cdot\|\nabla_l^{(0)}\|_F.
\end{aligned}$$

Here, for the difference of ∇, we have

$$\begin{aligned}
&\|\nabla_l^{(0)} - \nabla_l^{(j)}\|_F \\
&= m^{-\frac{L-l+1}{2}}\|\Theta_L^{(0)}\cdot(\prod_{q=l+1}^{L-1}\Gamma_q\cdot\Theta_q^{(0)})\Gamma_l - \Theta_L^{(j)}(\prod_{q=l+1}^{L-1}\Gamma_q\Theta_q^{(j)})\Gamma_l\|_F \\
&= m^{-\frac{1}{2}}\cdot\|\nabla_{l+1}^{(0)}\Gamma_l\cdot\Theta_{l+1}^{(0)} - \nabla_{l+1}^{(j)}\Gamma_l\cdot\Theta_{l+1}^{(j)}\|_F \\
&\le \frac{\zeta}{\sqrt{m}}\cdot(\|\nabla_{l+1}^{(0)}\cdot\Theta_{l+1}^{(0)} - \nabla_{l+1}^{(0)}\cdot\Theta_{l+1}^{(j)}\|_F + \|\nabla_{l+1}^{(0)}\cdot\Theta_{l+1}^{(j)} - \nabla_{l+1}^{(j)}\cdot\Theta_{l+1}^{(j)}\|_F) \\
&\le \frac{\zeta}{\sqrt{m}}\cdot(\|\nabla_{l+1}^{(0)}\|_F\|\cdot\Theta_{l+1}^{(0)} - \Theta_{l+1}^{(j)}\|_F + \|\Theta_{l+1}^{(j)}\|_F\|\nabla_{l+1}^{(0)} - \nabla_{l+1}^{(j)}\|_F).
\end{aligned}$$

To continue the proof, we need to bound the term $\|\nabla_l^{(0)}\|_F$ as

$$\|\nabla_l^{(0)}\|_F = m^{-0.5}\|\mathbf{\Gamma}_l\mathbf{\Theta}_{l+1}^{(0)} \cdot \nabla_{l+1}^{(0)}\|_F \le \zeta\beta_L \cdot \|\nabla_{l+1}^{(0)}\|_F.$$

Since for $l = L-1$ we have

$$\|\nabla_{L-1}^{(0)}\|_F \le \frac{\zeta\beta_3}{m},$$

we can derive

$$\|\nabla_l^{(0)}\|_F \le \frac{\beta_3}{m} \cdot (\zeta \cdot \beta_L)^L \le 1$$

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

$$\|\nabla_L^{(0)} - \nabla_L^{(j)}\|_F = m^{-0.5}\|\boldsymbol{h}_{L-1}^{(0)} - \boldsymbol{h}_{L-1}^{(j)}\|_F \le \Lambda^{(j)}/\sqrt{m}$$

Therefore, we have

$$\|\nabla_l^{(0)} - \nabla_l^{(j)}\|_F \le \frac{\zeta\Upsilon}{m} + \zeta \cdot (\beta_L + \Upsilon/m)\|\nabla_{l+1}^{(0)} - \nabla_{l+1}^{(j)}\|_F.$$

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

$$\|\nabla_l^{(0)} - \nabla_l^{(j)}\|_F \le \frac{\zeta\Upsilon}{m} \cdot \frac{(2\zeta\beta_L)^L - 1}{2\zeta\beta_L - 1} = \frac{\Lambda^{(j)}}{\beta_h}.$$

Therefore, we will have

$$\begin{aligned}\|\nabla_{\mathbf{\Theta}_l} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \nabla_{\mathbf{\Theta}_l} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_F &\le \big(\beta_h + \Lambda^{(j)}\big) \cdot \frac{\Lambda^{(j)}}{\beta_h} + \Lambda^{(j)} \\ &\le \frac{\Lambda^{(j)}}{\beta_h} \cdot (2\beta_h + 1) = \Lambda^{(j)} \cdot (2 + \frac{1}{\beta_h}) \le 3 \cdot \Lambda^{(j)}\end{aligned}$$

with sufficiently large m. This bound can also be derived for $\|\nabla_{\mathbf{\Theta}_{gnn}} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2$ with a similar procedure. For L-th layer, we have

$$\begin{aligned}\|\nabla_{\mathbf{\Theta}_L} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \nabla_{\mathbf{\Theta}_L} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_F &\le m^{-0.5} \cdot \|\boldsymbol{h}_{L-1}^{(0)} - \boldsymbol{h}_{L-1}^{(j)}\|_F \\ &\le \Lambda^{(j)}/\sqrt{m},\end{aligned}$$

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*

$$\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2 \le m^{-1}\beta_h \cdot \sqrt{L \cdot \beta_3^2 \cdot (\beta_L\zeta)^{2L} + m},$$

$$\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2 \le \Lambda^{(j)} \cdot \sqrt{9L + m^{-1}} + m^{-1}\beta_h \cdot \sqrt{L \cdot \beta_3^2 \cdot (\beta_L\zeta)^{2L} + m}$$

$$\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}) - g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2 \le \Lambda^{(j)} \cdot \sqrt{9L + m^{-1}}.$$

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

$$\begin{aligned}\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2 &= \sqrt{\|\nabla_{\mathbf{\Theta}_{gnn}} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2^2 + \sum_{l=1}^{L}\|\nabla_{\mathbf{\Theta}_l} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2^2} \\ &\le m^{-1}\beta_h \cdot \sqrt{L \cdot \beta_3^2 \cdot (\beta_L\zeta)^{2L} + m}.\end{aligned}$$

Then, for the norm of gradients, $\mathbf{\Theta} \in \{\mathbf{\Theta}_{gnn}, \mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_{L-1}\}$, we have

$$\begin{aligned}&\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}) - g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2 \\ &= \sqrt{\sum_{\mathbf{\Theta}}\|\nabla_{\mathbf{\Theta}} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}) - \nabla_{\mathbf{\Theta}} f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2^2} \\ &\le \sqrt{9L \cdot (\Lambda^{(j)})^2 + (\Lambda^{(j)})^2/m} = \Lambda^{(j)} \cdot \sqrt{9L + m^{-1}}.\end{aligned}$$

Then, for the network gradient after GD, we have

$$\begin{aligned}\|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2 &\le \|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}) - g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)})\|_2 + \|g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)})\|_2 \\ &\le \Lambda^{(j)} \cdot \sqrt{9L + m^{-1}} + m^{-1}\beta_h \cdot \sqrt{L \cdot \beta_3^2 \cdot (\beta_L\zeta)^{2L} + m}\end{aligned}$$

Lemma A.6. *With the probability of at least* $1 - O(TL) \cdot e^{-\Omega(m)}$, *for the initialized parameter* $\mathbf{\Theta}^{(0)}$, *we have*

$$\begin{aligned}&|f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \langle g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}), \mathbf{\Theta}^{(j)} - \mathbf{\Theta}^{(0)}\rangle| \\ &\qquad \le m^{-0.5} \cdot \big(\Lambda^{(j)}(1 + \beta_3) + \beta_3\beta_h + L \cdot \beta_h\Upsilon\big),\end{aligned}$$

and for the network parameter after GD, $\mathbf{\Theta}^{(j)}$, *we have*

$$\begin{aligned}&|f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \langle g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}), \mathbf{\Theta}^{(j)} - \mathbf{\Theta}^{(0)}\rangle| \le B_3 \\ &\quad = m^{-0.5}\big(\beta_3(\Lambda^{(j)} + \beta_h) + L \cdot \Upsilon \cdot (\Lambda^{(j)} + \beta_h)(\Lambda^{(j)}/\beta_h + 1)\big).\end{aligned}$$

Proof. For the sake of enumeration, we let $\mathbf{\Theta}_0 = \mathbf{\Theta}_{gnn}, \nabla_0 = \nabla_{gnn}, \boldsymbol{h}_0 = \boldsymbol{h}_{gnn}$ and $\boldsymbol{h}_{-1} = \boldsymbol{h}_S$. Then, we can derive

$$\begin{aligned}&|f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \langle g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(0)}), \mathbf{\Theta}^{(j)} - \mathbf{\Theta}^{(0)}\rangle| = |\frac{1}{\sqrt{m}}\langle \boldsymbol{h}_{L-1}^{(j)}, \mathbf{\Theta}_L^{(j)}\rangle \\ &\qquad - \frac{1}{\sqrt{m}}\langle \boldsymbol{h}_{L-1}^{(0)}, \mathbf{\Theta}_L^{(0)} - \mathbf{\Theta}_L^{(j)}\rangle - \sum_{l=0}^{L-1}(\boldsymbol{h}_{l-1}^{(0)})^\intercal(\mathbf{\Theta}_l^{(0)} - \mathbf{\Theta}_l^{(j)})\nabla_l^{(0)})| \\ &\le m^{-0.5}\|\boldsymbol{h}_L^{(j)} - \boldsymbol{h}_L^{(0)}\|_2\|\mathbf{\Theta}_L^{(j)}\|_2 + m^{-0.5}\|\boldsymbol{h}_{L-1}^{(0)}\|_2\|\mathbf{\Theta}_L^{(0)}\|_2 \\ &\qquad + \sum_{l=0}^{L-1}\|\boldsymbol{h}_{l-1}^{(0)}\|_2\|\mathbf{\Theta}_l^{(0)} - \mathbf{\Theta}_l^{(j)}\|_F\|\nabla_l^{(0)}\|_F \\ &\le m^{-0.5}\Lambda^{(j)}(\Upsilon/\sqrt{m} + \beta_3) + m^{-0.5}\beta_3\beta_h + L \cdot \beta_h\frac{\Upsilon}{\sqrt{m}} \\ &\le m^{-0.5} \cdot \big(\Lambda^{(j)}(1 + \beta_3) + \beta_3\beta_h + L \cdot \beta_h\Upsilon\big).\end{aligned}$$

On the other hand, for network parameter after GD, we can have

$$\begin{aligned}&|f(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}) - \langle g(\widetilde{\boldsymbol{X}};\mathbf{\Theta}^{(j)}), \mathbf{\Theta}^{(j)} - \mathbf{\Theta}^{(0)}\rangle| = |\frac{1}{\sqrt{m}}\langle \boldsymbol{h}_{L-1}^{(j)}, \mathbf{\Theta}_L^{(j)}\rangle \\ &\qquad - \frac{1}{\sqrt{m}}\langle \boldsymbol{h}_{L-1}^{(j)}, \mathbf{\Theta}_L^{(j)} - \mathbf{\Theta}_L^{(0)}\rangle - \sum_{l=0}^{L-1}(\boldsymbol{h}_{l-1}^{(j)})^\intercal(\mathbf{\Theta}_l^{(0)} - \mathbf{\Theta}_l^{(j)})\nabla_l^{(j)})| \\ &\le |m^{-0.5}\langle \boldsymbol{h}_{L-1}^{(j)}, \mathbf{\Theta}_L^{(0)}\rangle - \sum_{l=0}^{L-1}(\boldsymbol{h}_{l-1}^{(j)})^\intercal(\mathbf{\Theta}_l^{(0)} - \mathbf{\Theta}_l^{(j)})\nabla_l^{(j)}| \\ &\le m^{-0.5}\|\boldsymbol{h}_{L-1}^{(j)}\|_2\|\mathbf{\Theta}_L^{(0)}\|_2 + \sum_{l=0}^{L-1}\|\boldsymbol{h}_{l-1}^{(j)}\|_2\|\mathbf{\Theta}_l^{(0)} - \mathbf{\Theta}_l^{(j)}\|_F\|\nabla_l^{(j)}\|_F \\ &\le m^{-0.5}\beta_3(\Lambda^{(j)} + \beta_h) + L \cdot (\Lambda^{(j)} + \beta_h)(\Upsilon/\sqrt{m})(\Lambda^{(j)}/\beta_h + 1) \\ &\le m^{-0.5}\big(\beta_3(\Lambda^{(j)} + \beta_h) + L \cdot \Upsilon \cdot (\Lambda^{(j)} + \beta_h)(\Lambda^{(j)}/\beta_h + 1)\big).\end{aligned}$$

This completes the proof.

Proof sketch for Lemmas A.1-A.6. First we derive the conclusions in Lemma A.1 with the property of Gaussian matrices. Then, Lemmas A.2 and A.3 are proved through the induction after breaking the target into norms of individual terms (variables, weight matrices) and applying Lemma A.1. Finally, for Lemmas A.4-A.6, we also decompose targets into norms of individual terms. Then, applying Lemmas A.1-A.3 the to bound these terms (at random initialization / after GD) would give the result. ■

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 $\widehat{\boldsymbol{\Theta}}$ as $\widehat{\boldsymbol{\Theta}}_0 = (Z_0)^{-1} \cdot \boldsymbol{b}_0$ where $Z_0 = \lambda \boldsymbol{I} + \frac{1}{m}\sum_{\tau=1}^{t} g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}_0) g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}_0)^\intercal, \boldsymbol{b}_0 = \frac{1}{\sqrt{m}}\sum_{\tau=1}^{t} r_\tau \cdot g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}_0)$. We also define the gradient matrix w.r.t. the network parameters as

$$
\begin{aligned}
&\boldsymbol{G}^{(j)} = \big(g(\widetilde{\boldsymbol{X}}_1;\boldsymbol{\Theta}^{(j)}),\ldots,g(\widetilde{\boldsymbol{X}}_t;\boldsymbol{\Theta}^{(j)})\big)\\
&\boldsymbol{f}^{(j)} = \big(f(\widetilde{\boldsymbol{X}}_1;\boldsymbol{\Theta}^{(j)}),\ldots,f(\widetilde{\boldsymbol{X}}_t;\boldsymbol{\Theta}^{(j)})\big), \quad \boldsymbol{r} = (r_1,\ldots,r_t)\\
&\boldsymbol{\Theta}^{(j+1)} = \boldsymbol{\Theta}^{(j)} - \eta\cdot\big((\boldsymbol{G}^{(j)})^\intercal(\boldsymbol{f}^{(j)} - \boldsymbol{r})\big).
\end{aligned}
$$

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

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

$$
\begin{aligned}
&\|\boldsymbol{G}^{(0)}\|_F \le G_1 = m^{-1}\beta_h\cdot\sqrt{t\cdot(L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m)},\\
&\|\boldsymbol{G}^{(0)} - \boldsymbol{G}^{(j)}\|_F \le \Lambda^{(j)}\cdot\sqrt{t\cdot(9L+m^{-1})},\\
&\|\boldsymbol{G}^{(j)}\|_F \le \widetilde{I}_1 = \sqrt{t\cdot(L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m)} + \Lambda^{(j)}\sqrt{t\cdot(9L+m^{-1})}\\
&\|\boldsymbol{f}^{(j)} - (\boldsymbol{G}^{(j)})^\intercal(\widehat{\boldsymbol{\Theta}}^{(j)} - \widehat{\boldsymbol{\Theta}}^{(0)})\|_2 \le \sqrt{t}\cdot B_3\\
&= \sqrt{t}\cdot m^{-0.5}\big(\beta_3(\Lambda^{(j)}+\beta_h) + L\cdot\Upsilon\cdot(\Lambda^{(j)}+\beta_h)(\Lambda^{(j)}/\beta_h+1)\big)
\end{aligned}
$$

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

$$
\|\boldsymbol{G}^{(0)}\|_F = \sqrt{\sum_{\tau=1}^{t}\|g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}^{(0)})\|_2^2} \le m^{-1}\beta_h\cdot\sqrt{t\cdot L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m}
$$

with the conclusion from Lemma A.5. Then,

$$
\begin{aligned}
\|\boldsymbol{G}^{(0)} - \boldsymbol{G}^{(j)}\|_F &= \sqrt{\sum_{\tau=1}^{t}\|g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}^{(0)}) - g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}^{(j)})\|_2^2}\\
&\le \Lambda^{(j)}\cdot\sqrt{t\cdot(9L+m^{-1})}.
\end{aligned}
$$

For the third inequality in this Lemma, we have

$$
\begin{aligned}
&\|\boldsymbol{f}^{(j)} - (\boldsymbol{G}^{(j)})^\intercal(\widehat{\boldsymbol{\Theta}}^{(j)} - \widehat{\boldsymbol{\Theta}}^{(0)})\|_2\\
&= \sqrt{\sum_{\tau=1}^{t}|f(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}^{(j)}) - \langle g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}^{(j)}), \boldsymbol{\Theta}^{(j)} - \boldsymbol{\Theta}^{(0)}\rangle)|^2}\\
&\le \sqrt{t}\cdot m^{-0.5}\big(\beta_3(\Lambda^{(j)}+\beta_h) + L\cdot\Upsilon\cdot(\Lambda^{(j)}+\beta_h)(\Lambda^{(j)}/\beta_h+1)\big)
\end{aligned}
$$

based on Lemma A.6.

Analogous to [4, 46], we define another auxiliary sequence to bound the parameter difference. With $\widetilde{\boldsymbol{\Theta}}^{(0)} = \boldsymbol{\Theta}^{(0)}$, we have $\widetilde{\boldsymbol{\Theta}}^{(j+1)} = \widetilde{\boldsymbol{\Theta}}^{(j)} - \eta\cdot\Big(\boldsymbol{G}^{(j)}\big((\boldsymbol{G}^{(j)})^\intercal(\widetilde{\boldsymbol{\Theta}}^{(j)} - \widetilde{\boldsymbol{\Theta}}^{(0)}) - \boldsymbol{r}\big) + m\lambda(\widetilde{\boldsymbol{\Theta}}^{(j)} - \widetilde{\boldsymbol{\Theta}}^{(0)})\Big)$.

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

$$
\|\widetilde{\boldsymbol{\Theta}}^{(j)} - \boldsymbol{\Theta}^{(0)} - \widehat{\boldsymbol{\Theta}}_t/\sqrt{m}\|_2 \le \sqrt{t/(m\lambda)}
$$

Proof. The proof is analogous to Lemma 10.2 in [4] and Lemma C.4 in [46]. Switching $\boldsymbol{G}_0$ to $\boldsymbol{G}_j$ would give the result. ■

Then, we can have the following lemma to bridge the difference between the regression estimator $\widehat{\boldsymbol{\Theta}}$ and the network parameter $\boldsymbol{\Theta}$.

Lemma B.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*

$$
\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_0 - \widehat{\boldsymbol{\Theta}}_t/\sqrt{m}\|_2 \le (\widetilde{I}_1\cdot\sqrt{t}B_3 + m\cdot\widetilde{I}_2)/(m\lambda) + \sqrt{t/(m\lambda)}
$$

with proper m, η as in Lemma 5.3

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

$$
\begin{aligned}
&\|\widetilde{\boldsymbol{\Theta}}^{(j+1)} - \boldsymbol{\Theta}^{(j+1)}\|_2\\
&\le \eta\|\boldsymbol{G}^{(j)}\|_2\|\boldsymbol{f}^{(j)} - (\boldsymbol{G}^{(j)})^\intercal(\boldsymbol{\Theta}^{(j)} - \boldsymbol{\Theta}^{(0)})\|_2 + \eta m\lambda\|\boldsymbol{\Theta}^{(0)} - \boldsymbol{\Theta}^{(j)}\|_2\\
&+ \|\boldsymbol{I} - \eta\cdot(m\lambda\boldsymbol{I} + \boldsymbol{G}^{(j)}(\boldsymbol{G}^{(j)})^\intercal)\|_2\|\boldsymbol{\Theta}^{(j)} - \widetilde{\boldsymbol{\Theta}}^{(j)}\|_2 = I_1 + I_2 + I_3.
\end{aligned}
$$

With Lemma B.1, we can bound them as

$$
\begin{aligned}
&I_1 \le \eta\cdot\widetilde{I}_1\cdot\sqrt{t}B_3\\
&I_2 \le \eta m\lambda\sqrt{\sum_{i=0}^{L}\|\boldsymbol{\Theta}_l^{(0)} - \boldsymbol{\Theta}_l^{(j)}\|_F^2} \le \eta m\cdot\widetilde{I}_2 = \eta m\lambda\sqrt{L+1}\cdot\Upsilon/\sqrt{m}.
\end{aligned}
$$

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

$$
\begin{aligned}
\eta\cdot(m\lambda\boldsymbol{I} + \boldsymbol{G}^{(0)}(\boldsymbol{G}^{(0)})^\intercal) &\le \eta\cdot\boldsymbol{I}\\
&\cdot\big(m\lambda + (m^{-1}\beta_h\cdot\sqrt{t\cdot(L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m)})^2\big) \le \boldsymbol{I}
\end{aligned}
$$

with proper choice of m and η. It leads to

$$
\|\widetilde{\boldsymbol{\Theta}}^{(j+1)} - \boldsymbol{\Theta}^{(j+1)}\|_2 \le (1-\eta m\lambda)\|\widetilde{\boldsymbol{\Theta}}^{(j)} - \boldsymbol{\Theta}^{(j)}\|_2 + \widetilde{I}_1\cdot\sqrt{t}B_3 + \eta m\cdot\widetilde{I}_2
$$

which by induction and $\widetilde{\boldsymbol{\Theta}}^{(0)} = \boldsymbol{\Theta}^{(0)}$, we have

$$
\|\widetilde{\boldsymbol{\Theta}}^{(j)} - \boldsymbol{\Theta}^{(j)}\|_2 \le (\widetilde{I}_1\cdot\sqrt{t}B_3 + m\cdot\widetilde{I}_2)/(m\lambda).
$$

Finally,

$$
\begin{aligned}
&\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_0 - \widehat{\boldsymbol{\Theta}}_t/\sqrt{m}\|_2 \le \|\widetilde{\boldsymbol{\Theta}}^{(j)} - \boldsymbol{\Theta}^{(j)}\|_2 + \|\widetilde{\boldsymbol{\Theta}}_t - \boldsymbol{\Theta}_0 - \widehat{\boldsymbol{\Theta}}_0/\sqrt{m}\|_2\\
&\le (\widetilde{I}_1\cdot\sqrt{t}B_3 + m\cdot\widetilde{I}_2)/(m\lambda) + \sqrt{t/(m\lambda)},
\end{aligned}
$$

which completes the proof.

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

$$
\begin{aligned}
&\|Z_t\|_2 \le \lambda +\\
&\quad\frac{t(L+1)}{m}\big(\Lambda^{(j)}\cdot\sqrt{9L+m^{-1}} + m^{-1}\beta_h\cdot\sqrt{L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m}\big)^2,\\
&\|\boldsymbol{G}_t^\intercal\boldsymbol{G}_t - \boldsymbol{G}_0^\intercal\boldsymbol{G}_0\|_F \le 2t\cdot m^{-1}(\Lambda^{(j)}\cdot\sqrt{9L+m^{-1}})\\
&\quad\cdot\big(\Lambda^{(j)}\sqrt{9L+m^{-1}} + m^{-1}\beta_h\cdot\sqrt{L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m}\big) = B_G/m
\end{aligned}
$$

with proper m, η as in Lemma 5.3.

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

$$
\begin{aligned}
&\|Z_t\|_2 \le \lambda + m^{-1}\sum_{\tau=1}^{t}\|g(\widetilde{\boldsymbol{X}}_\tau;\boldsymbol{\Theta}_t)\|_2^2 \le \lambda +\\
&\quad\frac{t(L+1)}{m}\big(\Lambda^{(j)}\cdot\sqrt{9L+m^{-1}} + m^{-1}\beta_h\cdot\sqrt{L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m}\big)^2
\end{aligned}
$$

with the results from Lemma A.5. Then,

$$\|G_t^\intercal G_t - G_0^\intercal G_0\|_F \le m^{-1}\cdot$$
$$\sqrt{\sum_{i,j=1}^{t} \|g(\widetilde{X}_i;\Theta_t) + g(\widetilde{X}_j;\Theta_0)\|_2^2 + \|g(\widetilde{X}_i;\Theta_t) - g(\widetilde{X}_j;\Theta_0)\|_2^2}$$
$$\le 2t\cdot m^{-1}\cdot\left(\Lambda^{(j)}\sqrt{9L+m^{-1}} + m^{-1}\beta_h\cdot\sqrt{L\cdot\beta_3^2\cdot(\beta_L\zeta)^{2L}+m}\right)$$
$$(\Lambda^{(j)}\cdot\sqrt{9L+m^{-1}}) = B_G/m.$$

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 $\|\widetilde{\Theta}^{(j+1)} - \Theta^{(j+1)}\|_2$ 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 β_F, such that $\beta_F\cdot\eta < 1$, for any $j\in[J]$:*

$$\|V^{(j)}\|_2 \le \frac{1}{4}\eta\beta_F\cdot\|F_T^{(j)} - Y_T\|_2$$

where $F^{(j)} = [f(\mathcal{G}_T,\widetilde{X}_\tau;\Theta^{(j)})]_{\tau=1}^T$, and $Y_T = [r_\tau]_{\tau=1}^T$.

Proof. We prove this lemma following an analogous approach as Lemma B.6 in [13]. Given $\widetilde{X}$, we denote $\nabla\mathcal{L}(\Theta^{(j)}) = \frac{\partial\,\mathcal{L}(\Theta^{(j)})}{\partial\,\Theta}$, and $\nabla f(\Theta^{(j)}) = \frac{\partial\, f(\mathcal{G}_T,\widetilde{X};\Theta^{(j)})}{\partial\,\Theta}$, where $\Theta\in\{\Theta_{gnn},\Theta_1,\dots,\Theta_L\}$. By the definition of $\|V^{(j)}\|$, we have its element $|V^{(j)}(\widetilde{X})|$

$$\le\eta\cdot\max_{0\le s\le\eta}\left[\sum_{\Theta}\|\nabla\mathcal{L}(\Theta^{(j)})\|_F\|\nabla f(\Theta^{(j)}) - \nabla f(\Theta^{(j)},s)\|_F\right].$$

With the notation and conclusion from Lemma A.2, we have

$$\|\nabla\mathcal{L}(\Theta^{(j)})\|_F \le m^{-\frac{1}{2}}2\sqrt{T}\|F_T^{(j)} - Y_T\|_2\cdot\zeta^L\cdot(2\beta_L)^L\beta_h$$

Meantime, $\|\nabla f(\Theta_l^{(j)}) - \nabla f(\Theta_l^{(j)},s)\|_F = m^{-\frac{L-l+1}{2}}$ $\|\boldsymbol{h}_{l-1}^{(j)}(\boldsymbol{\Theta}_L^{(j)})^\intercal\left(\prod_{q=l+1}^{L-1}\boldsymbol{\Gamma}_q^{(j)}(\boldsymbol{\Theta}_q^{(j)})^\intercal\right)\cdot\boldsymbol{\Gamma}_l^{(j)} - \boldsymbol{h}_{l-1}^{(j),s}(\boldsymbol{\Theta}_L^{(j),s})^\intercal$ $\left(\prod_{q=l+1}^{L-1}\boldsymbol{\Gamma}_q^{(j),s}\cdot(\boldsymbol{\Theta}_q^{(j),s})^\intercal\right)\cdot\boldsymbol{\Gamma}_l^{(j),s}\|_F$. A similar form can also be derived for $\boldsymbol{\Theta}_{gnn}$.

With $\Upsilon/\sqrt{m}\le 1$ and $\Lambda^{(j)}\le\beta_h$ and a similar procedure as in Lemma A.3 and Lemma A.2, we have

$$\|\boldsymbol{\Theta}^{(j+1)} - \boldsymbol{\Theta}^{(j)}\|_F \le \eta\frac{\Upsilon^{(j)}}{\sqrt{m}},\quad \|\boldsymbol{\Theta}^{(j)}\|_F \le 2\beta_L\sqrt{m}$$
$$\|\boldsymbol{h}^{(j+1)} - \boldsymbol{h}^{(j)}\|_2 \le \eta\frac{2\zeta\beta_h}{\sqrt{m}}(2\zeta\beta_L)^L\Upsilon^{(j)},\quad \|\boldsymbol{h}^{(j)}\|_2\le 2\beta_h,$$
$$\|\boldsymbol{\Gamma}^{(j+1)} - \boldsymbol{\Gamma}^{(j)}\|_F \le 2\eta\zeta^2\beta_h(2\zeta\beta_L)^L\Upsilon^{(j)},\quad \|\boldsymbol{\Gamma}^{(j)}\|_2\le\zeta$$

With Lemma G.1 from [13], for $\Theta\in\{\Theta_{gnn},\Theta_1,\dots,\Theta_L\}$,

$$\|\nabla f(\Theta^{(j)}) - \nabla f(\Theta^{(j)},s)\|_F \le \frac{4\zeta}{\sqrt{m}}\eta\Upsilon^{(j)}\beta_h L(2\zeta\beta_L)^{2L}.$$

Combining with $\|\nabla\mathcal{L}(\Theta'^{(j)})\|_F$, we have

$$|V^{(j)}(\widetilde{X})| \le \eta^2\frac{4T}{m}(L+2)^2\beta_h^3\cdot\|F_T^{(j)} - Y_T\|_2^2(2\zeta\beta_L)^{4L}.$$

Since this inequality holds for an arbitrary $\widetilde{X}\in\{\widetilde{X}_\tau\}_{\tau\in[T]}$ and $\|F_T^{(0)} - Y_T\|_2 = O(\sqrt{T})$, given network width m, we finally have

$$\|V^{(j)}\| \le \frac{1}{4}\eta\beta_F\|F_T^{(j)} - Y_T\|_2^2.$$

with the choice of learning rate $\eta\le O(T^{-1}L^{-1}\beta_h^{-2}(2\zeta\beta_L)^{-2L})$. ■

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 $\Upsilon^{(j)}/\sqrt{m}$ with $m^{-\frac{1}{2}}2\sqrt{T}\|F_T^{(j)} - Y_T\|_2\cdot\zeta^L(2\beta_L)^L\beta_h$ as the upper bound based on Lemma A.2, with $\Lambda^{(j)}\le\beta_h$, we have

$$\|F_T^{(j)} - F_T^{(j+1)}\|_2^2 = \frac{1}{m}\sum_{\tau=1}^{T}\left((\boldsymbol{h}_{L-1,\tau}^{(j+1)})^\intercal\boldsymbol{\Theta}_L^{(j+1)} - (\boldsymbol{h}_{L-1,\tau}^{(j)})^\intercal\boldsymbol{\Theta}_L^{(j)}\right)^2$$
$$\le\frac{2}{m}\left(\|\boldsymbol{\Theta}_L^{(j+1)} - \boldsymbol{\Theta}_L^{(j)}\|_2^2\sum_{\tau=1}^{T}\|\boldsymbol{h}_{L-1,\tau}^{(j+1)}\|_2^2 + \|\boldsymbol{\Theta}_L^{(j)}\|_2^2\sum_{\tau=1}^{T}\|\boldsymbol{h}_{L-1,\tau}^{(j+1)} - \boldsymbol{h}_{L-1,\tau}^{(j)}\|_2^2\right)$$
$$\le\frac{2}{m}\left(\frac{T}{m}\eta^2(2\beta_h)^4\|F_T^{(j)} - Y_T\|_2^2 + T(2\beta_3)^2(\eta\frac{2\zeta\beta_h}{\sqrt{m}}(2\zeta\beta_L)^L\Upsilon^{(j)})^2\right)$$
$$\le\frac{1}{4}\eta\beta_F\|F_T^{(j)} - Y_T\|_2^2$$

where the last inequality is due to sufficiently large m and the choice of learning rate η. ■