Learning to Rank with Variable Result Presentation Lengths

Norman Knyazev
Radboud University
Nijmegen, The Netherlands
norman.knyazev@ru.nl

Harrie Oosterhuis
Radboud University
Nijmegen, The Netherlands
harrie.oosterhuis@ru.nl

Abstract

Learning to rank (LTR) methods generally assume that each document in a top-K ranking is presented in an equal format. However, previous work has shown that users' perceptions of relevance can be changed by varying presentations, i.e., allocating more vertical space to some documents to provide additional textual or image information. Furthermore, presentation length can also redirect attention, as users are more likely to notice longer presentations when scrolling through results. Deciding on the document presentation lengths in a fixed vertical space ranking is an important problem that has not been addressed by existing LTR methods.

We address this gap by introducing the variable presentation length ranking task, where simultaneously the ordering of documents and their presentation length is decided. Despite being a generalization of standard ranking, we show that this setting brings significant new challenges: Firstly, the probability ranking principle no longer applies to this setting, and secondly, the problem cannot be divided into separate ordering and length selection tasks.

We therefore propose VLPL – a new family of Plackett-Luce list-wise gradient estimation methods for the joint optimization of document ordering and lengths. Our semi-synthetic experiments show that VLPL can effectively balance the expected exposure and attractiveness of all documents, achieving the best performance across different ranking settings. Furthermore, we observe that even simple length-aware methods can achieve significant performance improvements over fixed-length models. Altogether, our theoretical and empirical results highlight the importance and difficulties of combining document presentation with LTR.

CCS Concepts

• **Information systems** → **Learning to rank**; **Presentation of retrieval results**.

Keywords

Learning to Rank, Variable Length Presentations, Policy Learning

ACM Reference Format:
Norman Knyazev and Harrie Oosterhuis. 2025. Learning to Rank with Variable Result Presentation Lengths. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3730020

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730020

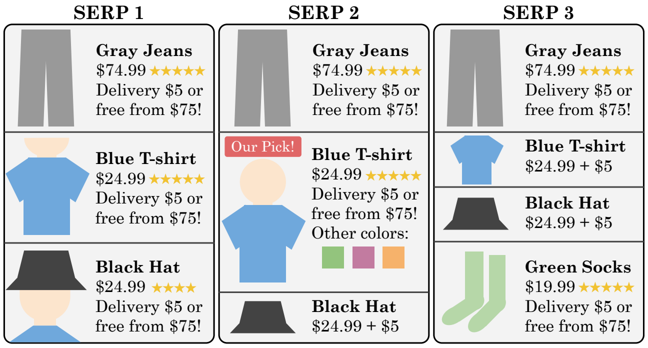

Figure 1: Rankings with variable presentation length. Compared to SERP 1, SERP 2 presents the second item with more vertical space (length), allowing more information to be displayed. In SERP 3, the second and third item are both presented with less vertical space, resulting in less information for each, but allowing an additional item to be displayed.

1 Introduction

Ranking models generally aim to identify the most relevant documents from a set of candidates and present them to the user in the order of decreasing relevance on a search engine result page (SERP) [20]. However, the presentation of individual documents within a SERP is rarely considered when choosing the document ordering, with the documents usually simply modeled as fixed units of information or relevance. Nevertheless, in many cases the platforms can choose how each document is presented to the user, which in turn affects how the user perceives and interacts with those documents. We call this setting where both the order of the documents as well as their presentation lengths are chosen as the variable document presentation length setting – see Figure 1 for an example. This setting is characterized by the trade-off between presenting more information for displayed documents – potentially making them more attractive – or displaying more documents on the SERP – potentially including more relevant results.

Earlier research has shown that increasing the size of individual SERP results to include additional textual information may make the user more likely to click through to the full document [25], whilst shorter results may discourage the users from clicking [19]. Similarly, in e-commerce, adding images and product details has also been found to increase the engagement with those items [2, 41].

Making a search result larger may reduce the probability that a user will miss it when browsing [10], and thereby, alleviate some position bias on that result [24]. However, as noted by Pathak et al. [30, 31], increasing the vertical size of some results generally means that fewer results fit on the SERP [19]. In general, increasing the presentation size of an individual retrieved document also means that the documents below it are pushed down by the same amount. This, in turn, may make them less likely to be observed by the user and lead to an increase in the effects of position bias [9, 16, 17]. Hence this setting requires a balancing of document ordering and their presentation lengths. To the best of our knowledge, no existing method can simultaneously optimize a ranking and variable presentation lengths.

In this work, we formally introduce and study the variable document presentation length setting. Our analysis reveals challenges that are not present in standard ranking: most importantly, the probability ranking principle (PRP) does not apply, and as a result, the optimal placement strategy cannot be separated into independent ranking and length-selection strategies. In order to tackle this problem, we extend the Plackett-Luce (PL) ranking approach to optimize both the order and presentation lengths of documents. Our novel model, the variable length Plackett-Luce (VLPL) distribution π_{VLPL}, represents each choice as a document placement paired with a specific length. We build on the PL-Rank family of gradient estimators [26, 27] to introduce the novel VLPL−1 and VLPL−2 estimators. These efficiently estimate the gradient of π_{VLPL}, with VLPL−2 having a higher sample-efficiency. Moreover, as we show that the gradient estimation can be leveraged during model training as either in-processing or post-processing [6], in total we propose four distinct methods for jointly optimizing ordering and length. Our semi-synthetic experiments demonstrate that our approaches achieve a large and consistent improvement over baseline methods. Overall, our findings underscore the need for ranking optimization methods with a more holistic view of result presentation that goes beyond the "*ten blue links*", as some links may look very different from others, and perhaps there should not be ten.

2 Related Work

Learning to rank (LTR) models aim to identify the most relevant documents and present them to the user, most commonly in the form of a ranked list. This is usually done by scoring each document and then sorting the documents based on their scores. According to the PRP [34], a scoring function should reflect the probability of each document being relevant to the user to produce optimal rankings. In recent years, the Plackett-Luce family of list-wise methods [32] have gained popularity, owing to their simplicity, robustness and interpretability [3, 26, 27]. Importantly, however, these and other widely used ranking models [3–5, 12, 14, 26, 27, 38, 40] only view ranking from the lens of document ordering, not concerning themselves with any further choices about document presentation.

A large body of work has shown that result presentation may significantly affect how users perceive and interact with each document. Maxwell et al. [25] observed that users are more likely to interact with a result on a SERP if it is presented with a longer snippet. This is also consistent with the observations of Kohavi [19], who found that allocating the same amount of screen space to fewer but larger and more interactive ads led to significant increases in user engagement and conversions. Marcos et al. [24] reported that lower-ranking results with larger and more interactive snippets are noticed earlier and for longer compared to other snippets, with snippets containing images being particularly salient. You et al. [41] also found that the image of a SERP result may have a drastic impact on how likely the user is to find the document attractive [1, 8, 42].

On the other hand, the number of documents visible to the user at a given time also significantly affects user interactions [9, 11, 14, 16, 17, 28, 35, 36]. Kelly and Azzopardi [16] showed that having more results on each SERP leads to users interacting with more documents, whilst Kim et al. [17] found that the need to scroll, in contrast with pagination, significantly reduces the speed at which the users are able to browse the result list. Taken together, the above findings suggest that decreasing snippet length, which allows to include more results on a single SERP, may lead to a higher number of interactions with the full ranking. Furthermore, the above results are also consistent with the findings of Joachims [14] on position bias [9, 10, 21], who saw that the probability that the user observes a document decreases with the position of that document, with particularly significant drops after the end of each page.

Our work is most related to that of Mao et al. [23] and Pathak et al. [30, 31]. Mao et al. aim to learn the causal graph based on the features of the SERP, which also includes document rank and document height. This causal graph is then used to correct for the impact of those variables on the click data in order to learn the intrinsic document relevance. However, the authors assume a fixed length for each document and do not learn how to the document lengths interact across different documents. Pathak et al. [31], on the other hand, focused on the setting where snippets can either contain the headline, summary, image, or any combination of the above, with each element taking up additional space on the screen. They find that changing the presentation of a document may also change where it should be placed in the ranking, particularly due to the increase in time the users spend examining the longer presentations. Nevertheless, the authors focus on the setting where the document lengths are chosen prior to determining the document order, whilst in Section 4.2 we prove that this can lead to suboptimal solutions. Finally, their approach assumes that user's examination behavior can be measured separately for each document, and does not provide a ranking function for previously unseen queries.

3 Background: The Standard Ranking Setting

The core LTR task is to find a ranking $y = [d_1, d_2, \dots, d_K]$ of length K that orders documents by decreasing relevance ρ_i for a given query q and document set $d \in D$. To assess the quality of a ranking, per-position discounting factors $\theta(i)$ are commonly used, e.g., one-over-rank and DCG weighting. Stochastic ranking models [26, 37] can optimize a policy π that provides a probability distribution over rankings to maximize an objective of the form:

$$\mathcal{R}_{\text{LTR}}(q) = \mathbb{E}_{\pi(y)}\left[\sum_{i=1}^{K} \theta(i)\rho(d_i)\right] = \sum_{y} \pi(y) \sum_{i=1}^{K} \theta(i)\rho(d_i). \quad (1)$$

In click modeling [8], the reward ρ may often represent the perceived relevance or *attractiveness* of the document, i.e., the probability the user will click on the document when exposed to it.
The policy of a Plackett-Luce LTR model [26, 32] is defined as:

$$\pi_{\text{PL}}(y) = \prod_{i=1}^{K} \pi_{\text{PL}}(y_i \mid y_{1:i-1}) = \prod_{i=1}^{K} \frac{e^{m(d_i)} \mathbb{1}\left[d_i \notin y_{1:i-1}\right]}{\sum_{d'=1}^{|D|} e^{m(d')} \mathbb{1}\left[d' \notin y_{1:i-1}\right]}, \quad (2)$$

where $m(d)$ is the score for document d under some scoring function m. In the Plackett-Luce distribution, documents are sequentially sampled from a softmax distribution without replacement.

Furthermore, the sampling probabilities in each position only depend on the scores of unplaced documents. Whilst the optimization of π_{PL} policies can be done with standard policy gradient methods (i.e., via REINFORCE [39]), Oosterhuis [26, 27] has proposed the PL-Rank family of methods, specialized to efficiently estimate gradients of Plackett-Luce distributions for LTR optimization.

Nevertheless, the standard ranking setting is not applicable if documents can be presented with different lengths. In particular, this is mainly due to two limitations of its objective (Equation 1): (1) the rank weights $\theta(i)$ only consider the ranks of documents and not their presentation lengths; and (2) a ranking always consists of K placed documents. However, as discussed in Section 2, the length of a document can affect the attractiveness and the amount of attention it receives from users [24, 25]. Therefore, $\theta(i)$ should adapt according to the attention that is expected from users [13] and, similarly, attractiveness $\rho(d_i)$ should also change with length. Furthermore, changing document presentation length should affect the number of documents that can be presented on a single SERP [31]. None of these aspects are captured by the standard ranking setting.

4 Variable Presentation Length Problem Setting

This section introduces a formalization of the variable presentation length ranking setting. Subsequently, we also discuss some of its theoretical implications.

4.1 Slot based ranking

To capture the unique characteristics of this setting, we represent each placement as a choice of document d and its length l, with a full ranking being $y = [(d_1, l_1), (d_2, l_2), \ldots, (d_{|y|}, l_{|y|})]$, where each d can only be placed once. Crucially, we assume that each ranking consists of K positions or slots: $\text{len}(y) = \sum_{j=1}^{|y|} l_i = K$. However, unlike the standard setting, each position now refers to a physical space in the ranking, not a moment of placement. The lengths l thus represent the number of slots each document occupies in the ranking. We also denote the first slot d_i occupies as $s_i = 1 + \sum_{j=1}^{i-1} l_i$. The above changes then allow us to define a new optimization objective called the expected attractiveness (EA) of a ranking:

$$\mathcal{R}(q) = \mathbb{E}_y \left[\sum_{i=1}^{|y|} \theta(s_i, l_i)\rho(d_i, l_i) \right] = \sum_y \pi(y) \sum_{i=1}^{|y|} \theta(s_i, l_i)\rho(d_i, l_i). \quad (3)$$

An important consequence of the above differences with standard ranking is that the number of unique documents $|y|$ in a ranking can be lower than K. Our task is then to choose the document-length pairs that maximize the weighted attractiveness in Equation 3 without exceeding the slot budget K or duplicate document placements. In contrast with standard ranking, both the position weight θ and the reward ρ depend on the presentation length l. For this work, we assume that ρ is fixed for each (d, l), i.e., for document d each presentation length is associated with a single relevance value, which is the expected click-through rate (CTR) of d when exposed to the user. Finally, to make our analysis and approach tractable, we assume a maximum document length of L slots. Naturally, document lengths may in reality be adjustable even up to a single pixel or may instead be fixed (but distinct) for each document. We consider both of these important next steps to be addressed by future work.

Finally, we note that our proposed setting has some overlap with that of Pathak et al. [31]: they also capture that varying document presentation lengths can lead to different number of items appearing above the fold. See Section 2 for a further discussion.

4.2 Theoretical analysis on optimal rankings

A key aspect of our problem setting is the trade-off between extending document lengths and increasing the number of documents on the SERP. Importantly, this trade-off is present even when the attractiveness of a document is independent of its length. As illustrated in Table 1, in a simple three-slot ranking with documents A, B and C, both rankings $[(A, 2), (B, 1)]$ and $[(B, 1), (A, 2)]$ are optimal in terms of EA under different discount weights θ. Notably, in this example, under θ_2 extending the second item into the third slot makes the user slightly more likely to observe it compared to the first item, and thus, placing the more relevant item second leads to a higher value of the EA metric. However, the observation that under θ_2 the most attractive document A should be placed second, contradicts the PRP [34], which states that placements should be in decreasing order of relevance. We thus conclude that PRP does not hold in the variable document presentation length setting.

Moreover, first choosing document lengths to maximize the exposure can similarly lead to suboptimal results. Whilst in the above example the optimal length for A is 2 under both values of θ, had document B not been available (i.e., only options are A and C), the optimal strategy in both cases would have instead been to present document A at length 3. Therefore, the above example proves that the task of choosing document length cannot be separated into separate ranking and document-specific length selection tasks:

Theorem 1. *Let $\pi^*(l, r \mid d, D)$ be the optimal policy for jointly choosing r the rank of document d and l its presentation length. The optimal policy cannot be separated into a separate ranking policy and length selection policy:*

$$\forall D, \nexists(\pi_r, \pi_l), \forall(d, l, r) : \ \pi^*(l, r \mid d, D) = \pi_r(r \mid d, D)\pi_l(l \mid d). \quad (4)$$

Proof. Assume documents and position weights as in Table 1. The optimal length policy for document A is then $\pi_l(l = 2 \mid A) = 1$. If we only change the relevance of B to 0, the optimal length policy for A is $\pi_l(l = 3 \mid A) = 1$, thus there is no universal optimal π_l. □

At first glance, it may seem feasible to solve the task with dynamic programming, but a further difficulty is that the optimal ranking for K slots is not necessarily an optimal sub-ranking for $K' > K$ slots. This can be also seen in Table 1: B should be placed ahead of A under θ_2 and $K = 3$, but the order is reversed under $K = 2$ (i.e., $[(A, 1), (B, 1)]$). Nevertheless, optimal rankings should vary document lengths due to the changes in exposure (and attractiveness) arising in this setting. For instance, we can see that under $K = 3$ the optimal single-length solutions $[(A, 1), (B, 1), (C, 1)]$, $[(A, 2)]$ and $[(A, 3)]$ are worse than the best mixed document length solution. Finally, we note that, whilst not shown in the above example, the optimal orderings may again be different for other values of ρ that maintain the relative order $A \succ B \succ C$ but not the ratio of their ρ values, e.g., if $\rho(B) = 0.01$ then $(A, 3)$ would always be preferred.

Altogether, whilst it appears crucial to present documents at different lengths and choose their ordering accordingly, the above

Table 1: The impact of discount weights on optimal ordering. For three documents A, B, C with respective relevances 1.0, 0.6 and 0 (left, same for all lengths l), different choices of discount weights θ_1 (center left) and θ_2 (center right) with $K = L = 3$ lead to different expected rewards for different rankings (right). BAA represents ranking $y = [(B, 1), (A, 2)]$, whilst $AB(C)$ is equivalent to both AB and ABC. Best ranking per θ is underlined.

d	$\rho(d)$
A	1.0
B	0.6
C	0.0

θ_1	l=1	l=2	l=3
s=1	0.500	0.667	0.750
s=2	0.333	0.500	-
s=3	0.250	-	-

θ_2	l=1	l=2	l=3
s=1	0.631	0.815	0.895
s=2	0.500	0.715	-
s=3	0.431	-	-

	AAA	AAB	ABB	BAA	BBA	BBB	AA	AB(C)	BA
θ_1	0.750	$\underline{0.817}$	0.800	0.800	0.650	0.450	0.667	0.700	0.633
θ_2	0.895	1.074	1.060	$\underline{1.094}$	0.920	0.537	0.815	0.931	0.879

examples reveal that there appears no straightforward optimal solution – moreover, we proved that this task cannot be broken down into separate ranking and length-selection policies. Accordingly, the remainder of this work proposes a framework for jointly optimizing document ordering and lengths.

5 Method: Variable Length Plackett-Luce

5.1 A distribution over rankings and lengths

We propose to build a novel model on top of the existing PL model for ranking (Equation 2); whereas the original only models a distribution over rankings, our novel model captures a distribution over both rankings and presentation lengths.

Our novel model also follows sequential sampling of document placements, but instead of sampling single documents d, document-length pairs (d, l) are sampled, where l is the presentation length of d. To avoid a document from appearing multiple times in the same ranking, the sampling of a (d, l) pair is followed by the removal of all other pairs that involve d. For our notation, we use the binary indicator variable $\mathbb{1}[d_i \notin y_{1:i-1}] = \mathbb{1}[\nexists l : (d_i, l) \in y_{1:i-1}]$ to indicate earlier placements of a document. Another difference with standard ranking is that the lengths of placed documents affect how many can be displayed in the K slots: specifically, a pair (d, l) should not be sampled if fewer than l slots are unfilled at the end of a partial ranking. Thus, we also introduce the indicator $\tau(s_i + l_i) = \mathbb{1}[s_i + l_i \leq K + 1] = \mathbb{1}[\text{len}(y_{1:i-1}) + l_i \leq K]$ to indicate if there are enough slots left for the placement of a document-length pair.

This notation then lets us define our novel *VLPL distribution*:

$$\pi(y) = \prod_{i=1}^{|y|} \pi(y_i \mid y_{1:i-1}) = \prod_{i=1}^{|y|} \pi(d_i, l_i \mid y_{1:i-1}), \tag{5}$$

$$\pi(y_i \mid y_{1:i-1}) = \frac{e^{m(d_i,l_i)} \mathbb{1}\left[d_i \notin y_{1:i-1}\right] \tau(s_i + l_i)}{\sum_{d'=1}^{|D|} \sum_{l'=1}^{L} e^{m(d',l')} \mathbb{1}\left[d' \notin y_{1:i-1}\right] \tau(s_i + l')}. \tag{6}$$

Analogous to the standard PL ranking model (Equation 2), the placement probability $\pi(d, l \mid y_{1:i-1})$ only depends on the score $m(d, l)$ and the scores of other eligible unplaced (d', l') pairs. However, our distribution differs in what pairs are considered eligible, as placing one pair can make several other pairs ineligible to avoid duplicate placements and space constraints. This goes beyond the *sampling without replacement* of the standard PL model, which can also be seen a special case $L = 1$ of our novel variable length model.

5.2 Variable length Plackett-Luce optimization

Previous work on Plackett-Luce optimization has found efficient techniques for the standard Plackett-Luce model [26, 27]. Fortunately, this existing approach can be re-used for our novel variable length model. In particular, we can apply the derivation steps corresponding to Equations 11-23 of Oosterhuis [26], with only minor changes to match our model and objective. For brevity and due to its high similarity, we do not repeat the full derivation here, but report the following resulting gradient:

$$\frac{\delta}{\delta m}\mathcal{R}(q) = \sum_{d,l=1}^{|D|,L} \left[\frac{\delta}{\delta m} m(d, l)\right] \mathbb{E}_y \Bigg[\underbrace{\sum_{i=r(d,l)+1}^{|y|} \theta(s_i, l_i)\rho(d_i, l_i)}_{\text{future reward after placement}} \tag{7}$$

$$+ \underbrace{\sum_{i=1}^{r(d)} \pi(d, l \mid y_{1:i-1}) \left[\theta(s_i, l)\rho(d, l) - \sum_{x=i}^{|y|} \theta(s_x, l_x)\rho(d_x, l_x)\right]}_{\text{expected reward minus the risk of placement}} \Bigg] \tag{8}$$

Importantly, analogous to Oosterhuis [26], the gradient of $\mathcal{R}(q)$ w.r.t. the score of a single document-length pair can be broken down into three terms: (i) the future reward observed following the placement, (ii) plus the expected direct reward of the placement, (iii) minus the expected risk imposed by the pair before it is placed.

Unsurprisingly, compared to PL-Rank [26], the gradient in Equation 7 is calculated for each (d, l) pair, with the reward ρ and weights θ now depending on the observed document lengths. However, a subtle but crucial difference is that we also differentiate between $r(d, l)$, which is the rank at which d is observed at length l, and $r(d)$, which is the rank of d at any length (the rank function is set to $|y|$ if its input is not observed). The distinction between $r(d)$ and $r(d, l)$ is important as, due to the ranking size limit K, placing the document d at distinct lengths may change which (d', l') can follow it. Therefore, the future reward term for each length l of d is thus included only when that particular pair (d, l) is observed. The risk term and the expected reward, however, accumulate up until d has been shown at any length, proportional to $\pi(d, l \mid y_{1:i-1})$, after which all lengths of d become ineligible for placement. If a presentation length l is too long, these terms are 0, as $\tau(s_i + l) = 0 \rightarrow \pi(d, l \mid y_{1:i-1}) = 0$.

The gradient formulation in Equation 7 can be turned into an estimator by replacing the expectation with an average over sampled rankings. For brevity, we omit the full equation but name this approach VLPL-1. To our knowledge, VLPL-1 is the first method to jointly optimize rankings and lengths in the variable document presentation length setting.

The remainder of this section describes how to efficiently sample rankings from the VLPL distribution (Section 5.3). Subsequently, in Section 5.4, we propose a more sample-efficient estimator through a smarter sampling technique for re-using samples. Finally, we discuss how VLPL can be applied in an in-processing and post-processing manner (Section 5.5).

5.3 Efficiently sampling rankings

Whilst it may appear that variable length rankings are substantially more difficult to sample than standard rankings [26], this can actually be done with a comparable computational complexity. The trick is to first sample an invalid ranking y' and then transform it with a function f such that $f(y')$ is a valid variable length ranking. We start by defining y', which, in contrast with the variable length rankings as defined before, places d once at every possible length l, whilst ignoring the maximum ranking length K. Still, y' does not allow the placement of duplicate documents of the same length. This gives us the following probability for document placements:

$$\pi'(d, l \mid y'_{1:i-1}) = \frac{e^{m(d,l)} \mathbb{1}\left[(d, l) \notin y'_{1:i-1}\right]}{\sum_{d'=1}^{|D|} \sum_{l'=1}^{L} e^{m(d',l')} \mathbb{1}\left[(d', l') \notin y'_{1:i-1}\right]}. \tag{9}$$

Let d'_i and l'_i indicate the document at position i in y' and its length respectively. The probability of the ranking y' is then simply the product of each placement probability:

$$\pi'(y') = \prod_{i=1}^{|D|L} p(d'_i, l'_i \mid y'_{1:i-1}). \tag{10}$$

We note that every document-length pair is placed in y' and thus $|y| = DL$. Obviously, this makes y' an invalid variable length ranking as each document will appear L times in the ranking, and thus it is guaranteed to exceed the maximum length of K.

Our next step is to use a function f^K that turns y' into a valid ranking of length K, such that we end up with the same ranking distribution: $\pi(f(y') = y) = \pi(y)$. Our choice of f is a function that sequentially takes the first document in y' that does not violate either of the variable ranking constraints, until length K is reached:

$$\begin{aligned} f(y')_i = y'_j \text{ where } j = \arg\min_{j'} j' \in \{1, 2, \ldots, DL\} \\ \text{s.t. } d'_{j'} \notin f(y')_{1:i-1} \land \operatorname{len}(f(y')_{1:i-1}) + l'_{j'} \leq K. \end{aligned} \tag{11}$$

In other words, $f(y')_1 = y'_1$, then $f(y')_2$ is the next document that would not violate the validity of the ranking when placed, and so forth. To show that this yields the same distribution as π, we first note that the probability of a document placement: $\pi(f(y')_i = d \mid f(y')_{1:i-1})$ is the probability of it being a valid option and being ranked in front of all other valid options that are still available after $f(y')_{1:i-1}$. Following Luce's choice axiom [22], the ordering probabilities of the existing options in a PL model are not changed under the introduction of additional alternatives, i.e., $\Pr(a \succ b \mid \{a, b\}) = \Pr(a \succ b \mid \{a, b, c\})$. Thus, the availability of invalid options in y' does not change the relative ranking of valid options, and therefore $\pi(f(y')_i = (d, l) \mid f(y')_{1:i-1}) = \pi(y_i = (d, l) \mid y_{1:i-1})$ when $f(y')_{1:i-1} = y_{1:i-1}$. By extension, $\pi(f(y') = y) = \pi(y)$.

5.4 Extending VLPL for sample-efficiency

As previously suggested, VLPL-1 may have suboptimal sample-efficiency when calculating the future reward. In order to address this, we first start by defining the total placement reward as:

$$g(y, i) = \sum_{x=i}^{|y|} \theta(s_x, l_x)\rho(d_x, l_x). \tag{12}$$

This allows us to re-formulate Equation 7 as:

$$g(y, d, l) = g(y, r(d, l) + 1), \tag{13}$$

$$h(y, d, l) = \sum_{i=1}^{r(d)} \pi(d, l \mid y_{1:i-1}) \left[\theta(s_i, l)\rho(d, l) - g(y, i)\right]. \tag{14}$$

The expected future reward with our sampling scheme is then:

$$\mathbb{E}\left[\sum_{i=r(d,l)+1}^{|y|} \theta(s_i, l_i)\rho(d_i, l_i)\right] = \mathbb{E}\left[g(y, d, l)\right] = \mathbb{E}\left[g(f(y'), d, l)\right]. \tag{15}$$

To increase the sample-efficiency, we want to reuse each y'. For this purpose, we introduce the following function: $f(y' \leftarrow d, l)$, which is the same transformation as f, except when d is being selected its length is set to l (instead of the actually sampled length). Furthermore, we introduce the variable $l(d, y')$, denoting the length of the first instance of d that is included in $f(y')$. Our aim is then to use samples arising from the sampling distribution of $l(d, y')$ to estimate the expected value under $l(d, y') = l$. We therefore introduce the following importance weight:

$$p(d, l, y') = \frac{p\big(l(d, y') = l\big)}{\sum_{l'=1}^{L} p\big(l(d, y') = l'\big)} = \frac{e^{m(d,l)}\tau'(d, l, y')}{\sum_{l'=1}^{L} e^{m(d,l')}\tau'(d, l', y')}, \tag{16}$$

where $\tau'(d, l, y') = \mathbb{1}\left[\operatorname{len}(f(y')_{1:r(d|f(y'))-1}) + l \leq K\right]$ is true only for the lengths at which d can be placed when it is observed in $f(y')$. This then enables the following reformulation:

$$\mathbb{E}_{y'}\left[g(f(y'), d, l)\right] = \mathbb{E}_{y'}\left[p(d, l, y')g(f(y' \leftarrow d, l), d, l)\right], \tag{17}$$

That is, we can use rankings y' with any observed length $l(d, y')$, process them for the target length l with $f(y \leftarrow d, l)$ and reweigh the resulting future reward with $p(d, l, y')$. Repeating this for all valid lengths at each position yields the gradient estimate:

$$\begin{aligned} \frac{\delta}{\delta m}\mathcal{R}(q) = \sum_{d,l=1}^{|D|,k} \left[\frac{\delta}{\delta m} m(d, l)\right] \\ \cdot \mathbb{E}_{y'}\left[p(d, l, y')g(f(y' \leftarrow d, l), d, l) + h(f(y'), d, l)\right]. \end{aligned} \tag{18}$$

Furthermore, whilst calculating a separate $f(y' \leftarrow d, l)$ for each (d, l) pair may seem time-consuming, in practice it is actually not needed. We first define $\Delta \in [1 - L, 2 - L, \ldots, L - 1]$ as the difference between the target length l and observed length $l(d, y')$. Importantly, for a source ranking y' which after applying $f(y')$ includes (d_i, l_i) and (d_j, l_j), the adjusted rankings $f(y' \leftarrow d_i, l_i + \Delta)$ and $f(y' \leftarrow d_j, l_j + \Delta)$ in fact share the same tail, i.e., $z = \max(i, j) \rightarrow f(y' \leftarrow d_i, l_i + \Delta)_{z+1:} = f(y' \leftarrow d_j, l_j + \Delta)_{z+1:}$. We can thus use the same ranking to calculate the future reward for multiple such shifts, respectively reweighting each with $p(d_1, l_1 + \Delta, y'), p(d_2, l_2 + \Delta, y'), \ldots, p(d_{|f(y')|}, l_{|f(y')|} + \Delta, y')$. Moreover, these future placements would also be the same if instead we still placed d_i at the original observed length l_i but then reduced the total number of available slots in the ranking by Δ, i.e. $f(y' \leftarrow d_i, l_i + \Delta)_{z+1:} = f^{K-\Delta}(y', d_i, l_i)_{z+1:} = f^{K-\Delta}(y')_{z+1:}$, where $f^{K'}(y')$ produces a ranking y of length K' instead of K.

As Δ can take on values between $1 - L$ and $L - 1$ (corresponding to replacing the longest length with the shortest and vice-versa), the future reward always arises from one of $2L - 1$ rankings $f^{K+\Delta}(y')$. However, whilst $f^K(y' \leftarrow d, l + \Delta)$ and $f^{K-\Delta}(y' \leftarrow d, l)$ lead to the same placements after d, the slot positions s_i in the latter would be shifted down by Δ. To maintain the consistency in the reward between the two cases and to ensure the correct range of θ's inputs, we define the adjusted total reward as:

$$g'(y, i, \Delta) = \sum_{x=i}^{|y|} \mathbb{1}[s_x + \Delta \geq 1]\theta(s_x + \Delta, l_x)\rho(d_x, l_x). \tag{19}$$

We note that the indicator variable will always be one when calculating the future reward $g'(y, i+1, \Delta)$ for $1 \leq l(d_i, y') + \Delta \leq L$.

Finally, as shown by Oosterhuis [27], the computation of a gradient estimator for the PL distribution can be further accelerated by utilizing helper variables, which are first pre-computed and re-used for all positions. Along with applying Equation 19, we can thus express the terms in Equation 18 as:

$$\mathrm{PR}^{\Delta}_{y,i} = g'(f^{K-\Delta}(y), i, \Delta), \quad \mathrm{PR}^{\Delta}_{y,(d)} = \mathrm{PR}^{\Delta}_{y,r(d)+1}, \tag{20}$$

$$\mathrm{RI}_{y,i} = \sum_{x=1}^{\min(|y|,i)} \frac{\mathrm{PR}^0_{y,i}}{\sum_{d'=1}^{|D|}\sum_{l'=1}^{L} e^{m(d',l')}\mathbb{1}[d \notin y_{1:x-1}]\tau[s_x + l']}, \tag{21}$$

$$\mathrm{DR}_{y,i,l} = \sum_{x=1}^{\min(|y|,i)} \frac{\theta(s_x, l)}{\sum_{d'=1}^{|D|}\sum_{l'=1}^{L} e^{m(d',l')}\mathbb{1}[d \notin y_{1:x-1}]\tau[s_x + l']}, \tag{22}$$

$$\mathrm{RI}_{y,(d,l)} = \mathrm{RI}_{y,\min(r(d),r(l))}, \quad \mathrm{DR}_{y,(d,l)} = \mathrm{DR}_{y,\min(r(d),r(l)),l}, \tag{23}$$

where $r(l)$ is $\sup i : \mathrm{len}(f(y')_{1:i-1}) + l \leq K$, i.e., the last position at which a document of length l can be placed, and $r(d)$ is the rank of d in $y^0 = y$. This, in turn, allows us to reformulate Equation 7 as:

$$\begin{aligned}\frac{\delta}{\delta m}\mathcal{R}(q) = \sum_{d,l=1}^{|D|,L} \Big[\frac{\delta}{\delta m} m(d,l)\Big]\mathbb{E}_{y'}\Big[&p(d,l,y')\mathrm{PR}^{l-l(d,y')}_{y',(d)} \\ &+ e^{m(d,l)}\Big(\rho(d,l)\mathrm{DR}_{f(y'),(d,l)} - \mathrm{RI}_{f(y'),(d,l)}\Big)\Big].\end{aligned} \tag{24}$$

This gradient can be approximated using the sampling procedure for π' described in Subsection 5.3, yielding our estimator VLPL-2:

$$\begin{aligned}\frac{\delta}{\delta m}\mathcal{R}(q) = \sum_{d,l=1}^{|D|,L} \frac{\delta}{\delta m} m(d,l) \sum_{i=1}^{N}\frac{1}{N}\Big[&p(d,l,y'^{(i)})\mathrm{PR}^{l-l(d,y'^{(i)})}_{y'^{(i)},(d)} \\ &+ e^{m(d,l)}\Big(\rho(d,l)\mathrm{DR}_{f(y'^{(i)}),(d,l)} - \mathrm{RI}_{f(y'^{(i)}),(d,l)}\Big)\Big].\end{aligned} \tag{25}$$

This estimator has higher sample-efficiency than VLPL-1 as it shares the future reward across different lengths of the same document. The risk and the expected direct reward terms RI and DR are still calculated using the standard ranking $y = y^0$ of length K. The exact steps of our approach are also outlined in Algorithm 1.

Finally, we note that VLPL-1 can similarly be implemented with PL-Rank-3-like sample-efficiency by using the reward $\mathrm{PR}^0_{y',r(d,l)+1}$ in place of $\mathrm{PR}^{\Delta}_{y',r(d)+1}$ and setting $p(d,l,y')$ to 1.

Algorithm 1 VLPL-2 Gradient Estimator

```
Input: Items: $D$; Scores: $m$; Relevances: $\rho$; Position weights: $\theta$;
    Slot number: $K$; Item length limit: $L$; Sample number: $N$.
$\{y'_{(1)}, y'_{(2)}, \ldots, y'_{(N)}\} \leftarrow$ SampleRankings(m, N, K, L)
Grad $\leftarrow \mathbf{0} \in \mathbb{R}^{|D| \times L}$
$S_1, \ldots, S_L \leftarrow \sum_{d=1}^{|D|} \exp(m(d,1)), \ldots \sum_{d=1}^{|D|} \exp(m(d,L))$
for $j \in [1, 2, \ldots, N]$ do
    // Each $y^x = [(d^x_1, l^x_1), \ldots, (d^x_{|y^x|}, l^x_{|y^x|})], s^x_i = 1 + len(y^x_{1:i-1})$
    $y^{-L+1}, \ldots, y^{L-1} \leftarrow f^{K+L-1}(y'_{(j)}), \ldots, f^{K-L+1}(y'_{(j)})$
    $S'_1, \ldots, S'_L \leftarrow S_1, \ldots, S_L$
    $\mathrm{PR}^{-L+1}_{K+L}, \ldots, \mathrm{PR}^{L-1}_{K+L} \leftarrow 0, \ldots, 0$
    for $i \in [K+L-1, K+L-2, \ldots, 1]$ do
        for $\Delta \in [-L+1, \ldots, L-1]$ do
            $u = \mathbb{1}[|y^\Delta| \geq i]\mathbb{1}[s_i + \Delta \geq 1]\rho(d^\Delta_i, l^\Delta_i)\theta(s^\Delta_i + \Delta, l^\Delta_i)$
            $\mathrm{PR}^\Delta_i \leftarrow \mathrm{PR}^\Delta_{i+1} + u$
    $\mathrm{RI}_0, \mathrm{DR}^1_0, \ldots, \mathrm{DR}^L_0 \leftarrow (0, 0, \ldots, 0)$
    for $i \in [1, \ldots, K]$ do
        $S' = \sum_{l=1}^{L} S'_l$
        $\mathrm{RI}_i \leftarrow \mathrm{RI}_{i-1} + \mathrm{PR}^0_i / S'$
        for $l \in [1, \ldots, L]$ do
            $\mathrm{DR}^l_i \leftarrow \mathrm{DR}^l_{i-1} + \theta(s^0_i, l)/S'$
            $S'_l \leftarrow \mathbb{1}[l \leq K - (s^0_i + l^0_i) + 1](S'_l - \exp(m(d^0_i, l)))$
            $p^l_i \leftarrow \exp(m(d^0_i, l))\mathbb{1}[l \leq K - s^0_i + 1]$
    for $d \in [1, \ldots, |D|]$ do
        for $l \in [1, \ldots, L]$ do
            $r_1 \leftarrow \min(\mathrm{rank}(d, y^0), K)$
            $r_2 \leftarrow \min(r_1, \sup i : \{l \leq K - s^0_i + 1\})$
            $u \leftarrow \exp(m(d,l))[\rho(d,l) \cdot \mathrm{DR}^l_{r_2} - \mathrm{RI}_{r_2}]$
            Grad$(d,l) \leftarrow$ Grad$(d,l) + p^l_{r_1} / \sum_{l'=1}^{L} p^{l'}_{r_1} \cdot \mathrm{PR}^{l-l^0_{r_1}}_{r_1+1} + u$
return Grad$/N$
```

5.5 Using VLPL as in- or post-processing

Existing LTR work mainly aims to optimize machine learning models for the ranking task [3, 26, 28]; we will refer to this as an *in-processing* application of LTR. However, LTR gradients can also be used to optimize a ranking directly, i.e., when m is a look-up table instead of a model. This also allows for a *post-processing* application of LTR, and by extension VLPL, where the reward values ρ are predicted, and subsequently, the ranking is optimized for these predictions. Post-processing is not very popular for standard ranking, as the optimal ranking simply sorts according to ρ, thus there is little added value, while it adds substantial costs during inference. However, in our variable document length setting the PRP does not hold and, therefore, post-processing could actually result in better rankings than is possible with a standard scoring model. For similar reasons, post-processing ranking has also been used for ranking fairness [6]. In this work, we evaluate VLPL both applied as an in-processing method and a post-processing method.

6 Experimental Setup

Our aim is to better understand the variable presentation length setting and evaluate our proposed methods for this task. To begin, as discussed in Section 4.2, there is no known method that can provide the optimal ranking and lengths in this setting, even when all the attractiveness values ρ for the query are available (i.e., the oracle setting). We are therefore interested in determining whether VLPL can find high reward rankings, both when the ρ values are given or unknown. Thus our first two research questions are:

RQ1 When provided with true attractiveness labels, does VLPL learn rankings with a higher EA metric than the baselines?

RQ2 Does VLPL also learn higher EA rankings when it does not have access to the true attractiveness?

Furthermore, as it is unclear what optimized variable length rankings look like, our third research question asks:

RQ3 What document lengths does VLPL place in each position for different choices of θ?

Finally, since VLPL uses samples for its gradient estimation, it seems important to understand its sample-efficiency. Our final research question is therefore:

RQ4 Does VLPL have comparable performance with a lower number of sampled rankings?

Thus, our research questions cover different aspects of VLPL's performance and the variable presentation length setting.

6.1 Datasets and length-specific attractiveness

To our knowledge, there are no datasets featuring different presentation lengths for the same document that can be used to answer our research questions. We thus use *Yahoo! Webscope* [7] and *MSLR-WEB30k* [33] datasets, consisting of queries and associated documents. Each query-document pair is represented by a feature vector and an expert relevance judgement label $R \in \{0, 1, \ldots, 4\}$. Yahoo contains 29,921 queries with an average of 24 documents per query, whilst MSLR consists of 30,000 queries with an average of 125 documents per query. For computational efficiency we focus on the queries with 250 or fewer documents, retaining all queries in Yahoo and over 95% of queries in MSLR.

To generate L attractiveness labels for every document, we first remap the labels in Yahoo to $5L$ non-overlapping equally sized bins. For a document d with original label R, the attractiveness ρ is higher for longer document presentation lengths l, yet is still primarily determined by R: $\rho \in [(RL + l - 1)/(5L), (RL + l)/(5L)]$. On the other hand, due to the higher number of documents per query in MSLR, we emphasize the most relevant documents and double each bin's size relative to the last, renormalizing their sum.

In order to determine $\rho(d, l)$, we then follow Fang et al. [11], generating $L + 1$ scores using query-document features. L of these are normalized with other scores in (d, l)'s bin and map to a quantile within that bin's range, yielding $\rho(d, l)$. Furthermore, as some documents may not actually benefit from a longer presentation [36], we randomly reorder the ρ values of different lengths for half of the documents determined by the $L + 1$'th score.

6.2 Models, baselines and training

For our in-processing models, we use VLPL-1 and VLPL-2 to calculate the gradient of $\frac{\delta}{\delta m}\mathcal{R}(q)$ in order to train a multilayer perceptron (MLP) that learns L scores $m(d, l)$ in Equation 6 for each query-document pair based on its features. In contrast, for the post-processing approach, we use the binary cross-entropy loss to first train an MLP to predict ρ (instead of m). Then for each test query, we instantiate a look-up table with a single score for each (d, l) pair, and use VLPL-1/2 to learn these scores to maximize Equation 3 specifically for that query, using the outputs of MLP in place of ρ.

We contrast our four VLPL models with the following baselines. The first method family, referred to as sort-l, similar to the VLPL post-processing, use an MLP to estimate ρ for a single length l to sort and place documents of only that length. In addition, we train L separate single-length PL-Rank-3 models using in-processing, which we denote as PLR-3-l. For length-adaptive baselines, the greedy model uses MLP estimates of ρ to repeatedly select an eligible (d, l) pair maximizing $\theta(s, l)\rho(d, l)$. However, the above policy may also prefer one longer document over multiple shorter yet only slightly less relevant ones. The slot-avg model thus instead ranks (d, l) pairs on $\theta(s, l)\rho(d, l)/l$, in effect choosing the document with the highest per-slot expected reward. Finally, we also evaluate the above models with full access to the true relevances (oracle setting). In this case the fixed-length sorting models are an upper bound to PL-Rank-3, which is therefore omitted.

All models contain 1-5 hidden layers of size $\{16, 32, \ldots, 128\}$ and sigmoid activations, trained with Adam optimizer [18] with learning rate, $L2$ regularization in $\{0, 1^{-6}, 1^{-5}, \ldots, 1\}$ and dropout in $[0, 0.8]$. We report test set results over 5 independent runs performed under identical circumstances for the best configuration of each model, selected on the validation set results of the first run. VLPL trained with $N = 10{,}000$ sampled rankings per update step. All models trained in PyTorch [29] on a single NVIDIA RTX A5000 GPU.

6.3 Evaluation and position bias

We evaluate all models using the EA objective in Equation 3. We fix the number of slots K at 30 and maximum document length L at 3, representing one possible production setting. Under the standard rank-based position bias model [28], users are assumed to observe the contents of each position with a probability $\theta(i)$ that only depends on the position. As in our setting a document can span multiple positions, we consider the document to be observed as long as any of its slots are observed, yielding $\theta(s, l) = 1 - \prod_{i=s}^{s+l-1} (1 - \theta(i, 1))$, where $\theta(i, 1)$ is equivalent to $\theta(i)$ in Equation 1 – the standard observation probability for rank i [15]. We evaluate two choices of $\theta(i)$ – the slowly decaying DCG weights $\theta_{\text{DCG}}(i) = 1/\log_2(i+1)$ and the steeper inverse rank $\theta_{\text{rank}^{-1}}(i) = 1/i$, corresponding to DCG and the sum of reciprocal ranks if $\forall l = 1$.

7 Results

7.1 Ranking with known relevance

We start by addressing the first research question (**RQ1**): whether VLPL is able to more efficiently order documents and their lengths to achieve higher reward when compared to the baselines, when the attractiveness of each document is known. Table 2 shows the performance in terms of EA in this oracle setting where the number of slots $K = 30$ and maximum document length is $L = 3$ slots.

We can see that adaptively selecting document lengths has a very large impact on the EA metric. Both variable length baselines achieve a substantial and consistent improvement over the single-length baselines, with the greedy model performing better under θ_{DCG} and the slot-avg model with $\theta_{\text{rank}^{-1}}$. Different strategies also achieve the best performance among the single-length models in different settings, highlighting the importance of tailoring the document presentation to the setting where the ranking is shown.

On the other hand, both VLPL-1 and VLPL-2 demonstrate a very large and statistically significant improvement over *all* models across every setting, with a relative performance increase in EA_{DCG} over the best baseline on Yahoo exceeding 10%. As such, VLPL is clearly able to generate highly attractive rankings in this setting. Section 7.4 further compares the performance of VLPL models.

Table 2: EA for $K = 30$ and $L = 3$ in the oracle setting with DCG and inverse rank slot weights. Results are averages over five independent runs; standard deviations in parentheses. Best result in each task denoted in bold. Significant improvement of a model over all (other) baselines denoted by △ (separate one-sided Wilcoxon signed rank tests, $p < 0.05$).

	MSLR		Yahoo!	
Model	$\mathrm{EA}_{\mathrm{DCG}}$	$\mathrm{EA}_{\mathrm{rank}^{-1}}$	$\mathrm{EA}_{\mathrm{DCG}}$	$\mathrm{EA}_{\mathrm{rank}^{-1}}$
sort-1	1.500 (0.002)	0.908 (0.002)	2.832 (0.000)	1.695 (0.000)
sort-2	1.628 (0.005)	0.925 (0.004)	3.113 (0.000)	1.680 (0.000)
sort-3	1.733 (0.010)	0.981 (0.006)	3.137 (0.000)	1.662 (0.000)
greedy	1.917 (0.008)	1.086 (0.005)	$3.326^{\triangle}$(0.000)	1.778 (0.000)
slot-avg	1.911 (0.005)	$1.113^{\triangle}$(0.003)	3.282 (0.000)	$1.926^{\triangle}$(0.000)
VLPL-1	$\mathbf{2.064}^{\triangle}$(0.008)	$\mathbf{1.187}^{\triangle}$(0.005)	$3.665^{\triangle}$(0.000)	$\mathbf{2.031}^{\triangle}$(0.000)
VLPL-2	$\mathbf{2.064}^{\triangle}$(0.008)	$\mathbf{1.187}^{\triangle}$(0.005)	$\mathbf{3.670}^{\triangle}$(0.000)	$2.030^{\triangle}$(0.000)

Overall, we observe that adaptively choosing document presentation length is very important, with particularly high performance shown by VLPL. We thus answer **RQ1** positively: when true attractiveness is known, compared to the baselines, VLPL is able to find significantly more attractive variable presentation length rankings.

7.2 Ranking with learned relevance

We have established that VLPL can effectively optimize variable presentation length rankings when documents' attractivenesses are given. Nevertheless, for previously unseen queries these are unknown. Accordingly, we aim to answer our second research question (**RQ2**): can VLPL be used to generalize to unseen queries?

As shown in Table 3, the performance of all models is substantially lower compared to the oracle setting (cf. Table 2). Here, sort−3 now also consistently performs the best among sort−l models. Together, these observations suggest that relevance misestimation can have a significant impact on learned rankings in this setting.

Nevertheless, both in-processing and post-processing versions of VLPL still achieve statistically significant improvements over all baselines across all settings. We observe particularly high performance for the post-processing approaches, and especially VLPL-2. The high performance of VLPL strongly suggests that being able to tailor document length choices to the specific document pool of each query may be beneficial in the variable length ranking setting.

Interestingly, VLPL-1 appears to generally perform better than VLPL-2 for in-processing. As the performance of both VLPL models, and particularly VLPL-2, is higher for post-processing, we speculate that in-processing may introduce additional noise in the future reward and which may be amplified by reward-sharing of VLPL-2.

Nevertheless, we observe that all VLPL methods perform the best across all our settings. Therefore, we can answer **RQ2** affirmatively: VLPL methods also achieve the highest EA on unseen queries with learned attractiveness.

7.3 Learned rankings

As we've shown that VLPL can learn high reward rankings in the variable presentation length task, we then turn to **RQ3**: what document lengths does VLPL place in the learned rankings?

Table 3: EA for $K = 30$ and $L = 3$ with learned relevance and DCG and inverse rank slot weights. Results are averages over five independent runs, with standard deviations in parentheses. Best result for each task denoted in bold. Best in-processing model in each task is underscored. Significant improvement over all baselines denoted by △ (separate one-sided Wilcoxon signed rank tests, $p < 0.05$).

	MSLR		Yahoo!	
Model	$\mathrm{EA}_{\mathrm{DCG}}$	$\mathrm{EA}_{\mathrm{rank}^{-1}}$	$\mathrm{EA}_{\mathrm{DCG}}$	$\mathrm{EA}_{\mathrm{rank}^{-1}}$
		In-processing		
PLR-3-1	0.777 (0.008)	0.421 (0.003)	2.466 (0.001)	1.385 (0.000)
PLR-3-2	0.839 (0.006)	0.458 (0.003)	2.699 (0.001)	1.394 (0.002)
PLR-3-3	0.931 (0.016)	0.522 (0.008)	2.761 (0.001)	1.423 (0.001)
VLPL-1	$0.953^{\triangle}$(0.019)	$\underline{0.533}^{\triangle}$(0.008)	$\underline{2.982}^{\triangle}$(0.021)	$\underline{1.603}^{\triangle}$(0.001)
VLPL-2	$\underline{0.959}^{\triangle}$(0.018)	$0.527^{\triangle}$(0.005)	$2.976^{\triangle}$(0.002)	$1.590^{\triangle}$(0.003)
		Post-processing		
sort-1	0.781 (0.005)	0.419 (0.002)	2.469 (0.001)	1.385 (0.001)
sort-2	0.843 (0.006)	0.452 (0.003)	2.698 (0.001)	1.404 (0.001)
sort-3	0.929 (0.015)	0.513 (0.010)	2.757 (0.001)	1.424 (0.001)
greedy	0.935 (0.018)	0.492 (0.011)	2.747 (0.001)	1.428 (0.005)
slot-avg	0.896 (0.020)	0.516 (0.007)	2.600 (0.010)	1.520 (0.001)
VLPL-1	$\mathbf{0.971}^{\triangle}$(0.020)	$0.535^{\triangle}$(0.009)	$3.042^{\triangle}$(0.001)	$1.627^{\triangle}$(0.000)
VLPL-2	$\mathbf{0.971}^{\triangle}$(0.020)	$\mathbf{0.536}^{\triangle}$(0.008)	$\mathbf{3.049}^{\triangle}$(0.001)	$\mathbf{1.628}^{\triangle}$(0.001)

The lengths of documents occupying each slot for MSLR are shown in Figure 2. We first examine the behavior of VLPL-2 in the oracle setting, which corresponds to the task of finding the best possible ranking. We see that VLPL-2 starts with a high probability of placing documents at length 1 at the top of the ranking, followed by the increase in the proportion of lengths 2 and 3. Length 3 then occupies the vast majority of the ranking in the final positions. Such behavior of showing lower-ranked documents at longer length was previously found to be beneficial by Marcos et al. [24]. The shift from shorter to longer lengths is especially pronounced for $\theta_{\mathrm{rank}^{-1}}$, suggesting that longer lengths and the associated increase in exposure may be particularly useful under stronger position bias.

Surprisingly, when document attractiveness is not known and instead estimated, VLPL follows a pattern that is roughly similar to the one in the oracle setting, yet with almost no use of length 2. We hypothesize that this may be due to the misestimation of ρ, which, as discussed in Subsection 4.2 may lead to a different solution even if the relative ordering of all estimated $\rho(d, l)$ values is correct.

In contrast to VLPL, the slot-avg model places documents at a short length for longer, whilst the greedy model strongly prefers maximum length for most of the ranking, producing highly similar rankings for different θ. As the performance of VLPL-2 was previously shown to be significantly higher, the difference between models across different settings suggests that balancing document lengths is a critical factor that allows VLPL-2 to achieve strong performance, particularly in the oracle setting.

We saw that VLPL-2 strongly modulates its lengths. As such, for **RQ3** we conclude that: VLPL places shorter documents at the

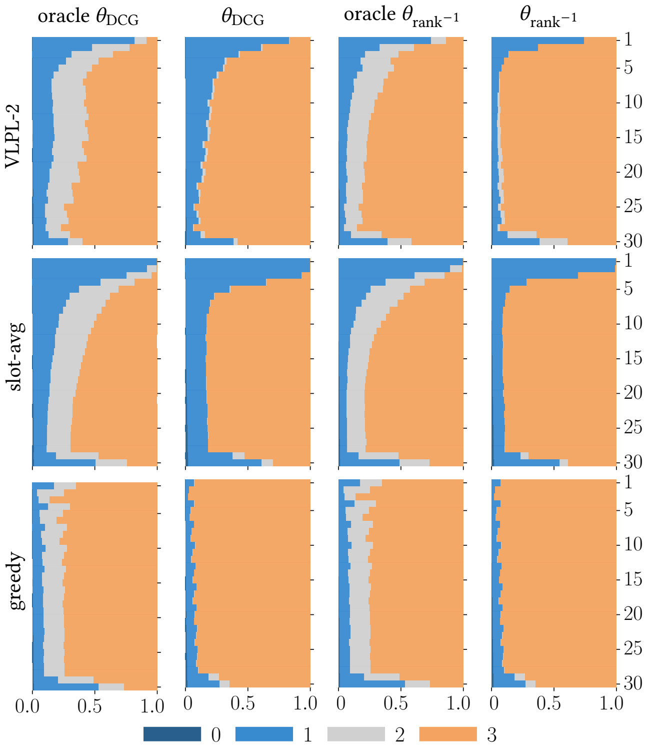

Figure 2: Distribution of document lengths occupying each slot in learned rankings in MSLR. y-axis of stacked bar chart denotes the slot, x-axis denotes the proportion of rankings in the first test fold where the slot is occupied by the corresponding length. Length 0 denotes padding.

start of the ranking and progressively increases the placed document lengths, with a quicker transition for $\theta_{\text{rank}^{-1}}$.

7.4 Sample-efficiency

We then arrive at our final research question (**RQ4**): can VLPL models also achieve strong performance whilst using fewer samples? To answer this, Figure 3 shows VLPL performance under a varied number of samples N in the oracle and non-oracle settings.

We can see that increasing the number of samples leads to improvements in EA, with smaller, but still observable gains at larger sample numbers. We also observe that whilst the performance of VLPL-1 and VLPL-2 is very similar at $N = 10{,}000$, VLPL-2 shows stronger performance at lower values of N, with particularly substantial improvements over VLPL-1 at $N \leq 100$. Nevertheless, for $N \geq 100$, both VLPL models generally achieve strong performance and always outperform the best baseline at $N = 1{,}000$, highlighting the high applicability of VLPL to the variable presentation length ranking task. As such, in response to **RQ4**, we affirm that: VLPL models, and especially VLPL-2, achieve large improvements over the baselines even when using fewer samples.

8 Conclusion and Future Work

In this paper we introduced the variable presentation length ranking setting, where the task is to simultaneously choose the document order and the size of each document's presentation in the ranking. We showed that this setting is substantially more complex than standard ranking – the best performance can only be achieved when document order and presentation lengths are decided jointly. To tackle this problem, we introduced the variable document length Plackett-Luce model, and four VLPL methods for optimizing it. Our experiments show that VLPL is highly suited for ranking with variable presentation lengths and underscore the importance of considering document presentation in LTR.

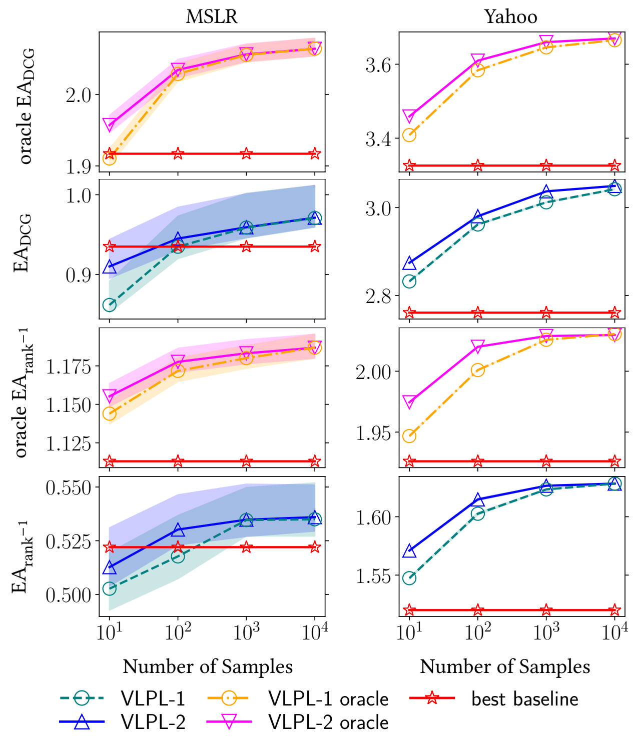

Figure 3: EA for post-processing VLPL models under different number of samples. Results are averages over five independent runs, the difference between the best and worst runs denoted by shading. Best baseline per setting denoted by star.

Importantly, whilst we model document presentation lengths in terms of slots, it may also be possible to extend it to continuous values to represent pixel lengths of each document. Furthermore, as different policies may be best suited for different presentation contexts, being able to learn a policy that is guaranteed to be effective across multiple settings (e.g., on mobile and desktop) may further help the application of such models. Altogether, our work highlights the importance, the difficulty and the opportunities of LTR with variable result presentation lengths.

Reproducibility Our experimental implementation is publicly available at https://github.com/NKNY/varlenranksigir2025.

Acknowledgments

This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-9790. This work is partially supported by the Dutch Research Council (NWO), grant number VI.Veni.222.269.

References

[1] Luca Aiello, Ioannis Arapakis, Ricardo Baeza-Yates, Xiao Bai, Nicola Barbieri, Amin Mantrach, and Fabrizio Silvestri. 2016. The Role of Relevance in Sponsored Search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 185–194.

[2] Eugene M Bland, Gregory S Black, and Kay Lawrimore. 2005. Determinants of Effectiveness and Success for EBay Auctions. *The Coastal Business Journal* 4, 1 (2005).

[3] Alexander Buchholz, Jan Malte Lichtenberg, Giuseppe Di Benedetto, Yannik Stein, Vito Bellini, and Matteo Ruffini. 2022. Low-variance estimation in the Plackett-Luce model via quasi-Monte Carlo sampling. *ArXiv* abs/2205.06024 (2022).

[4] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. *Learning* 11, 23-581 (2010), 81.

[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning*. ACM, 129–136.

[6] Simon Caton and Christian Haas. 2024. Fairness in Machine Learning: A Survey. *Comput. Surveys* 56, 7 (2024), 1–38.

[7] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research* 14 (2011), 1–24.

[8] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

[9] 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.

[10] Edward Cutrell and Zhiwei Guan. 2007. What Are You Looking for?: An Eye-Tracking Study of Information Usage in Web Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 407–416.

[11] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-Dependent Examination-Bias Estimation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[12] Norbert Fuhr. 1989. Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle. *ACM Trans. Inf. Syst.* 7, 3 (1989), 183–204.

[13] Olivier Jeunen, Ivan Potapov, and Aleksei Ustimenko. 2024. On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top- *n* Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1222–1233.

[14] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 133–142.

[15] Thorsten Joachims. 2003. Evaluating Retrieval Performance Using Clickthrough Data. In *Text Mining*.

[16] Diane Kelly and Leif Azzopardi. 2015. How Many Results per Page?: A Study of SERP Size, Search Behavior and User Experience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 183–192.

[17] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. 2016. Pagination versus Scrolling in Mobile Web Search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 751–760.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization.

[19] Ron Kohavi. 2015. Online Controlled Experiments: Lessons from Running A/B/n Tests for 12 Years. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1–1.

[20] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[21] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. 2008. Eye Tracking and Online Search: Lessons Learned and Challenges Ahead. *Journal of the American Society for Information Science and Technology* 59, 7 (2008), 1041–1052.

[22] R. Duncan Luce. 1977. The Choice Axiom after Twenty Years. *Journal of Mathematical Psychology* 15, 3 (1977), 215–233.

[23] Haitao Mao, Lixin Zou, Yujia Zheng, Jiliang Tang, Xiaokai Chu, Jiashu Zhao, Qian Wang, and Dawei Yin. 2024. Whole Page Unbiased Learning to Rank. In *Proceedings of the ACM Web Conference 2024*. ACM, 1431–1440.

[24] Mari-Carmen Marcos, Ferran Gavin, and Ioannis Arapakis. 2015. Effect of Snippets on User Experience in Web Search. In *Proceedings of the XVI International Conference on Human Computer Interaction*. ACM, 1–8.

[25] David Maxwell, Leif Azzopardi, and Yashar Moshfeghi. 2017. A Study of Snippet Length and Informativeness: Behaviour, Performance and User Experience. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 135–144.

[26] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[27] Harrie Oosterhuis. 2022. Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation with Minimal Computational Complexity. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2266–2271.

[28] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 489–498.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, Vol. 32.

[30] Kanaad Pathak, Leif Azzopardi, and Martin Halvey. 2024. The Influence of Presentation and Performance on User Satisfaction. *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval*.

[31] Kanaad Pathak, Leif Azzopardi, and Martin Halvey. 2024. Ranking Heterogeneous Search Result Pages using the Interactive Probability Ranking Principle. In *European Conference on Information Retrieval*.

[32] Robin L. Plackett. 1975. The Analysis of Permutations. *Journal of The Royal Statistical Society Series C-applied Statistics* 24 (1975), 193–202.

[33] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *ArXiv* abs/1306.2597 (2013).

[34] S.E. Robertson. 1977. The Probability Ranking Principle in IR. *Journal of Documentation* 33, 4 (1977), 294–304.

[35] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations As Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. 1670–1679.

[36] Simon Sweeney and Fabio Crestani. 2006. Effective Search Results Summary Size and Device Screen Size: Is There a Relationship? *Information Processing & Management* 42, 4 (2006), 1056–1074.

[37] Aleksei Ustimenko and Liudmila Prokhorenkova. 2020. StochasticRank: Global Optimization of Scale-Free Discrete Functions. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 9669–9679.

[38] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1313–1322.

[39] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8, 3 (1992), 229–256.

[40] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 1192–1199.

[41] Sheng You, Chao Wang, Baohua Wu, Jingping Liu, Quan Lu, Guanzhou Han, and Yanghua Xiao. 2023. What Image Do You Need? A Two-stage Framework for Image Selection in E-commerce. In *Companion Proceedings of the ACM Web Conference 2023*. ACM, 452–456.

[42] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 1011–1018.