Learning Neural Ranking Models Online from Implicit User Feedback

Yiling Jia
University of Virginia
Charlottesville, VA, USA
yj9xs@virginia.edu

Hongning Wang
University of Virginia
Charlottesville, VA, USA
hw5x@virginia.edu

ABSTRACT

Existing online learning to rank (OL2R) solutions are limited to linear models, which are incompetent to capture possible non-linear relations between queries and documents. In this work, to unleash the power of representation learning in OL2R, we propose to directly learn a neural ranking model from users' implicit feedback (e.g., clicks) collected on the fly. We focus on RankNet and LambdaRank, due to their great empirical success and wide adoption in offline settings, and control the notorious explore-exploit trade-off based on the convergence analysis of neural networks using neural tangent kernel. Specifically, in each round of result serving, exploration is only performed on document pairs where the predicted rank order between the two documents is uncertain; otherwise, the ranker's predicted order will be followed in result ranking. We prove that under standard assumptions our OL2R solution achieves a gap-dependent upper regret bound of $O(\log^2(T))$, in which the regret is defined on the total number of mis-ordered pairs over T rounds. Comparisons against an extensive set of state-of-the-art OL2R baselines on two public learning to rank benchmark datasets demonstrate the effectiveness of the proposed solution.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; • **Theory of computation** → **Regret bounds**; **Online learning theory**.

KEYWORDS

online learning to rank, online neural ranking, explore-exploit

ACM Reference Format:
Yiling Jia and Hongning Wang. 2022. Learning Neural Ranking Models Online from Implicit User Feedback. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512250

1 INTRODUCTION

In the past decade, advances in deep neural networks (DNN) have made significant strides in improving offline learning to rank models [6, 37], thanks to DNN's strong representation learning power. But quite remarkably, most existing work in online learning to rank (OL2R) still assume a linear scoring function [41, 46, 51]. Compared with linear ranking models, nonlinear models induce a more general hypothesis space, which provides a system more flexibility and capacity in modeling complex relationships between a document's ranking features and its relevance quality. Such a clear divide between the current OL2R solutions and the successful practices in offline solutions seriously restricts OL2R's real-world impact.

The essence of OL2R is to learn from users' implicit feedback on the presented rankings, which suffers from the explore-exploit dilemma, as the feedback is known to be noisy and biased [2, 10, 24, 25]. State-of-the-art OL2R approaches employ random exploration to obtain a trade-off, and mainstream OL2R solutions are mostly different variants of dueling bandit gradient descent (DBGD) [51]. In particular, DBGD and its extensions [36, 41, 42, 51] were inherently designed for linear models, where they rely on random perturbations to sample model variants and estimate the gradient for the model update. Given the complexity of a DNN, such a random exploration method can hardly be effective. Oosterhuis and de Rijke [35] proposed PDGD, which samples the next ranked document from a Plackett-Luce model and estimates an unbiased gradient from the inferred pairwise preference. Though PDGD with a neural ranker reported promising empirical results, its theoretical property is still unknown. Most recently, Jia et al. [23] proposed to learn a pairwise ranker online using a divide-and-conquer strategy. Improved performance against all aforementioned OL2R solutions was reported by the authors. However, this solution is still limited to linear ranking functions in nature.

Turning a neural ranker online is non-trivial. While deep neural networks can be accurate on learning given user feedback, i.e., exploitation, developing practical methods to balance exploration and exploitation in complex online learning problems remains largely unsolved. In essence, quantifying a neural model's uncertainty on new data points remains challenging. Fortunately, substantial progress has been made to understand the representation learning power of DNNs. Studies in [4, 7, 8, 11, 13] showed that by using (stochastic) gradient descent, the learned parameters of a DNN are located in a particular regime, and the generalization error bound of the DNN can be characterized by the best function in the corresponding neural tangent kernel space [22]. In particular, under the framework of the neural tangent kernel, studies in [52, 54] proposed that the confidence interval of the learned parameters of a DNN can be constructed based on the random feature mapping defined by the neural network's gradient on the input instances. These efforts prepare us to study neural OL2R.

In this work, we choose RankNet [6] as our base ranker for OL2R because of its promising empirical performance in offline settings [9]. We devise exploration in the pairwise document ranking space and balance exploration and exploitation based on the ranker's

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04.
https://doi.org/10.1145/3485447.3512250

confidence about its pairwise estimation. In particular, we construct pairwise uncertainty from the tangent features of the neural network [7, 8]. In each round of result serving, all the estimated pairwise comparisons are categorized into two types, certain pairs and uncertain pairs. Documents associated with uncertain pairs are randomly shuffled for exploration, while the order among certain pairs is preserved in the presented ranking for exploitation.

We rigorously proved that our model's exploration space shrinks exponentially fast as the ranker estimation converges, such that the cumulative regret defined on the number of mis-ordered pairs has a sublinear upper bound. As most existing ranking metrics can be reduced to different kinds of pairwise document comparisons [48], we also extended our solution to LambdaRank [39] to directly optimize ranking metrics based on users' implicit feedback on the fly. To the best of our knowledge, this is the first neural OL2R solution with theoretical guarantees. Our extensive empirical evaluations also demonstrated the strong advantage of our model against a rich set of state-of-the-art OL2R solutions over two public learning to rank benchmark datasets on standard ranking metrics.

2 RELATED WORK

Online learning to rank. We broadly group existing OL2R solutions into two main categories. The first type learns the best ranked list for each individual query separately, by modeling users' click and examination behaviors with multi-armed bandit algorithms [26, 29, 40, 55]. Typically, such solutions depend on specific click models to decompose relevance estimation on each query-document pair; as a result, exploration is performed on the ranking of individual documents. For example, by assuming users examine documents from top to bottom until reaching the first relevant document, cascading bandit models rank documents based on the upper confidence bound of their estimated relevance [26, 27, 31]. The second type of solutions leverage ranking features for relevance estimation, and search for the best ranker in the model space [30, 35, 51]. The most representative work is Dueling Bandit Gradient Descent (DBGD) [42, 51]. To ensure an unbiased gradient estimate, DBGD uniformly explores in the model space, which costs high variance and high regret. Subsequent methods improved DBGD with more efficient sampling strategies, such as multiple interleaving and projected gradient, to reduce variance [20, 34, 46, 47, 53].

However, almost all of the aforementioned OL2R solutions are limited to linear models, which are incompetent to capture any non-linear relations between queries and documents. This shields OL2R away from the successful practices in offline learning to rank models, which are nowadays mostly empowered by deep neural networks [6, 37]. This clear divide has motivated some recent efforts. Oosterhuis and de Rijke [35] proposed PDGD which samples the next ranked document from a Plackett-Luce model and estimates gradients from the inferred pairwise result preferences. Though PDGD with a neural ranker achieved empirical improvements, there is no theoretical guarantee on its performance. A recent work learns a pairwise logistic regression ranker online and reports the best empirical results on several OL2R benchmarks [23]. Though non-linearity is obtained via the logistic link function, its expressive power is still limited by the manually crafted ranking features.

Theoretical analysis of neural networks. Recently, substantial progress has been made to understand the convergence of deep neural networks [19, 32, 33, 43, 44, 49, 50, 56, 58]. A series of recent studies showed that (stochastic) gradient descent can find global minimal of training loss under moderate assumptions [3, 14, 32, 57, 58]. Besides, Jacot et al. [22] proposed the neural tangent kernel (NTK) technique, which describes the change of a DNN during gradient descent based training. This motivates the theoretical study of DNNs with kernel methods. Research in [4, 7, 8, 11, 13] showed that by connecting DNN with kernel methods, (stochastic) gradient descent can learn a function that is competitive with the best function in the corresponding neural tangent kernel space. In particular, under the framework of NTK, some recent work show that the confidence interval of the learned parameters of a DNN can be constructed based on the random feature mapping defined by the neural network's gradient [52, 54]. This makes the quantification of a neural model's uncertainty possible, and enables our proposed uncertainty-based exploration for neural OL2R.

3 METHOD

In this section, we present our solution, which trains a neural ranking model with users' implicit feedback online. The key idea is to partition the pairwise document ranking space and only explore the pairs where the ranker is currently uncertain while exploiting the predicted rank of document pairs where the ranker is already certain. We rigorously prove a sublinear regret which is defined on the cumulative number of mis-ordered pairs over the course of online result serving.

3.1 Problem Setting

In OL2R, at round $t \in [T]$, the ranker receives a query q_t and its associated V_t documents represented by a set of d-dimensional query-document feature vectors: $\mathcal{X}_t = \{\mathbf{x}_1^t, ..., \mathbf{x}_{V_t}^t\}$ with $\mathbf{x}_i^t \in \mathbb{R}^d$. The ranking $\tau_t = \big(\tau_t(1), ..., \tau_t(V_t)\big) \in \Pi([V_t])$, is generated by the ranker based on its knowledge so far, where $\Pi([V_t])$ represents the set of all permutations and $\tau_t(i)$ is the rank position of document i.

The user examines the returned ranked list and provides his/her feedback, i.e., clicks $C_t = \{c_1^t, c_2^t, ..., c_{V_t}^t\}$, where $c_i^t = 1$ if the user clicked on document i at round t; otherwise $c_i^t = 0$. Then, the ranker updates itself and precedes the next round. Numerous studies have shown C_t is subject to various biases and noise, e.g., presentation bias and position bias [2, 24, 25]. In particular, it is well-known that non-clicked documents cannot be simply treated as irrelevant. Following the practice in [24], we treat clicks as relative preference feedback and assume that clicked documents are preferred over the *examined* but unclicked ones. In addition, we adopt a simple examination assumption: every document that precedes a clicked document and the first subsequent unclicked document are examined. This approach has been widely employed and proven effective in learning to rank [2, 35, 46]. We use o_t to represent the index of the last examined position in the ranked list τ_t at round t. It is worth mentioning that our solution can be easily adapted to other examination models, e.g., position based model [12], as we only use the derived result preferences as model input.

As the ranker learns from user feedback while serving, cumulative regret is an important metric for evaluating OL2R. In this work,

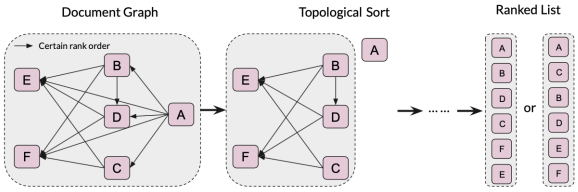

Figure 1: At round t, the ranker is confident about its order estimation between all the pairs expect $(B, C), (C, D), (E, F)$. Hence, in the ranking, the ranking orders among the certain pairs are preserved, while the uncertain pairs are shuffled.

our goal is to minimize the following regret, which is defined by the number of mis-ordered pairs from the presented ranked list to the ideal one, i.e., the Kendall's Tau rank distance,

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} K(\tau_t, \tau_t^*)\right] \tag{3.1}$$

where $K(\tau_t, \tau_t^*) = |\{(i,j) : i<j, \big(\tau_t(i)<\tau_t(j) \wedge \tau_t^*(i)>\tau_t^*(j)\big) \vee \big(\tau_t(i)>\tau_t(j) \wedge \tau_t^*(i)<\tau_t^*(j)\big)\}|$.

Remark 3.1. As shown in [48], most ranking metrics, such as Average Rank Position (ARP) and Normalized Discounted Cumulative Gain (NDCG), can be decomposed into pairwise comparisons; hence, this regret definition connects an OL2R algorithm's online performance with classical rank evaluations. We consider it more informative than "pointwise" regret defined in earlier work [26, 29].

3.2 Online Neural Ranking Model Learning

In order to unleash the power of representation learning of neural models in OL2R, we propose to directly learn a neural ranking model from its interactions with users. We balance the trade-off between exploration and exploitation based on the model's confidence about its predicted pairwise rank order. The high-level idea of the proposed solution is explained in Figure 1.

Neural Ranking Model. We focus on RankNet and LambdaRank because of their promising performance and wide adoption in offline settings [6]. In the following sections, we will focus on RankNet to explain the key components of our proposed solution for simplicity, and later we discuss how to extend the solution to LambdaRank.

We assume that there exists an unknown function $h(\cdot)$ that models the relevance quality of document $\mathbf{x}$ under the given query q as $h(\mathbf{x})$. In order to learn this function, we utilize a fully connected neural network $f(\mathbf{x};\boldsymbol{\theta}) = \sqrt{m}\mathbf{W}_L\phi(\mathbf{W}_{L-1}\phi(\dots\phi(\mathbf{W}_1\mathbf{x}))$, where depth $L \geq 2$, $\phi(\mathbf{x}) = \max\{\mathbf{x}, 0\}$, and $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_i \in \mathbb{R}^{m\times m}$, $2 \leq i \leq L-1$, $\mathbf{W}_L \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{\theta} = [\mathrm{vec}(\mathbf{W}_1)^\top, \dots, \mathrm{vec}(\mathbf{W}_L)^\top]^\top \in \mathbb{R}^p$ with $p = m + md + m^2(L-2)$. Without loss of generality, we assume the width of each hidden layer is the same as m, concerning the simplicity of theoretical analysis. We also denote the gradient of the neural network function as $\mathbf{g}(\mathbf{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x};\boldsymbol{\theta}) \in \mathbb{R}^p$.

RankNet specifies a distribution on pairwise comparisons. In particular, the probability that document i is more relevant than document j is calculated by $\mathbb{P}(i \succ j) = \sigma(f(\mathbf{x}_i;\boldsymbol{\theta}) - f(\mathbf{x}_j;\boldsymbol{\theta}))$, where $\sigma(s) = 1/(1+\exp(-s))$. For simplicity, we use f_{ij}^t to denote $f(\mathbf{x}_i;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j;\boldsymbol{\theta}_{t-1})$. Therefore, the objective function for $\boldsymbol{\theta}$ estimation in RankNet can be derived under a cross-entropy loss between the predicted pairwise comparisons and those inferred from user feedback till round t and a L2-regularization term centered at

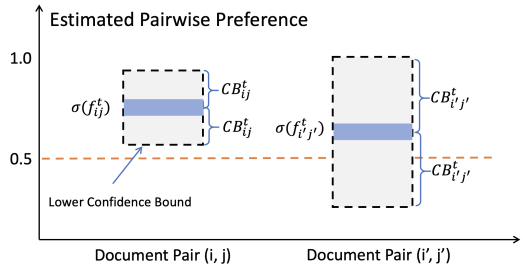

Figure 2: Illustration of certain and uncertain rank orders.

the randomly initialized parameter $\boldsymbol{\theta}_0$:

$$\mathcal{L}_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} \sum_{(i,j)\in\Omega_s} -(1-y_{ij}^s)\log(1-\sigma(f_{ij})) - y_{ij}^s \log(\sigma(f_{ij})) + m\lambda/2\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|^2, \tag{3.2}$$

where λ is the L2 regularization coefficient, Ω_s denotes the set of document pairs that received different click feedback at round s, i.e. $\Omega_s = \{(i,j) : c_i^s \neq c_j^s, \forall \tau_s(i) \leq \tau_s(j) \leq o_t\}$, y_{ij}^s indicates whether document i is preferred over document j in the click feedback, i.e., $y_{ij}^s = (c_i^s - c_j^s)/2 + 1/2$ [6].

The online estimation of RankNet boils down to the construction of $\{\Omega_t\}_{t=1}^T$ over time. However, the conventional practice of using all the inferred pairwise preferences from clicks becomes problematic in an online setting. For example, in the presence of click noise (e.g., a user mistakenly clicks on an irrelevant document), pairing documents would cause a quadratically increasing number of noisy training instances, and therefore impose a strong negative impact on the quality of the learned ranker and subsequent result serving. To alleviate this deficiency, we propose to only use *independent* pairwise comparisons to construct the training set, e.g., $\Omega_t^{ind} = \{(i,j) : c_i^t \neq c_j^t, \forall(\tau_t(i), \tau_t(j)) \in D\}$, where D represents the set of disjointed position pairs, for example, $D = \{(1,2),(3,4),\dots(o_t-1, o_t)\}$. In other words, we only use a subset of non-overlapping pairwise comparisons for update.

Result Ranking Strategy. Another serious issue in the online collected training instances is bias. As discussed before, the ranking model is updated based on the acquired feedback from what it has presented to the users so far, which is subject to various types of biases, e.g., presentation bias and position bias [2, 24, 25]. Hence, it is vital to *effectively explore the unknowns* to complete the ranker's knowledge about the ranking space, while *serving users with qualified ranking results* to minimize regret. As our solution of result ranking, we explore in the pairwise document ranking space with respect to the ranker's current uncertainty about the comparisons.

To quantify the source of uncertainty, we follow conventional click models to assume that on the *examined* documents where $\tau_t(i) \leq o_t$, the obtained feedback C_t is independent from each other given the *true relevance* of documents, so is their noise [17, 18, 24]. As a result, the noise in each collected preference pair becomes the sum of noise from the clicks in the two associated documents. Because we only use the independent pairs Ω_t^{ind}, the pairwise noise is thus independent of each other and the history of result serving, which leads to the following proposition.

Proposition 3.2. For any $t \geq 1$, $\forall(i,j) \in \Omega_t^{ind}$, the pairwise feedback follows $y_{ij}^t = \sigma(h(\mathbf{x}_i) - h(\mathbf{x}_j)) + \xi_{ij}^t$, where ξ_{ij}^t satisfying that for

all $\beta \in \mathbb{R}$, $\mathbb{E}[\exp(\beta\xi_{ij}^t)|\{\{\xi_{i'j'}^s\}_{i'j'\in\Omega_s^{ind}}\}_{s=1}^{t-1}, \Omega_{1:t-1}^{ind}] \leq \exp(\beta^2\nu^2)$, is a ν-sub-Gaussian random variable

Based on the property of sub-Gaussian random variables, the proposition above can be easily satisfied in practice as long as the pointwise click noise follows a sub-Gaussian distribution. Typically the pointwise noise is modeled as a binary random variable related to the document's true relevance under the given query, which follows a $\frac{1}{2}$-sub-Gaussian distribution. Let Ψ_t represent the set of all possible document pairs at round t, e.g., $\Psi_t = \{(i,j) \in [V_t]^2, i \neq j\}$ and $|\Psi_t| = V_t^2 - V_t$. Based on the objective function Eq (3.2) over training dataset $\{\Omega_s^{ind}\}_{s=1}^t$, we have the following lemma bounding the uncertainty of the estimated pairwise rank order at round t.

Lemma 3.3. (Confidence Interval of Pairwise Rank Order). There exist positive constants C_1 and C_2 such that for any $\delta_1 \in (0,1)$, if the step size of gradient descent $\eta \leq C_1(TmL + m\lambda)^{-1}$ and $m \geq C_2 \max\left\{\lambda^{-1/2}L^{-3/2}(\log(TV_{\max}L^2/\delta_1))^{3/2}, T^7\lambda^{-7}L^{21}(\log m)^3\right\}$, then at round $t < T$, for any document pair $(i,j) \in \Psi_t$ under query q_t, with probability at least $1-\delta_1$,

$$|\sigma(f_{ij}^t) - \sigma(h_{ij})| \leq \alpha_t \|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}} + \epsilon(m), \tag{3.3}$$

where $V_{\max}$ represents the maximum number of documents under a query over time, $h_{ij} = h(\mathbf{x}_i) - h(\mathbf{x}_j)$, $\mathbf{g}_{ij}^s = \mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}_s) - \mathbf{g}(\mathbf{x}_j;\boldsymbol{\theta}_s)$, $\mathbf{A}_t = \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \frac{1}{m}\mathbf{g}_{i'j'}^s \mathbf{g}_{i'j'}^{s\top} + \lambda\mathbf{I}$, $\bar{C}_1$, $\bar{C}_2$, $\bar{C}_3$ and $\bar{C}_4$ are positive constants,

$$\begin{aligned}
\epsilon(m) =& \bar{C}_1\Big(T^{\frac{7}{6}}m^{-\frac{1}{6}}\lambda^{-\frac{7}{6}}L^4\sqrt{\log(m)}(1+\sqrt{T/\lambda}) + (1-\eta m\lambda)^{\frac{J}{2}}\sqrt{TL/\lambda} \\
&+ T^{\frac{1}{6}}m^{-\frac{1}{6}}\lambda^{-\frac{1}{6}}L^{\frac{7}{2}}\sqrt{\log(m)}S + T^{\frac{2}{3}}m^{-\frac{1}{6}}\lambda^{-\frac{2}{3}}L^3\sqrt{\log(m)}\Big), \\
\alpha_t =& \Big(1 + \bar{C}_2 T^{\frac{7}{6}}m^{-\frac{1}{6}}\sqrt{\log(m)}\lambda^{-\frac{7}{6}}L^4\Big)^{\frac{1}{2}} \cdot \bar{\alpha}_t, \\
\bar{\alpha}_t =& \Big(\sqrt{\lambda}\bar{C}_3 + (\nu^2\log(\frac{\det(\mathbf{A}_t)}{\delta_1^2\det(\lambda\mathbf{I})}) + \bar{C}_4 T^{\frac{5}{3}}m^{-\frac{1}{6}}\lambda^{-\frac{1}{6}}L^4\sqrt{\log(m)})^{\frac{1}{2}}\Big).
\end{aligned}$$

We provide the detailed proof of Lemma 3.3 in the appendix. This lemma provides a tight high probability bound of the pairwise rank order estimation uncertainty under RankNet. The uncertainty caused by the variance from the pairwise observation noise is controlled by α_t, and $\epsilon(m)$ is the approximation error incurred in the estimation of the true scoring function. This enables us to perform efficient exploration in the pairwise document ranking space for the model update. To illustrate our ranking strategy, we introduce the following notion on the estimated pairwise preference.

Definition 3.4. (Certain Rank Order) At round t, the rank order between documents $(i,j) \in \Psi_t$ is in a certain rank order if and only if $\sigma(f_{ij}^t) - CB_{ij}^t > \frac{1}{2}$, where $CB_{ij}^t = \alpha_t\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}} - \epsilon(m)$ is the width of confidence bound about the estimated pairwise rank order.

Based on Lemma 3.3, if an estimated rank order $(i \succ j)$ is a certain rank order, with a high probability that the estimated preference is consistent with the ground-truth. Hence, they should be followed in the returned ranked list. For example, as shown in Figure 2, the lower bound for $\sigma(f_{ij}^t)$ is larger than 1/2, which indicates consistency between the estimated and ground-truth order between (i,j). But with $\sigma(f_{i'j'}^t) - CB_{i'j'}^t < 1/2$, the estimated order $(i' \succ j')$ is still uncertain as the ground-truth may present an opposite order.

We use ω_t to represent the set of all certain rank orders at round t, $\omega_t = \{(i,j) \in \Psi_t : \sigma(f_{i,j}^t) - CB_{i,j}^t > \frac{1}{2}\}$. For pairs in ω_t, we can directly exploit the current estimated rank order as it is already consistent with the ground-truth. But, for the uncertain pairs that do not belong to ω_t, exploration is necessary to obtain feedback for further model update (and thus to reduce uncertainty). For example, in the document graph shown in Figure 1, when generating the ranked list, we should exploit the current model by preserving the certain orders, while randomly swap the order between documents (B, C), (C, D), (E, F) to explore (in order to conquer feedback bias).

Algorithm 1 Online Neural Ranking Algorithm

Input: L2 coefficient λ, step size η, number of iterations for gradient descent J, network width m, network depth L.
Initialize $\boldsymbol{\theta}_0 = (\text{vec}(\mathbf{W}_1), \ldots \text{vec}(\mathbf{W}_L)) \in \mathbb{R}^p$, where for each $1 \leq l \leq L-1$, $\mathbf{W}_l = (\mathbf{W}, \mathbf{0}; \mathbf{0}, \mathbf{W})$, each entry of $\mathbf{W}$ is initialized independently from $N(0, 4/m)$; $\mathbf{W}_L = (\mathbf{w}^\top, -\mathbf{w}^\top)$, where each entry of $\mathbf{w}$ is initialized independently from $N(0, 2/m)$.
Initialize $A_1 = \lambda\mathbf{I}$
for $t = 1, \ldots, T$ **do**
  $q_t \leftarrow receive\_query(t)$
  $\mathcal{X}_t = \{\mathbf{x}_1^t, \cdots, \mathbf{x}_{n_t}^t\} \leftarrow retrieve\_documents(q_t)$
  $\omega_t \leftarrow construct\_certain\_rank\_order\_set(\mathcal{X}_t, \boldsymbol{\theta}_{t-1}, A_t)$
  $\tau_t \leftarrow topological\_sort(\omega_t)$
  $C_t \leftarrow collect\_click\_feedback(\tau_t)$
  $\Omega_t^{ind} \leftarrow construct\_independent\_pairs(C_t)$
  Set $\boldsymbol{\theta}_t$ to be the output of gradient descent with step size η for J rounds on minimize Eq (3.2).
  $\mathbf{A}_{t+1} = \mathbf{A}_t + \sum_{(i,j)\in\Omega_t^{ind}} \mathbf{g}_{ij}^t \mathbf{g}_{ij}^{t\top}/m$
end for

The estimated pairwise rank order, $\sigma(f_{ij}^t)$, is derived based on relevance score calculated by the current neural network, i.e., $f(\mathbf{x}_i;\boldsymbol{\theta}_{t-1})$ and $f(\mathbf{x}_j;\boldsymbol{\theta}_{t-1})$. Hence, as shown in Figure 1, due to the monotonicity and transitivity of the sigmoid function, the document graph constructed with the candidate documents as the vertices and the certain rank order as the directed edges is a directed acyclic graph (DAG). We can perform a topological sort on the constructed document graph to efficiently generate the final ranked list. The certain rank orders are preserved by topological sort to exploit the ranker's high confidence predictions. On the other hand, the topological sort randomly chooses vertices with zero in-degree, among which there is no certain rank orders. This naturally achieves exploration among uncertain rank orders. In Figure 1, as document A is predicted to be better than all the other documents by certain rank orders, it will be first added to the ranked list and removed from the document graph by topological sort. In the updated document graph, both document B and C become vertices with zero in-degree as the estimated rank order between them is still uncertain. Topological sort will randomly choose one of them as the next document in the ranked list, which induces exploration on the uncertain rank orders. Two possible ranked lists are shown in the figure. As exploration is confined to the pairwise ranking space, it effectively reduces the exponentially sized exploration space of result ranking to quadratic. Algorithm 1 shows the details of the proposed solution.

Extend to LambdaRank. LambdaRank directly optimizes the ranking metric of interest (e.g., NDCG) with a modified gradient based on RankNet [6]. For a given pair of documents, the confidence interval of LambdaRank's estimation can be calculated by gradients of the neural network in the same way as in RankNet (i.e., by Lemma 3.3). However, as the objective function of LambdaRank is

unknown, it prevents us from theoretically analyzing the resulting online algorithm's regret. But similar empirical improvement from LambdaRank against RankNet known in the offline settings [6] is also observed in our online versions of these two algorithms.

4 REGRET ANALYSIS

Our regret analysis is built on the latest theoretical studies in deep neural networks. Recent attempts show that in the neural tangent kernel (NTK) space, the generalization error bound of a DNN can be characterized by the corresponding best function [4, 7, 8, 11, 13]. In our analysis, we denote the NTK matrix of all possible pairwise document tangent features as $\mathbf{H} \geq \lambda_0 \mathbf{I}$, with the effective dimension of $\mathbf{H}$ denoted as $\widetilde{d}$. Due to limited space, we leave the detailed definition of $\mathbf{H}$ and $\widetilde{d}$ in the appendix.

We define event E_t as: $E_t = \left\{\forall (i,j) \in \Psi_t, |\sigma(f_{ij}^t) - \sigma(h_{ij})| \leq CB_{i,j}^t\right\}$ at round t. E_t suggests that the estimated pairwise rank order on all the candidate document pairs under query q_t is close to the ground-truth at round t. According to Lemma 3.3, it is easy to reach the following conclusion,

Corollary 4.1. On the event E_t, it holds that $\sigma(h_{ij}) > \frac{1}{2}$ if $(i,j) \in \omega_t$, i.e., in a certain rank order.

Based on the definition of pairwise regret in Eq (3.1), the ranker only suffers regret as a result of misplacing a pair of documents, i.e., swapping a pair into an incorrect order. According Corollary 4.1, under event E_t, the certain rank order identified is consistent with the ground-truth. As in our proposed solution, the certain rank order is preserved by the topological sort, it is easy to verify that regret only occurs on the document pairs with uncertain rank order. Therefore, the key step in our regret analysis is to count the expected number of uncertain rank orders. According to Definition 3.4, a pairwise estimation is certain if and only if $|\sigma(f_{ij}^t) - \frac{1}{2}| \geq CB_{i,j}^t$. Hence, we have the following lemma bounding the probability that an estimated rank order being uncertain.

Lemma 4.2. With η, m satisfying the same conditions in Lemma 3.3, with $\delta_1 \in (0, 1/2)$ defined in Lemma 3.3, and $\delta_2 \in (0, 1/2)$, such that for $t \geq t' = O(\log(1/\delta_2) + \log(1/\delta_1))$, under event E_t, the following holds with probability at least $1 - \delta_2$:

$$\forall (i,j) \in \Psi_t, \mathbb{P}((i,j) \notin \omega_t) \leq \frac{C_u \log(1/\delta_1)}{(\Delta_{\min} - 2\epsilon(m))^2} \|\mathbf{g}_{ij}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{-1}},$$

where $C_u = 8\nu^2 k_\mu^2 / c_\mu^2$ with k_μ and c_μ as the Lipschitz constants for the sigmoid function, $\Delta_{\min} = \min_{t \in T, (i,j) \in \Psi_t} |\sigma(h_{ij}) - \frac{1}{2}|$ represents the smallest gap of pairwise difference between any pair of documents under the same query over time.

Remark 4.3. With m satisfying the condition in Lemma 3.3, and setting the corresponding η and $J = \widetilde{O}(TL/\lambda)$, $\epsilon(m) = O(1)$ can be achieved. More specifically, there exists a positive constant c such that $\Delta_{\min} - 2\epsilon(m) = c\Delta_{\min}$.

Lemma 4.2 gives us a tight bound for an estimated pairwise order being uncertain. Intuitively, it targets to obtain a tighter bound on the uncertainty of the neural model's parameter estimation compared to the bound determined by δ_1 in Lemma 3.3. With this bound, the corresponding confidence interval will exclude the possibility of flipping the estimated rank order, i.e., the lower confidence bound of this pairwise estimation is above 0.5.

In each round of result serving, as the model $\boldsymbol{\theta}_t$ will not change before the next round starts, the expected number of uncertain rank orders, denoted as $\mathbb{E}[U_t]$, can be estimated by the summation of the uncertain probabilities over all possible pairwise comparisons under the query q_t, e.g., $\mathbb{E}[U_t] = \frac{1}{2}\sum_{(i,j)\in\Psi_t} \mathbb{P}((i,j) \notin \omega_t)$. Denote p_t as the probability that the user examines all documents in τ_t at round t, and let $p^* = \min_{1 \leq t \leq T} p_t$ be the minimal probability that all documents in a query are examined over time. We present the upper regret bound as follows.

Theorem 4.4. With δ_1 and δ_2 defined in Lemma 3.3, 4.2, η, m satisfying the same conditions in Lemma 3.3, there exist positive constants $\{C_i^r\}_{i=1}^2$ that with probability at least $1 - \delta_1$, the T-step regret is bounded by:

$$R_T \leq R' + (C_1^r \log(1/\delta_1)\widetilde{d}\log(1 + TV_{\max}/\lambda) + C_2^r)(1 - \delta_2)/(\Delta_{\min}^2 p^*)$$

where $R' = t'V_{\max}^2 + (T - t')\delta_2 V_{\max}^2$, with t' and $V_{\max}$ defined in Lemma 4.2. By choosing $\delta_1 = \delta_2 = 1/T$, the expected regret is at most $O(\widetilde{d}\log^2(T))$.

Proof Sketch. The detailed proof is provided in the appendix. We only provide the key ideas behind our regret analysis here. The regret is first decomposed into two parts. First, R' represents the regret when Lemma 4.2 does not hold, in which the regret is out of our control. We use the maximum number of pairs associated with a query over time, i.e., $V_{\max}^2$, to upper bound it. The second part corresponds to the cases when Lemma 4.2 holds. Then, the instantaneous regret at round t can be bounded by $r_t = \mathbb{E}\left[K(\tau_t, \tau_t^*)\right] \leq \mathbb{E}[U_t]$, as only the uncertain rank orders would induce regret. □

In this analysis, we provide a gap-dependent regret upper bound, where the gap $\Delta_{\min}$ characterizes the intrinsic difficulty of sorting the V_t candidate documents at round t. Intuitively, when $\Delta_{\min}$ is small, e.g., comparable to the network's resolution $\epsilon(m)$, many observations are needed to recognize the correct rank order between two documents. As the matrix $\mathbf{A}_t$ only contains information from examined document pairs, our algorithm guarantees that the cumulative pairwise regret of the examined documents until round t ($\{1 : o_s\}_{s=1}^t$) to be sub-linear, while the regret in the leftover documents ($\{o_s + 1 : V_s\}_{s=1}^t$) is undetermined. We adopt a commonly used technique that leverages the probability that a ranked list is fully examined to bound the regret on those unexamined documents [27, 28, 31]. This probability is a constant independent of T. It is worth noting that our algorithm does not need the knowledge of p^* for model learning or result ranking; it is solely used for the regret analysis to handle the partial observations. From a practical perspective, the ranking quality of documents ranked below o_s for $s \in [T]$ does not affect users' online experience, as the users do not examine them. Hence, if we only count regret in the examined documents, R_T does not need to be scaled by p^*

Remark 4.5. Our regret is defined over the number of mis-ordered pairs, which is the *first* pairwise regret analysis for a neural OL2R algorithm. Existing OL2R algorithms optimize their own metrics (e.g., utility function as defined in [51]), which can hardly link to any conventional ranking metrics. As shown in [48], most classical ranking evaluation metrics, such as NDCG, are based on pairwise document comparisons. Our regret analysis connects our OL2R solution's theoretical property with such metrics, which is also confirmed in our empirical evaluations.

5 EXPERIMENTS

In this section, we empirically compare our proposed models with an extensive list of state-of-the-art OL2R algorithms on two large public learning to rank benchmark datasets.

Reproducibility Our entire codebase, baselines, analysis, and experiments can be found on Github [1].

5.1 Experiment Setup

Datasets. We experiment on two publicly available learning to rank datasets, Yahoo! Learning to Rank Challenge dataset [9], which consists of 292,921 queries and 709,877 documents represented by 700 ranking features, and MSLR-WEB10K [38], which contains 30,000 queries, each having 125 documents on average represented by 136 ranking features. Both datasets are labeled on a five-grade relevance scale: from not relevant (0) to perfectly relevant (4). We followed the train/test/validation split provided in the datasets to make our results comparable to the previously reported results.

Non-linearity analysis. Most of the existing OL2R models assume that the expected relevance of a document under the given query can be characterized by a linear function in the feature space. However, such an assumption often fails in practice, as the potentially complex non-linear relations between queries and documents are ignored. For example, classical query-document features are usually constructed in parallel to the design and choices of ranking models. As a result, a lot of correlated and sometimes redundant features are introduced for historical reasons; and the ranker is expected to handle it. For instance, the classical keyword matching based features, such as TF-IDF, BM25 and language models, are known to be highly correlated [15]; and the number of in-links is also highly related to the PageRank feature.

To verify this issue, we performed a linear discriminative analysis (LDA) [5] on both datasets. The technique of LDA is typically used for multi-class classification that automatically performs dimensionality reduction, providing a projection of the dataset that can best linearly separate the samples by their assigned class. We provide the entire labeled dataset for the algorithm to learn the separable representation. We set the reduced dimension to be two to visualize the results. In Figure 3, we can clearly observe that a linear model is insufficient to separate the classes in both datasets.

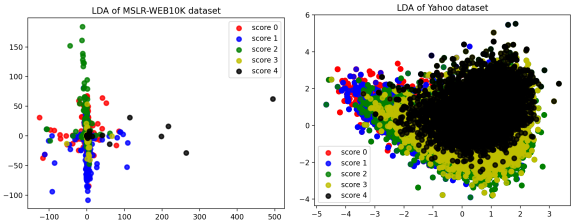

Figure 3: LDA results on both datasets

User interaction simulation. For reproducibility, user clicks are simulated via the standard procedure for OL2R evaluations [35]. At each round, a query is uniformly sampled from the training set for result serving. Then, the model determines the ranked list and returns it to the user. User click is simulated with a dependent click model (DCM) [18], which assumes that the user will sequentially scan the list and make click decisions on the examined documents. In DCM, the probabilities of clicking on a given document and stopping examination are both conditioned on the document's true relevance label. We employ three different model configurations to represent three different types of users, for which details are shown in Table 1. Basically, we have the *perfect* users, who click on all relevant documents and do not stop browsing until the last returned document; the *navigational* users, who are very likely to click on the first encountered highly relevant document and stop there; and the *informational* users, who tend to examine more documents, but sometimes click on irrelevant documents, such that contributing a significant amount of noise in their click feedback. To reflect presentation bias, all models only return the top 10 ranked results.

Table 1: Configuration of simulated click models.

	Click Probability					Stop Probability				
R	0	1	2	3	4	0	1	2	3	4
per	0.0	0.2	0.4	0.8	1.0	0.0	0.0	0.0	0.0	0.0
nav	0.05	0.3	0.5	0.7	0.95	0.2	0.3	0.5	0.7	0.9
inf	0.4	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4	0.5

Baselines. We list the OL2R solutions used for our empirical comparisons below. And we name our proposed model as olRankNet and olLambdaRank in the experiment result discussions.

- ϵ-**Greedy** [21]: At each position, it randomly samples an unranked document with probability ϵ or selects the next best document based on the currently learned RankNet.
- **Linear-DBGD and Neural-DBGD** [51]: DBGD uniformly samples a direction from the entire model space for exploration and model update. We apply it to both linear and neural rankers.
- **Linear-PDGD and Neural-PDGD** [35]: PDGD samples the next ranked document from a Plackett-Luce model and estimates gradients from the inferred pairwise preferences. We also apply it to both linear and neural network rankers.
- **PairRank** [23]: This is a recently proposed OL2R solution based on a pairwise logistic regression ranker. As it is designed for logistic regression, it cannot be used for learning a neural ranker.
- **olLambdaRank GT**: At each round, we estimate a new LambdaRank model with ground-truth relevance labels of all the presented queries. This serves as the skyline in all our experiments.

Hyper-Parameter Tuning. MSLR-WEB10K and Yahoo Learning to Rank dataset are equally partitioned into five folds, of which three parts are used for for training, one part for validation and and one part test. We did cross validation on each dataset. For each fold, the models are trained on the training set, and the hyper-parameters are selected based on the performance on the validation set.

In the experiment, a two-layer neural network with width $m = 100$ is applied for all the neural rankers. We did a grid search for olRankNet and olLambdaRank for regularization parameter λ over $\{10^{-i}\}_{i=1}^{4}$, exploration parameter α over $\{10^{-i}\}_{i=1}^{4}$, learning rate over $\{10^{-i}\}_{i=1}^{3}$. The same set of parameter tuning is applied for PairRank, except the model is directly optimized with L-BFGS. The model update in PDGD and DBGD is based on the optimal settings in their original paper. The hyper-parameters for PDGD and DBGD are the update learning rate and the learning rate decay, for which we performed a grid search for learning rate over $\{10^{-i}\}_{i=1}^{3}$, and the learning rate decay is set as 0.999977.

[1] https://github.com/HCDM/OnlineLearningToRank

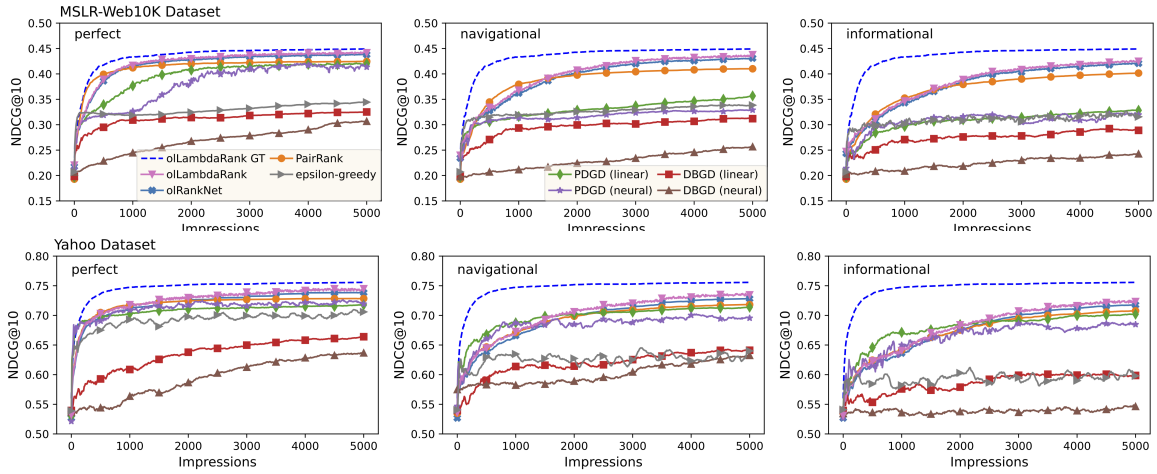

Figure 4: Offline performance on two benchmark datasets under three different click model configurations.

5.2 Experiment Results

Offline performance. The offline performance is evaluated in an "online" fashion: the newly updated ranker is immediately evaluated on a hold-out testing set against its ground-truth labels. This measures how rapidly an OL2R model improves its ranking quality, and it is an important metric about users' instantaneous satisfaction. This can be viewed as using one portion of traffic for online model update, while serving another portion with the latest model. We use NDCG@10 to assess the ranking quality, and we compare all algorithms over three click models and two datasets. For olRankNet and olLambdaRank, since it is computationally expensive to store and operate on a complete $\mathbf{A}_t$ matrix, we only used its diagonal elements as an approximation. We fixed the total number of iterations T to 5000. The experiments are executed for 10 times with different random seeds and the averaged results are reported in Figure 4.

We can clearly observe that our proposed online neural ranking models achieved significant improvement compared to all baselines. Under different click models, both linear and neural DBGD performed the worst. This is consistent with previous findings: DBGD depends on interleave tests to determine the update direction in the model space. But such model-level feedback cannot inform the optimization of any rank-based metric. Moreover, with a neural ranker, random exploration becomes very ineffective. PDGD consistently outperformed DBGD under different click models. However, its document sampling based exploration limits its learning efficiency, especially when users only examine a small portion of documents, e.g., the navigational users. It is worth noting that in the original paper [35], PDGD with a neural ranker outperformed linear ranker after much more interactions, e.g., 20000 iterations. Our proposed solutions with only 5000 iterations already achieved better performance than the best results reported for PDGD, which demonstrates the encouraging efficiency of our proposed OL2R solution. Compared to PairRank, our neural rankers had a worse start at the beginning. We attribute it to the limited training samples available at the initial rounds, i.e., the network parameters were not well estimated yet. But the neural model enables non-linear relation learning and quickly leads to better performance than the linear models when more observations arrive. Compared to olRankNet, olLambdaRank directly optimizes the evaluation metrics, e.g., NDCG@10, with corresponding gradients. We can observe similar improvements from LambdaRank compared to RankNet as previously reported in offline settings. It is worth noting that though the improvement of olRankNet and olLambdaRank compared to PairRank is not as large as their improvement against other baselines in the figure, small improvement in the performance metric often means a big leap forward in practice as most real-world systems serve millions of users, where even a small percentage improvement can be translated into huge utility gain to the population.

Online performance. In OL2R, in addition to the offline evaluation, the models' ranking performance during online result serving should also be considered, as it reflects user experience during model update. Sacrificing users experience for model training will compromise the goal of OL2R. We adopt the cumulative Normalized Discounted Cumulative Gain to assess models' online performance. For T rounds, the cumulative NDCG is calculated as

$$\text{Cumulative NDCG} = \sum_{t=1}^{T} \text{NDCG}(\tau_t) \cdot \gamma^{(t-1)},$$

which computes the expected utility a user receives with a probability γ that he/she stops searching after each query [35]. Following the previous work [35, 46], we set $\gamma = 0.9995$. Figure 5 shows the online performance of the proposed online neural ranking model and all the other baselines. It is clear to observe that DBGD-based models have a much slower convergence and thus have worse online performance. Compared to the proposed solution, PDGD showed consistently worse performance, especially under the navigational and informational click models with a neural ranker. We attribute this difference to the exploration strategy used in PDGD: PDGD's sampling-based exploration can introduce unwanted distortion in the ranked results, especially at the early stage of online learning. We should note the earlier stages in cumulative NDCG plays a much more important role due to the strong shrinking effect of γ.

Our proposed models demonstrated significant improvements over all baseline methods on both datasets under three different click models. Such improvement indicates the effectiveness our uncertainty based exploration, which only explores when the ranker's pairwise estimation is uncertain. Its advantage becomes more apparent in this online ranking performance comparison, as an overly

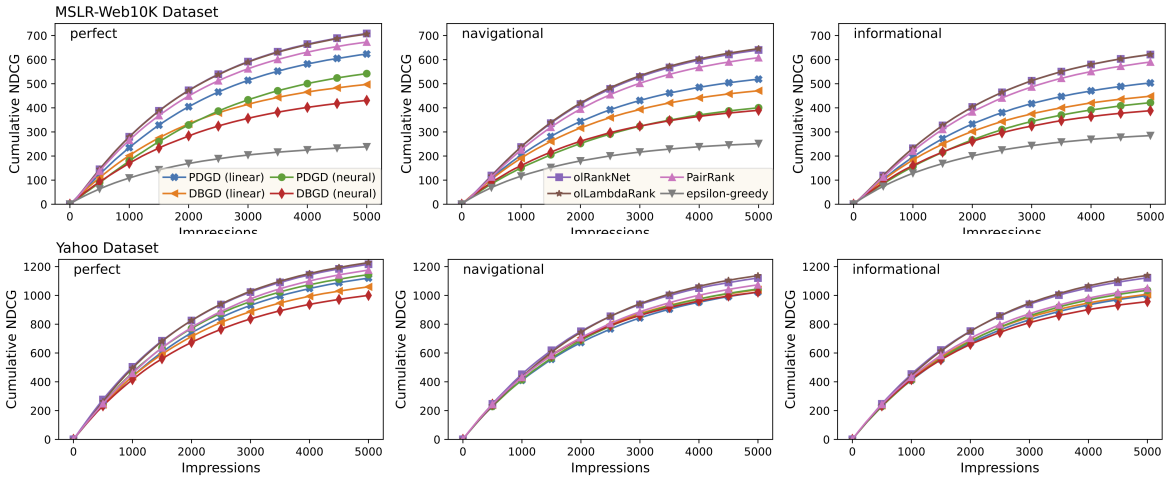

Figure 5: Online performance on two datasets under three different click model configurations.

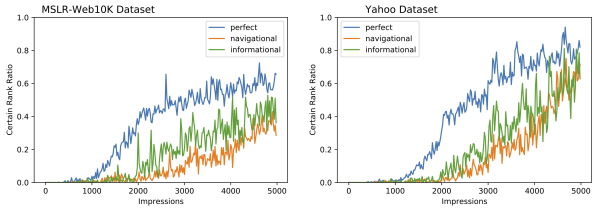

Figure 6: Ratio of certain rank orders at Top-10 positions over the rounds of online model update.

aggressive exploration in the early stage costs more in cumulative NDCG. We can also observe the improvement of olLambdaRank compared to olRankNet in this online evaluation, although the difference is not very significant. The key reason is also the strong discount applied to the later stage of model learning: olLambdaRank's advantage in directly optimizing the rank metric becomes more apparent in the later stage, as suggested by the offline performance in Figure 4. At the beginning of model learning, both models are doing more explorations and therefore the online performance got more influenced by the number of document pairs with uncertain rank orders, rather than those with certain rank orders.

Shrinkage of the number of uncertain rank orders. To further verify the effectiveness of the exploration strategy in our proposed online neural ranking model, we zoom into the trace of the number of identified certain rank orders under each query during online model update. As the model randomly shuffles the uncertain rank orders to perform the exploration, a smaller ratio of uncertain rank orders is preferred to reduce the regret, especially at the top ranked positions. Figure 6 reports the ratio of certain rank orders among all possible document pairs at top-10 positions in our olRankNet model. We can clearly observe that the certain rank orders quickly reach a promising level, especially on the Yahoo dataset. This confirms our theoretical analysis about the convergence of the number of uncertain rank orders. Comparing the results under different click models, we can observe that the convergence under navigational click model is slower. We attribute it to the limited feedback observed during the online interactions, because the stop probability is much higher in the navigational click model.

6 CONCLUSION

Existing OL2R solutions are limited to linear models, which have shown to be incompetent to capture the potential non-linear relations between queries and documents. Motivated by the recent advances in the theoretical deep learning, we propose to directly learn a neural ranker on the fly. During the course of online learning, we assess the ranker's pairwise rank estimation uncertainty based on the tangent features of the neural network. Exploration is performed only on the pairs where the ranker is still uncertain; and for the rest of pairs we follow the predicted rank order. We prove a sub-linear upper regret bound defined on the number of mis-ordered pairs, which directly links the proposed solution's convergence with classical ranking evaluations. Our empirical experiments support our regret analysis and demonstrate significant improvement over several state-of-the-art OL2R solutions.

Our effort sheds light on deploying powerful offline learning to rank solutions online and directly optimizing rank-based metrics, e.g., RankNet and LambdaRank. Furthermore, our solution can be readily extended to more recent and advanced neural rankers (e.g., those directly learn from query-document pairs without manually constructed features). On the other hand, computational efficiency is a practical concern for online algorithms. Our current solution requires gradient descent on the online collected training instances, which is undeniably expensive. We would like to investigate the feasibility of online stochastic gradient descent and its variants, in the setting of continual learning, which would greatly reduce the computational complexity of our solution.

ACKNOWLEDGMENTS

This paper is based upon the work supported by the National Science Foundation under grant IIS-1553568 and IIS-2128019, and Google Faculty Research Award.

REFERENCES

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

[2] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th ACM SIGIR*. ACM, 19–26.

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

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

[5] Suresh Balakrishnama and Aravind Ganapathiraju. 1998. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing* 18, 1998 (1998), 1–8.

[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[7] Yuan Cao and Quanquan Gu. 2019. Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks. In *Advances in Neural Information Processing Systems*.

[8] Yuan Cao and Quanquan Gu. 2020. Generalization Error Bounds of Gradient Descent for Learning Over-parameterized Deep ReLU Networks. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

[9] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*. 1–24.

[10] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. *ACM TOIS* 30, 1 (2012), 6.

[11] Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. 2019. How Much Over-parameterization Is Sufficient to Learn Deep ReLU Networks? *arXiv preprint arXiv:1911.12360* (2019).

[12] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.

[13] Amit Daniely. 2017. SGD learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*. 2422–2430.

[14] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2019. Gradient Descent Provably Optimizes Over-parameterized Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1eK3i09YQ

[15] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 49–56.

[16] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. 2010. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*. 586–594.

[17] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click chain model in web search. In *Proceedings of the 18th WWW*. 11–20.

[18] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the 2nd WSDM*. 124–131.

[19] Boris Hanin and Mark Sellke. 2017. Approximating Continuous Functions by ReLU Nets of Minimal Width. *arXiv preprint arXiv:1710.11278* (2017).

[20] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2012. Estimating interleaved comparison outcomes from historical click data. In *Proceedings of the 21st CIKM*. 1779–1783.

[21] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2013. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval* 16, 1 (2013), 63–90.

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

[23] Yiling Jia, Huazheng Wang, Stephen Guo, and Hongning Wang. 2021. Pairrank: Online pairwise learning to rank by divide-and-conquer. In *Proceedings of the Web Conference 2021*. 146–157.

[24] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th ACM SIGIR*. ACM, 154–161.

[25] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS* 25, 2 (2007), 7.

[26] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. 2015. Cascading bandits: Learning to rank in the cascade model. In *ICML*. 767–776.

[27] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Combinatorial cascading bandits. In *NIPS*. 1450–1458.

[28] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*. 535–543.

[29] Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. 2018. Toprank: A practical algorithm for online stochastic ranking. In *NIPS*. 3945–3954.

[30] Shuai Li, Tor Lattimore, and Csaba Szepesvári. 2018. Online learning to rank with features. *arXiv preprint arXiv:1810.02567* (2018).

[31] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual Combinatorial Cascading Bandits.. In *ICML*, Vol. 16. 1245–1253.

[32] Shiyu Liang and R Srikant. 2016. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161* (2016).

[33] Haihao Lu and Kenji Kawaguchi. 2017. Depth Creates No Bad Local Minima. *arXiv preprint arXiv:1702.08580* (2017).

[34] Harrie Oosterhuis and Maarten de Rijke. 2017. Balancing speed and quality in online learning to rank for information retrieval. In *Proceedings of the 26th 2017 ACM CIKM*. 277–286.

[35] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM CIKM*. 1293–1302.

[36] Harrie Oosterhuis, Anne Schuth, and Maarten de Rijke. 2016. Probabilistic multileave gradient descent. In *European Conference on Information Retrieval*. Springer, 661–668.

[37] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2019. Tf-ranking: Scalable tensorflow library for learning-to-rank. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2970–2978.

[38] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. arXiv:1306.2597 [cs.IR]

[39] C Quoc and Viet Le. 2007. Learning to rank with nonsmooth cost functions. *Proceedings of the Advances in Neural Information Processing Systems* 19 (2007), 193–200.

[40] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *ICML*. 784–791.

[41] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In *Proceedings of the 9th ACM WSDM*. 457–466.

[42] Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. 2014. Multileaved comparisons for fast online evaluation. In *Proceedings of the 23rd ACM CIKM*. ACM, 71–80.

[43] Matus Telgarsky. 2015. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101* (2015).

[44] Matus Telgarsky. 2016. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485* (2016).

[45] Roman Vershynin. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027* (2010).

[46] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. 2019. Variance Reduction in Gradient Exploration for Online Learning to Rank. In *SIGIR 2019*. 835–844.

[47] Huazheng Wang, Ramsey Langley, Sonwoo Kim, Eric McCord-Snook, and Hongning Wang. 2018. Efficient exploration of gradient space for online learning to rank. In *SIGIR 2018*. 145–154.

[48] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *CIKM '18*. ACM, 1313–1322.

[49] Dmitry Yarotsky. 2017. Error bounds for approximations with deep ReLU networks. *Neural Networks* 94 (2017), 103–114.

[50] Dmitry Yarotsky. 2018. Optimal approximation of continuous functions by very deep ReLU networks. *arXiv preprint arXiv:1802.03620* (2018).

[51] Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*. 1201–1208.

[52] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Thompson Sampling. *arXiv preprint arXiv:2010.00827* (2020).

[53] Tong Zhao and Irwin King. 2016. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of the 25th ACM CIKM*. 1643–1652.

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

[55] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. 2017. Online learning to rank in stochastic click models. In *ICML 2017*. 4199–4208.

[56] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. 2019. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning* (2019).

[57] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. 2020. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning* 109, 3 (2020), 467–492.

[58] Difan Zou and Quanquan Gu. 2019. An Improved Analysis of Training Over-parameterized Deep Neural Networks. In *Advances in Neural Information Processing Systems*.

A PROOF OF LEMMA 3.3

Before we provide the detailed proofs, we first assume that there are n_t possible documents to be evaluated during the model learning until round t. It is easy to conclude that $n_t = \sum_{s=1}^{t} V_s \leq tV_{\max}$. We also assume that there are n_t^p document pairs in the training dataset. As we only use the independent observed pairs, it is easy to verify that $n_t^p \leq \sum_{s=1}^{t} o_t/2 \leq kt/2$.

Following Definition 4.1 in [54], we define the neural tangent kernel matrix $\mathbf{H}$ of the n_T query-document features $\{\boldsymbol{x}_i\}_{i=1}^{n_T}$ across T rounds. We also adopt the same assumption on the context $\{\boldsymbol{x}_i\}_{i=1}^{n_T}$ and the kernel matrix $\mathbf{H}$ with Assumption 4.2 in [54]. As the input $\{\mathbf{x}_i\}_{i=1}^{n_T}$ are based on manually crafted ranking features, such assumptions can be easily satisfied. Equipped with this assumption, it can be verified that with $\boldsymbol{\theta}_0$ initialized as in Algorithm 1, $f(\mathbf{x}_i;\boldsymbol{\theta}_0) = 0$ for any $i \in [n_t]$. It is also well known that the sigmoid function σ is continuously differentiable, Lipschitz with constant $k_\mu = 1/4$ and $c_\mu = \inf \dot{\sigma} > 0$.

In order to prove Lemma 3.3, we need the following lemmas, in addition to the technical lemmas, Lemma 5.1, Lemma B.2, Lemma B.3, Lemma B.4, Lemma B.5 and Lemma B.6 from [54]. The first lemma is based on the generalized linear bandit [16] and the analysis of linear bandit in [1]. For Lemma A.2 and Lemma A.3, we adopt it from the original paper based on our pairwise cross-entropy loss, and covariance matrix $\mathbf{A}_t$ on the pairwise feature vectors.

Lemma A.1. For any $t \in [T]$, with $\widehat{\boldsymbol{\gamma}}_t$ defined as the solution of the following equation,

$$\sum_{s=1}^{t-1} \sum_{(i,j)\in\Omega_s^{ind}} (\sigma(\langle \mathbf{g}_{ij}^{s,0}, \boldsymbol{\gamma}\rangle) - y_{ij}^s)\mathbf{g}_{ij}^{s,0} + m\lambda\boldsymbol{\gamma} = 0 \tag{A.1}$$

Then, with the pairwise noise ξ_{ij}^s satisfying Proposition 3.2, for any $(i,j) \in \Psi_t$, with probability at least $1-\delta_1$, we have,

$$\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t} \leq c_\mu^{-2}(\sqrt{\nu^2 \log(\det(\bar{\mathbf{A}}_t))/(\delta_1^2 \det(\lambda\mathbf{I}))} + \sqrt{\lambda}S)$$

Lemma A.2. There exist constants $\{\bar{C}_i\}_{i=1}^5 > 0$ such that for any $\delta > 0$, if for all $t \in [T]$, η and m satisfy

$$\sqrt{n_t^p/(m\lambda)} \geq \bar{C}_1 m^{-3/2} L^{-3/2} [\log(n_t L^2/\delta)]^{3/2},$$
$$\sqrt{n_t^p/(m\lambda)} \leq \bar{C}_2 \min\left\{L^{-6}[\log m]^{-3/2}, \left(m(\lambda\eta)^2 L^{-6}(n_t^p)^{-1}(\log m)^{-1}\right)^{3/8}\right\},$$
$$\eta \leq \bar{C}_3(m\lambda + n_t^p mL)^{-1},\ m^{\frac{1}{6}} \geq \bar{C}_4\sqrt{\log m}L^{7/2}(n_t^p)^{7/6}\lambda^{-7/6}(1+\sqrt{n_t^p/\lambda}),$$

then with probability at least $1-\delta$, $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2 \leq 2\sqrt{2n_t^p/(m\lambda)}$ and

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\|_2 \leq \bar{C}_5 m^{-2/3}\sqrt{\log m}L^{7/2}(n_t^p)^{7/6}\lambda^{-7/6}(1+\sqrt{n_t^p/\lambda}) + (1-\eta m\lambda)^{J/2}\sqrt{n_t^p/(m\lambda)}.$$

Lemma A.3. There exist constants $\{C_i^\epsilon\}_{i=1}^5 > 0$ such that for any $\delta > 0$, if m satisfies that

$$C_1^\epsilon m^{-3/2}L^{-3/2}[\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq C_2^\epsilon L^{-6}[\log m]^{-3/2}$$

with τ as the upper bound of $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|$, then with probability at least $1-\delta$, for any $t \in [T]$, we have

$$\|\mathbf{A}_t\|_2 \leq \lambda + C_3^\epsilon n_t^p L,$$
$$\|\bar{\mathbf{A}}_t - \mathbf{A}_t\|_F \leq C_4^\epsilon n_t^p\sqrt{\log(m)}\tau^{1/3}L^4,$$
$$\left|\log\frac{\det(\bar{\mathbf{A}}_t)}{\det(\lambda\mathbf{I})} - \log\frac{\det(\mathbf{A}_t)}{\det(\lambda\mathbf{I})}\right| \leq C_5^\epsilon (n_t^p)^{5/3}\lambda^{-1/6}\sqrt{\log(m)}m^{-1/6}L^4$$

where $\bar{\mathbf{A}}_t = \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \frac{1}{m}\mathbf{g}_{i'j'}^0 \mathbf{g}_{i'j'}^{0\top} + \lambda\mathbf{I}$. The constants $\{C_i^\epsilon\}_{i=1}^5$ can be constructed based on the constants in the technical lemmas, Lemma B.4, B.5 and B.6 from [54].

Proof of Lemma 3.3. We first bound the estimated pairwise order based on the Lipschitz continuity:

$$\begin{aligned}&\left|\sigma(f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1})) - \sigma(h(\mathbf{x}_i^t) - h(\mathbf{x}_j^t))\right| \\ \leq& k_\mu\left|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}) - \left(h(\mathbf{x}_i^t) - h(\mathbf{x}_j^t)\right)\right|\end{aligned}$$

According to Lemma 5.1 in [54] and $f(\mathbf{x};\boldsymbol{\theta}_0) = 0$, we have,

$$\begin{aligned}f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - h(\mathbf{x}_i^t) =& f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_i^t;\boldsymbol{\theta}_0) - \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0\rangle \\ &+ \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0\rangle - \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle \\ &+ \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle - \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_0), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle.\end{aligned}$$

Based on the triangle inequality, we have,

$$\begin{aligned}&\left|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}) - \left(h(\mathbf{x}_i^t) - h(\mathbf{x}_j^t)\right)\right| \\ \leq& \left|\langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - \mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*\rangle\right| \\ &+ \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2\left(\|\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_0)\|_2 + \|\mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}) - \mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_0)\|_2\right) \\ &+ \left|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_i^t;\boldsymbol{\theta}_0) - \langle \mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0\rangle\right| \\ &+ \left|f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_0) - \langle \mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0\rangle\right| \\ \leq& 2C_1^t\tau^{4/3}L^3\sqrt{m\log m} + 2C_2^t S\sqrt{\log m}\tau^{1/3}L^3\sqrt{mL} + \left|\langle \mathbf{g}_{ij}^t, \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*\rangle\right|,\end{aligned}$$

where C_1^t and C_2^t are positive constants, S is the upper bound of $\sqrt{\mathbf{h}^\top\mathbf{H}\mathbf{h}}$. The last inequality is due to Lemma 5.1, Lemma B.4, Lemma B.5, Lemma B.6 in [54], with τ as the upper bound of $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2$.

Now we start to bound the last term $\left|\langle \mathbf{g}_{ij}^t, \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*\rangle\right|$.

$$\left|\langle \mathbf{g}_{ij}^t, \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*\rangle\right| \leq |\langle \mathbf{g}_{ij}^t, \boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\rangle| + \|\mathbf{g}_{ij}^t\|\|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\| \tag{A.2}$$

For the first term, we have the following analysis.

$$\begin{aligned}&|\langle \mathbf{g}_{ij}^t, \boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\rangle| \\ \leq& \|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\sqrt{(1 + \|\mathbf{A}_t - \bar{\mathbf{A}}_t\|_2/\lambda)}\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t} \\ \leq& \sqrt{1 + C_3^t m^{\frac{-1}{6}}\sqrt{\log m}L^4 t^{7/6}\lambda^{-7/6}} \cdot \|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t}\|\frac{1}{\sqrt{m}}\mathbf{g}_{ij}^t\|_{\mathbf{A}_t^{-1}},\end{aligned}$$

where the first inequality is due to the fact that $\mathbf{x}^\top\mathbf{P}\mathbf{x} \leq \mathbf{x}^\top\mathbf{Q}\mathbf{x} \cdot \|\mathbf{P}\|_2/\lambda_{\min}(\mathbf{Q})$, and $\lambda_{\min}(\bar{\mathbf{A}}_t) \geq \lambda$, the third inequality is based on Lemma B.3 in [54] with $\|\mathbf{A}_t - \bar{\mathbf{A}}_t\|_2 \leq \|\mathbf{A}_t - \bar{\mathbf{A}}_t\|_F$. According to Lemma A.1, with probability $1-\delta_1$, we have

$$\begin{aligned}&\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t} \\ \leq& c_\mu^{-2}(\sqrt{\nu^2\log(\det(\mathbf{A}_t))/(\delta_1^2\det(\lambda\mathbf{I})) + C_4^t m^{-1/6}\sqrt{\log m}L^4 t^{5/3}\lambda^{-1/6}} + \sqrt{\lambda}S),\end{aligned}$$

where the inequality is based on Lemma B.3. For the second term of Eq (A.2), it can be bounded according to Lemma B.5 in [54] and Lemma A.2. By chaining all the inequalities, with $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\| \leq \tau \leq 2\sqrt{2n_t^p/(m\lambda)}$, and the satisfied m and η, we complete the proof. □

B PROOFS OF LEMMA 4.2

The following lemma is derived from random matrix theory. We adapted it from Equation (5.23) of Theorem 5.39 from [45].

Lemma B.1. Let $\mathbf{M} \in \mathbb{R}^{N\times p}$ be a matrix whose rows $\mathbf{M}_i$ are independent sub-Gaussian isotropic random vectors in $\mathbb{R}^p$ with parameter ρ, namely $\mathbb{E}[\exp(\mathbf{g}_{i'j'}^\top(\mathbf{M}_i - \mathbb{E}[\mathbf{M}_i])/\sqrt{m}] \le \exp(\rho^2\|\mathbf{g}_{i'j'}/\sqrt{m}\|^2/2)$ for any $\mathbf{g}_{i'j'} \in \mathbb{R}^p$. Then, there exist positive universal constants C_1 and C_2 such that, for every $t \ge 0$, the following holds with probability at least $1 - 2\exp(-C_2t^2)$, where $v = \rho(C_1\sqrt{p/N} + t/\sqrt{N})$: $\|\frac{1}{N}\mathbf{M}^\top\mathbf{M} - \mathbf{I}_p\| \le \max\{v, v^2\}$.

Proof of Lemma 4.2. At initialization, DNNs are equivalent to Gaussian processes in the infinite-width limit. With $\mathbf{\Sigma} = \mathbb{E}[\mathbf{g}_{ij}^0 \mathbf{g}_{ij}^{0\top}]$ as the second moment matrix, define $\mathbf{Z} = \mathbf{\Sigma}^{-1/2}\mathbf{X}$, where $\mathbf{X}$ is a random vector drawn from the same distribution v. Then $\mathbf{Z}$ is isotropic, namely $\mathbb{E}[\mathbf{Z}\mathbf{Z}^\top] = \mathbf{I}_p$. Define $\mathbf{D} = \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \mathbf{Z}_{i'j'}^s \mathbf{Z}_{i'j'}^{s\top}$, where $\mathbf{Z}_{i'j'}^s = \mathbf{\Sigma}^{-1/2}\mathbf{g}_{i'j'}^{s,0}$. It is trivial to have $\mathbf{D} = \mathbf{\Sigma}^{-1/2}(\bar{\mathbf{A}}_t - \lambda\mathbf{I})\mathbf{\Sigma}^{-1/2}$. From Lemma B.1, we know that for any l, with probability at least $1 - 2\exp(-C_2l^2)$, $\lambda_{\min}(\mathbf{D}) \ge n_t - C_1\sigma^2 n_t - \sigma^2 l\sqrt{n_t}$, where σ is the sub-Gaussian parameter of $\mathbf{Z}$, which is upper-bounded by $\|\mathbf{\Sigma}^{-1/2}\| = \lambda_{\min}(\mathbf{\Sigma})$, and $n_t = \sum_{s=1}^{t-1}|\Omega_s^{ind}|$ represents the number of pairwise observations so far. Thus, we can rewrite the above inequality which holds with probability $1-\delta_2$ as $\lambda_{\min}(\mathbf{D}) \ge n_t - \lambda_{\min}^{-1}(\mathbf{\Sigma})(C_1 n_t + l\sqrt{n_t})$, and: $\lambda_{\min}(\bar{\mathbf{A}}_t - \lambda\mathbf{I}) = \min_{x\in\mathbb{B}^p} x^\top\mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{\Sigma}^{1/2}x \ge \lambda_{\min}(\mathbf{\Sigma})n_t - C_1 n_t - C_2\sqrt{n_t\log(1/\delta_2)}$

Under event E_t, based on the definition of ω_t in Section 3, we know that for any document i and j at round t, $(i,j) \notin \omega_t$ if and only if $\sigma(f_{ij}^t) - CB_{ij}^t \le 1/2$ and $\sigma(f_{ji}^t) - CB_{ji}^t \le 1/2$. For a logistic function, we know that $\sigma(s) = 1 - \sigma(-s)$. Therefore, according to Lemma 3.3, we can conclude that $(i,j) \notin \omega_t$ if and only if $|\sigma(f_{ij}^t) - 1/2| \le CB_{ij}^t$; and accordingly, $(i,j) \in \omega_t$, when $|\sigma(f_{ij}^t) - 1/2| > CB_{ij}^t$. According to the discussion above, at round t, we have the probability that the estimated preference between document i and j in an uncertain rank order, i.e., $(i,j) \notin \omega_t$,

$$\mathbb{P}\big((i,j)\notin\omega_t\big) \le \mathbb{P}\left(\Delta_{\min} - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \le CB_{ij}^t\right).$$

Based on Lemma 3.3, the probability can be further bounded by

$$\begin{aligned}&\mathbb{P}\left(\Delta_{\min} - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \le CB_{ij}^t\right)\\ \le&\mathbb{P}\left(\|\mathbf{W}_t\|_{\mathbf{A}_t^{-1}} \ge \frac{c_\mu}{2k_\mu}\Big(\frac{(\Delta_{\min} - 2\epsilon(m))}{||\mathbf{g}_{ij}^t/\sqrt{m}||_{\mathbf{A}_t^{-1}}} - \alpha_t\Big)\right).\end{aligned}$$

where $\mathbf{W}_t = \sum_{s=1}^t\sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s \mathbf{g}_{i'j'}^s$. For the right-hand side, we know that $\lambda_{\min}(\mathbf{A}_t) \ge \lambda_{\min}(\bar{\mathbf{A}}_t) + \|\mathbf{A}_t - \bar{\mathbf{A}}_t\| \ge \lambda_{\min}(\bar{\mathbf{A}}_t - \lambda\mathbf{I}) + \lambda + \|\mathbf{A}_t - \bar{\mathbf{A}}_t\|$. With some positive constants $\{C_i^u\}_{i=1}^5$, for $t \ge t' = \big(C_1^u + C_2^u\sqrt{\log(1/\delta_2)} + C_3^u V_{\max}\big)^2 + C_4^u\log(1/\delta_1) + C_5^u$, as $n_t > t$, we have $n_t - \sqrt{n_t}\big(C_1^u + C_2^u\sqrt{\log(1/\delta_2)} + C_3^u V_{\max}\big) > C_4^u\log(1/\delta_1) + C_5^u$. Hence, when $t \ge t'$, the right-hand side of the inequality is positive. Therefore, we have:

$$\mathbb{P}\left(\Delta_{\min} - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \le CB_{ij}^t\right) \le \frac{C_u\log(1/\delta_1)}{(\Delta_{\min} - 2\epsilon(m))^2}\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}}^2.$$

This completes the proof. □

C PROOF OF THEOREM 4.4

Lemma C.1. There exist positive constants $\{C_i\}_{i=1}^2$ such that for any $\delta \in (0,1)$, if $m \ge \bar{C}_2\max\{T^7\lambda^{-7}L^{21}(\log m)^3, n_T^6L^6(\log(TV_{\max}L^2/\delta))^{3/2}\}$, and $\eta \le \bar{C}_1(TmL + m\lambda)^{-1}$, with probability at least $1-\delta$, we have

$$\sum_{t=1}^T\sum_{(i',j')\in\Omega_t}\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}} \le 2\log\frac{\det\mathbf{A}_T}{\det\lambda\mathbf{I}} \le \widetilde{d}\log(1 + TV_{\max}^2/\lambda) + 1$$

where $\widetilde{d}$ is defined as the effective dimension of $\widehat{\mathbf{H}}$.

Proof of Theorem 4.4. With δ_1 and δ_2 defined in the previous lemmas, we have with probability at least $1-\delta_1$, the T-step regret is upper bounded as:

$$R_T \le t' * V_{\max}^2 + (T - t')\delta_2 V_{\max}^2 + (1-\delta_2)\sum_{t=t'}^T r_t \tag{C.1}$$

When event E_t and the event defined in Lemma 3.3 both occur, the instantaneous regret at round t is bounded by $r_t = \mathbb{E}\big[K(\tau_s, \tau_s^*)\big] \le \mathbb{E}[U_t]$, where U_t denotes the number of uncertain rank orders under the ranker at round t. As the ranked list is generated by topological sort on the certain rank orders, the random shuffling only happens between the documents that are in uncertain rank orders, which induce regret in the proposed ranked list. In each round of result serving, as the model $\boldsymbol{\theta}_t$ would not change until the next round, the expected number of uncertain rank orders can be estimated by summing the uncertain probabilities over all possible pairwise comparisons under the current query q_t, e.g., $\mathbb{E}[U_t] = 1/2\sum_{(i,j)\in\Psi_t}\mathbb{P}((i,j)\notin\omega_t)$. Based on Lemma 4.2, the cumulative number of mis-ordered pairs can be bounded by the probability of observing uncertain rank orders in each round, which shrinks with more observations become available over time,

$$\begin{aligned}\mathbb{E}\Big[\sum_{s=t'}^T U_t\Big] \le&\mathbb{E}\big[1/2\sum_{s=t'}^T\sum_{(i',j')\in\Psi_s}\mathbb{P}((i',j')\notin\omega_t)\big]\\ \le&\mathbb{E}\Big[\sum_{s=t'}^T\sum_{(i',k')\in\Psi_s} C_6^u\log(1/\delta_1)\|\mathbf{g}_{i'j'}^t\|_{\mathbf{A}_t^{-1}}^2/\Delta_{\min}^2\Big].\end{aligned}$$

Because $\mathbf{A}_t$ only contains information of observed document pairs so far, our algorithm guarantees the number of mis-ordered pairs among the observed documents in the above inequality is upper bounded. To reason about the number of mis-ordered pairs in those unobserved documents (i.e., from o_t to L_t for each query q_t), we leverage the constant p^*, which is defined as the minimal probability that all documents in a query are examined over time,

$$\begin{aligned}&\mathbb{E}\Big[\sum_{t=t'}^T\sum_{(i',j')\in\Psi_t}\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\Big]\\ \le&p^{*-1}\mathbb{E}\Big[\sum_{t=t'}^T\sum_{(i',j')\in\Psi_t}\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}}\mathbf{1}\{o_t = V_t\}\Big]\end{aligned}$$

Besides, we only use the independent pairs, Ω_t^{ind} to update the model and the corresponding $\mathbf{A}_t$ matrix. Therefore, to bound the regret, we rewrite the above equation as:

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=t'}^T\sum_{(i',j')\in\Psi_t}\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{-1}}^2\right]\\ \le&\mathbb{E}\left[\sum_{t=t'}^T\sum_{(i',j')\in\Omega_t^{ind}}\left(L_t\|\frac{\mathbf{g}_{i'j'}^t}{\sqrt{m}}\|_{\mathbf{A}_s^{-1}}^2 + \sum_{k\in[V_t]\setminus\{i',j'\}}\frac{2\mathbf{g}_{i'k}^{t\top}\mathbf{A}_t^{-1}\mathbf{g}_{j'k}^t}{m}\right)\right]\\ \le&\mathbb{E}\left[\sum_{t=t'}^T\left(\sum_{(i',j')\in\Omega_t^{ind}}L_t\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_s^{-1}}^2 + 2C_3^z V_{\max}^2L^2/\lambda_{\min}(\mathbf{A}_t)\right)\right]\end{aligned}$$

where the last inequality is due to Lemma B.6 in [54]. According to the analysis of $\lambda_{\min}(\mathbf{A}_t)$ and $\lambda_{\min}(\bar{\mathbf{A}}_t)$, the convergence rate the above upper bound is faster than the self-normalized term. Hence, by chaining all the inequalities, we have with probability at least $1-\delta_1$, the regret satisfies,

$$\begin{aligned}R_T \le&R' + (1-\delta_2)C_6^u\log(1/\delta_1)(w + V_{\max}(\widetilde{d}\log(1 + TV_{\max}^2/\lambda) + 1)/\Delta_{\min}^2)\\ \le&R' + (C_1^r\log(1/\delta_1)\widetilde{d}\log(1 + TV_{\max}^2/\lambda) + C_2^r)(1-\delta_2)/(\Delta_{\min}^2p^*)\end{aligned}$$

where $\{C_i^r\}_{i=1}^2$ are positive constants, $R' = t'V_{\max}^2 + (T-t')\delta_2V_{\max}^2$. By choosing $\delta_1 = \delta_2 = 1/T$, the theorem shows that the expected regret is at most $R_T \le O(\widetilde{d}\log^2(T))$. □