



# On Adaptive Knowledge Distillation with Generalized KL-Divergence Loss for Ranking Model Refinement

Yingrui Yang

Department of Computer Science  
University of California  
Santa Barbara, California, USA  
yingruiyang@cs.ucsb.edu

Shanxiu He

Department of Computer Science  
University of California  
Santa Barbara, California, USA  
shanxiuhe@cs.ucsb.edu

Tao Yang

Department of Computer Science  
University of California  
Santa Barbara, California, USA  
tyang@cs.ucsb.edu

## ABSTRACT

Knowledge distillation is useful in training a neural document ranking model by employing a teacher to guide model refinement. As a teacher may not perform well in all cases, over-calibration between the student and teacher models can make training less effective. This paper studies a generalized KL divergence loss in a weighted form for refining ranking models in searching text documents, and examines its formal properties in balancing knowledge distillation in adaption to the relative performance of the teacher and student models. This loss differentiates the role of positive and negative documents for a training query, and allows a student model to take a conservative or deviate approach in imitating the teacher's behavior when the teacher model is worse than the student model. This paper presents a detailed theoretical analysis with experiments on the behavior and usefulness of this generalized loss.

## CCS CONCEPTS

• Information systems → Learning to rank.

## KEYWORDS

Neural document ranking, knowledge distillation, KL divergence

### ACM Reference Format:

Yingrui Yang, Shanxiu He, and Tao Yang. 2024. On Adaptive Knowledge Distillation with Generalized KL-Divergence Loss for Ranking Model Refinement. In *Proceedings of the 2024 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '24), July 13, 2024, Washington DC, DC, USA*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3664190.3672522>

## 1 INTRODUCTION

Large-scale search systems for text documents typically employ multi-stage ranking in practice. The first retrieval stage extracts top candidate documents matching a query from a large search index with a fast and relatively efficient ranking method. The second stage or a later stage uses a more complex machine learning algorithm to re-rank top results thoroughly. Recent sparse retriever studies exploit learned neural representations DeepImpact [29], uniCOIL [13, 24] and SPLADE [8, 10]. An alternative method is dense retrieval which uses a dual encoder architecture



This work is licensed under a Creative Commons Attribution International 4.0 License.

ICTIR '24, July 13, 2024, Washington, DC, USA  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0681-3/24/07.  
<https://doi.org/10.1145/3664190.3672522>

with single-vector [34, 49], multi-vector document representations (e.g. [18, 35]).

To boost the relevance of these models, knowledge distillation [15] is critical during training to transfer knowledge from a powerful teacher model through behavior imitation [12, 16, 25]. KL divergence is a popular training loss for knowledge distillation in document ranking [25, 34, 35, 43, 44].

One drawback of KL-divergence loss for document ranking is that it does not exploit characteristics of contrastive learning in ranking model refinement because it does not differentiate positive and negative documents for a training query. As a result, it over-calibrates between the student and teacher models with a tight distribution matching in every document without prioritization even when the teacher performs worse than the student model. The previous work has used the weighted sum of a contrastive loss such as log-likelihood with KL divergence as a regularization to reduce over-fitting, and a recent BKL study [46] improves this by regularizing KL divergence with an entropy and L1-norm loss. As discussed in Section 4, this BKL regularization can behave incorrectly in three significant case regions when a teacher is better than and is worse than a student model for a training query.

To address the aforementioned weakness, the contribution of this paper is a generalized KL-divergence loss formula called weighted KL divergence (WKL) with a detailed analysis of its theoretical properties. Instead of following the regularization approach, this generalized loss guides knowledge distillation adaptively in ranking model refinement by differentiating the role of positive and negative documents and prioritizing the alignment of a student model and a teacher model for effectively separating positive and negative documents. This paper provides a loss lower bound analysis and a relative gradient contribution study to characterize the behavior of WKL during model training, compared to KL divergence. Our analysis shows that this generalized loss can dynamically assess the relative performance of the teacher and student model in each training query, and adaptively adjust the imitating behavior of the student model, so that the teacher model is followed when it performs better than the student model, and is conservatively followed or not followed at all when this teacher performs worse than the student model.

Our evaluation with MS MARCO passage and BEIR datasets shows that WKL works well with three student models including SPLADE sparse retrieval, ColBERT ranking with a multi-vector representation [35], and a single-vector SimLM dense retriever [43]. WKL can outperform a few other loss options for refinement after starting from the same warmup checkpoint in the evaluated models.

## 2 BACKGROUND AND RELATED WORK

**Problem definition.** We follow the notation used in [46]. Given query  $Q$ , document search on a collection of  $N$  text documents (i.e.,  $\mathcal{D} = \{d_i\}_{i=1}^N$ ) finds top  $k$  results with a ranking mainly based on their query-document similarity. For training a retriever or re-ranker, contrastive learning is widely used. Let  $\mathcal{D}^+$  be the subset of all positive documents, and  $\mathcal{D}^-$  be a subset containing all negative documents for query  $Q$ . We assume that in a training dataset, all positive documents are ranked equally. That is true for the MS MARCO passage dataset where there are only binary labels.

The top one probability distribution over these documents is:

$$P(d_i|Q, \mathcal{D}^+, \mathcal{D}^-, \Theta) = \frac{\exp(S(Q, d_i, \Theta))}{\sum_{j=1}^N \exp(S(Q, d_j, \Theta))}$$

where  $\Theta$  is the vector of neural parameters involved.  $S(Q, d_i, \Theta)$  is a scoring function that captures the semantic similarity of a document with a query. For the simplicity of presentation when no confusion is caused, we will not list  $\Theta$  and  $Q$  explicitly in each symbol below and the loss function is specified for each query  $Q$  based on parameters  $\Theta$  under the training documents  $\mathcal{D}^+$  and  $\mathcal{D}^-$ . Knowledge distillation is a training methodology that guides the refinement of a neural student model using a teacher model. Let  $p_i$  or  $q_i$  denote  $P(d_i|Q, \mathcal{D}^+, \mathcal{D}^-, \Theta)$  where  $p_i$  and  $q_i$  refer to the teacher's and student's predictions, respectively.

To train a ranking model, the standard loss function includes the negative log-likelihood or its variation:  $-\sum_{d_j \in \mathcal{D}^+} \log q_j$ . KL-divergence defined below is a popular choice for knowledge distillation as seen in recent ranking studies [34, 35, 39, 43, 50].

$$L_{KL} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} p_i \ln \frac{p_i}{q_i} \quad (1)$$

where  $p_i$  and  $q_i$  refer to the teacher and student's top one probability for instance  $d_i$  in  $\mathcal{D}^+$  or  $\mathcal{D}^-$ , respectively. KL-divergence measures the distance between teacher's and student's distributions. It is known that the lower bound of KL-divergence loss is 0 and this is achieved when  $\forall d_i, p_i = q_i$ .

**Related retrieval methods.** Large-scale search systems for text documents typically employ multi-stage ranking in practice. The first stage retriever aims to fetch top  $k$  documents using a fast and relatively simple method. There are two categories of retrieval techniques in deriving a document and query representation. One category of document retrieval is lexical sparse retrieval models, such as BM25, which take advantage of fast inverted index implementations on CPUs. This method gains its popularity recently due to the advancement of learned sparse representations that derive token weights from a BERT-based neural model [7, 10, 13, 24, 29, 38].

Dense retrieval is an alternative approach for first-stage search with a dual encoder architecture (e.g. [11, 45]). Distillation is shown to be effective for dense retrieval training and KL-divergence loss is a popular choice in recent studies, such as RocketQAv2 [34], SimLM [43] and RetroMAE [44], AR2 [50], and UnifiedR [39].

**Re-ranking and multi-vector representations.** The second or later stage of search can employ a more complex re-ranker to re-evaluate the top  $k$  documents fetched by an earlier stage. There is a possibility to use a single-vector dense retrieval model for re-ranking. As pointed out in recent studies [23, 36, 41], single-vector dense models can struggle in handling out-of-domain datasets

where training data is limited (including zero-shot retrieval), and in answering entity-centric questions. As a remedy, multi-vector representations including ColBERT and its new enhancements [22, 23, 28, 32] have been proposed to improve the model expressiveness by capturing fine-grained token-level information.

**Listwise losses.** A listwise loss design that considers the impact of relative rank positions of matched documents for a query has been shown to be useful in learning-to-rank and aligning such a loss with a targeted ranking metric approximately such as NDCG is ideal [27, 42]. Since neural information retrieval typically requires a large number of training examples to be effective, and training data such as MS MARCO only contains few labeled positive documents and sampled negative documents on a relatively large scale, it is more important to separate positive and negative documents properly for a query-specific loss. This motivates our design. The previous work has considered the relevance gain by swapping two documents in a listwise loss, e.g. LambdaMART [1]. CL-DRD [48] uses a listwise loss based on rank position. Weighting training instances is studied in the focal loss for visual object classification [26], and such a loss is not designed for knowledge distillation. Nevertheless, our work is influenced by the above studies.

**Regularization of knowledge distillation with a contrastive loss.** A key weakness of knowledge distillation with KL divergence loss for document ranking is that a teacher model may not perform well in all cases and adaptive deprioritization is needed. A common approach to balance knowledge distillation is to combine the KL divergence loss with a contrastive rank loss such as the log-likelihood using a weighted sum as a regularization, defined as:

$$L_{KLL} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} p_i \ln \frac{p_i}{q_i} - \lambda \sum_{d_i \in \mathcal{D}^+} \log q_i \quad (2)$$

where  $\ln$  means  $\log_e$  and  $\log$  means  $\log_2$ . The above loss is not adaptive to the relative performance of a teacher model and a student model. An improvement called BKL [46] combines the negative entropy component of positive documents and the L1-norm expression of negative documents for a given query to balance knowledge distillation.

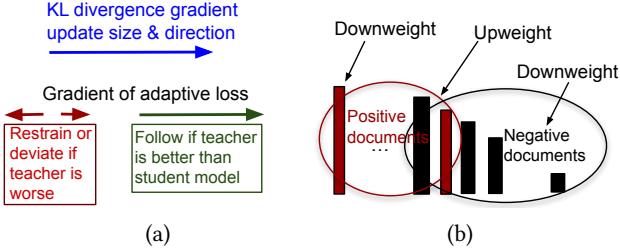
## 3 LOSS DESIGN AND ANALYSIS

### 3.1 Design considerations

Our goal of loss design optimization is to control the imitation of the teacher's rank scoring when refining a student model based on each training query so that when the student mimics when the teacher is better and it should restrain distillation or deviate when the teacher is worse. This can be analyzed by examining the gradient contribution of each document for parameter update during SGD-based training compared to KL divergence. As illustrated in Figure 1(a), a desired loss should follow the gradient update direction of KL divergence loss when the teacher model performs better than the student model for a training query. When this teacher performs worse, this targeted loss should deviate in an opposite update direction or at least restrain the update size cautiously even in the same update direction.

The weakness of BKL [46] is that its formula over-corrects the behavior of KL divergence and fails to meet the above objective in three significant case regions. As shown in Section 4, when a

teacher's model performs much better than a student in ranking a negative example, BKL's regularization formula unintentionally lets the student model deviate from the teacher's ranking score in a wrong learning direction. It also fails in some cases to follow aggressively even when the teacher model performs worse for a positive document.



**Figure 1: (a) Loss design goal: Adaptive control of student model learning from teacher compared to the KL divergence loss. (b) Prioritize separation of positives and negatives in student ranking.**

To meet the above expectation and goal illustrated in Figure 1(a), and avoid the misbehavior in both BKL and KLL, our approach described below does not take the regularization approach of BKL and KLL. Instead we directly generalize the KL divergence loss with an easy-to-implement weighting formula described in Section 3.2. In generalizing the KL divergence loss, our main strategy is to explicitly prioritize the separation of positive and negative documents for each query in student ranking through a weight adjustment by down-weighting positive documents ranked high on the top positions, and negative documents ranked low at the bottom positions by a student model. This is illustrated in Figure 1(b) where Y axis is the student rank score of documents and X axis is the rank position of documents from top to low based on their student rank score. Positives are marked in red and negatives are marked in black.

Through a gradient contribution analysis, Section 4 analytically reveals that the generalized KL divergence loss described below can accomplish the goal illustrated in Figure 1(a) by adopting the weighting strategy illustrated in Figure 1(b).

### 3.2 Generalized KL-Divergence

This generalized KL-divergence loss in a weighted format (WKL) is defined as follows:

$$L_{WKL} = \sum_{d_j \in \mathcal{D}^+} (1 - q_j)^{\gamma_1} p_j \ln \frac{p_j}{q_j} + \sum_{d_i \in \mathcal{D}^-} (q_i)^{\gamma_2, i} p_i \ln \frac{p_i}{q_i}.$$

The weight for each divergence term  $p_i \log \frac{p_i}{q_i}$  corresponds to the importance to align the student's scoring of such a document with the teacher's model. For a positive document  $d_j$ , the goal is to have  $q_j$  as large as possible towards 1, and thus we use  $(1 - q_j)^{\gamma_1}$  as the weight. Here  $\gamma_1$  is a fixed hyperparameter controlling the scale of weight in the exponent. We require  $\gamma_1 \geq 0$ . For a negative document  $d_i$ , the goal is to have  $q_i$  as small as possible towards 0, and thus we use  $(q_i)^{\gamma_2, i}$  as the weight. We require either  $\gamma_2, i > 0$  for all negative documents or  $\gamma_2, i = 0$  for all negative documents.

For two negative documents  $d_i$  and  $d_j$  where  $q_i \geq q_j$ , we require  $\gamma_{2,i} \leq \gamma_{2,j}$ .

Notice that KL divergence loss is a special form of WKL when setting all control parameters as zero ( $\gamma_1 = \gamma_{2,i} = 0$ ). WKL weights the divergence loss contribution from positive documents and negative documents differently. We explain how the above design matches the design consideration illustrated in Figure 1(b).

- Given two positive documents  $d_i$  and  $d_j$ , if  $q_i \geq q_j$ , then  $(1 - q_i)^{\gamma_1} \leq (1 - q_j)^{\gamma_1}$ . Thus a low-scoring positive document is weighted more than a high-scoring positive document. When such a document is ranked close to negative documents, or even below some negative documents, that results in a poor boundary separation of positive and negative documents. Thus the alignment with the teacher's model for such a positive document should be prioritized.
- Among negative documents, if  $q_i \leq q_j$ , requiring  $\gamma_{2,i} \leq \gamma_{2,j}$  implies that  $(q_i)^{\gamma_{2,i}} \geq (q_j)^{\gamma_{2,j}} \geq (q_i)^{\gamma_{2,j}}$ . High-scoring negative documents are weighted more and low-scoring negative documents have a reduced priority to follow what the teacher does. When the score of a negative example in a student model is high and is getting closer or exceeds some of the positive examples, the positive and negative document regions would overlap as shown in Figure 1(b) and then that is a high-priority case to address.

### 3.3 Loss minimization and its bound

The result below shows that the WKL loss has a constant lower bound, and thus training that minimizes such a loss has a boundary to hit. If a loss function has no lower bound, training would not converge. Note that  $p_i$  values from the teacher's model are constant.

**Theorem 1.** *Loss minimization. When  $\gamma_1 \geq 1$  or  $\gamma_1 = 0$ ,*

$$L_{WKL} \geq \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i. \quad (3)$$

When  $0 < \gamma_1 < 1$ ,

$$L_{WKL} \geq r_1 \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j + (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i. \quad (4)$$

**PROOF.** When  $\gamma_1 \geq 1$ , we follow Bernoulli's inequality  $(1 - q_i)^{\gamma_1} \geq 1 - r_1 q_i$ , given  $0 \leq q_j \leq 1$ . Since  $\ln p_i \leq 0$  and  $\ln q_i \leq 0$ ,

$$\begin{aligned} L_{WKL} &= \sum_{d_j \in \mathcal{D}^+} (1 - q_j)^{\gamma_1} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln p_i \\ &\quad - \sum_{d_j \in \mathcal{D}^+} (1 - q_j)^{\gamma_1} p_j \ln q_j - \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln q_i \\ &\geq \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i \\ &\quad - \sum_{d_j \in \mathcal{D}^+} (1 - q_j \gamma_1) p_j \ln q_j - \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln q_i \end{aligned}$$

$$\begin{aligned} &\geq \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i \\ &\quad + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i. \end{aligned}$$

When  $\gamma_1 = 0$ ,

$$\begin{aligned} L_{WKL} &\geq \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln q_i \\ &\quad + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i. \end{aligned}$$

When  $0 < \gamma_1 < 1$ , we first show that  $(1-x)^{\gamma_1} \geq \gamma_1(1-x)$  with  $x \in [0, 1]$ . It is true when  $x = 0$  and  $x = 1$ . For  $x \in (0, 1)$ , let function  $f(x) = (1-x)^{\gamma_1} + \gamma_1 x$ . Then  $f'(x) = -r_1(1-x)^{\gamma_1-1} + \gamma_1 < 0$ . Then  $f(x)$  is monotonically decreasing and  $f(x) > f(1)$  for  $x \in (0, 1)$ , which leads to  $(1-x)^{\gamma_1} > \gamma_1(1-x)$ . We apply this inequality for  $x = q_j$  for a positive document below.

$$\begin{aligned} L_{WKL} &= \sum_{d_j \in \mathcal{D}^+} (1-q_j)^{\gamma_1} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln p_i \\ &\quad - \sum_{d_j \in \mathcal{D}^+} (1-q_j)^{\gamma_1} p_j \ln q_j - \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln q_i \\ &\geq \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i \\ &\quad - \sum_{d_j \in \mathcal{D}^+} \gamma_1(1-q_j)p_j \ln q_j - \sum_{d_i \in \mathcal{D}^-} q_i^{\gamma_{2,i}} p_i \ln q_i \\ &\geq r_1 \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j \\ &\quad + (1-\gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i. \end{aligned}$$

□

The first component of the right hand side in Inequalities (3) and (4) is KL divergence for positive documents. The sum of the first and second components on the right-hand side approaches a constant lower bound, reached when  $p_i = q_i$  for all positive documents and  $q_i = 0$  for all negative documents. The third component is the negative entropy of positive documents. The third component is bounded by  $-\frac{2\gamma_1}{e}$ , approached when all  $q_j$  values are equal for all positive documents  $d_j$ . This is shown in the theorem below.

**Theorem 2.** *Constant-bounded loss. If  $\gamma_{2,i} > 0$ , and when  $\gamma_1 \geq 1$  or  $\gamma_1 = 0$ ,*

$$L_{WKL} \geq \sum_{d_i \in \mathcal{D}^-} p_i(-1 + \ln p_i) - \frac{2\gamma_1}{e}. \quad (5)$$

*If  $\gamma_{2,i} > 0$ , and when  $0 < \gamma_1 < 1$ ,*

$$L_{WKL} \geq \sum_{d_i \in \mathcal{D}^-} p_i(-\gamma_1 + \ln p_i) - \frac{2\gamma_1}{e}. \quad (6)$$

*If  $\gamma_{2,i} = 0$ , and when  $\gamma_1 \geq 1$  or  $\gamma_1 = 0$ ,*

$$L_{WKL} \geq -\frac{2\gamma_1}{e}. \quad (7)$$

*If  $\gamma_{2,i} = 0$ , and when  $0 < \gamma_1 < 1$ ,*

$$L_{WKL} \geq (1-\gamma_1) \sum_{d_j \in \mathcal{D}^+} (-p_j + p_j \ln p_j) - \frac{2\gamma_1}{e}. \quad (8)$$

Notice that  $p_i$  and  $p_j$  from the teacher's model in the above bound expressions are constants. The proof for Theorem 2 is based on Theorem 1 and is listed in Appendix A.

Based on the components of the derived lower bound in Theorem 1, minimizing WKL will minimize the original KL-divergence loss for positive documents and maximize the entropy among them. This lower bound minimization implies a balanced trend towards a narrower gap between teacher's and student's predictions of positive documents and relatively equal student predictions among them while preferring low scores for negative documents.

## 4 RELATIVE GRADIENT CONTRIBUTIONS

We analyze the impact of up-weighting and down-weighting individual KL-divergence terms in terms of their corresponding gradient contributions for parameter update during model refinement because gradients of the loss controls the update size to the network weight parameters in the SGD-based training process. Let  $\theta$  be one of parameters  $\Theta$  used in the computation network that maps the input features to score  $S(Q, d_i, \Theta)$  for each document  $d_i$  defined in Section 2. Then given Loss  $L_A$ , and  $A$  can be  $WKL$ ,  $BKL$ , or others.

$$\frac{\partial L_A}{\partial \theta} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} \frac{\partial L_A(i)}{\partial q_i} \frac{\partial q_i}{\partial S(Q, d_i, \Theta)} \frac{\partial S(Q, d_i, \Theta)}{\partial \theta}$$

where  $L_A(i)$  is the relevant loss term contributed by document  $d_i$ .

For KL divergence loss  $L_{KL}$  in Equation (1),  $L_{KL}(i) = p_i \ln \frac{p_i}{q_i}$ .

$$\frac{\partial L_{KL}(i)}{\partial q_i} = -\frac{p_i}{q_i}.$$

To understand if a loss function  $L_A$  follows the KL divergence loss when a teacher model performs better than a student or not, we compare the pairwise ratio of the gradient contribution from document  $d_i$  in above additive formulas for  $\frac{\partial L_A(i)}{\partial q_i}$  compared to  $\frac{\partial L_{KL}(i)}{\partial q_i}$ . Namely

$$\frac{\partial L_A(i)}{\partial q_i} = g_A \frac{\partial L_{KL}(i)}{\partial q_i} \quad (9)$$

The top portion of Table 1 gives the expected behavior of a knowledge distillation loss compared to KL divergence loss when a teacher model performs better or worse than a student. The bottom portion of Table 1 explains the meaning of different ranges of  $g_A$  value on the gradient contribution of document  $d_i$ . Here the relative performance assessment of a teacher model and a student model for a document is defined below based on the relative ratio of teacher prediction and student prediction.

- A teach model performs better than a student model when  $p_i > q_i$  if  $d_i$  is a positive document when  $p_i < q_i$  if  $d_i$  is a negative document
- A teach model performs worse than a student model when  $p_i < q_i$  if  $d_i$  is a positive document, and when  $p_i > q_i$  if  $d_i$  is a negative document.

For  $L_{WKL}$ , the contribution  $L_{WKL}(i)$  from document  $d_i$  is  $(1-q_i)^{\gamma_1} p_i \ln \frac{p_i}{q_i}$  for a positive document, and  $q_i^{\gamma_{2,i}} p_i \ln \frac{p_i}{q_i}$  for a negative document. It is easy to verify that

$$\frac{\partial L_{WKL}(i)}{\partial q_i} = g_{WKL} \frac{\partial L_{KL}(i)}{\partial q_i} \quad (10)$$

Scenarios	Expected behavior
Teacher is better than student	$g_A \geq 1$ preferred. At least $g_A > 0$
Teacher is worse	$g_A \leq 0$ preferred. At most $g_A < 1$
Condition	Behavior interpretation on $d_i$ contribution by $L_A$
$g_A > 1$	Aggressively follow KL divergence
$g_A = 1$	Exactly follow KL divergence
$0 < g_A < 1$	Conservatively follow
$g_A = 0$	Not follow. No contribution from $d_i$ in $L_A$ .
$g_A < 0$	Not follow. Deviate from $d_i$ from KL Divergence

**Table 1: Expected gradient contribution behavior from document  $d_i$  in loss  $L_A$  compared to KL divergence**

where

$$g_{WKL} = \begin{cases} (1 - q_i)^{\gamma_1 - 1} \times \left( \gamma_1 q_i \ln \frac{p_i}{q_i} + 1 - q_i \right) & \text{if } d_i \in \mathcal{D}^+; \\ q_i^{\gamma_2, i} \times \left( 1 - \gamma_2, i \ln \frac{p_i}{q_i} \right) & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

For KL divergence regularized together with the log-likelihood (Equation (2)),

$$g_{KLL} = \begin{cases} 1 + \frac{\lambda}{p_i} & \text{if } d_i \in \mathcal{D}^+; \\ 1 & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

Thus KLL always follows KL divergence loss even a teacher performs worse than a student. BKL in [46] improves this by combining the KL divergence with a log likelihood rank loss linearly using a small  $\lambda$  parameter value.

$$g_{BKL} = \begin{cases} 1 - \frac{\lambda}{p_i} q_i \log(e \times q_i) & \text{if } d_i \in \mathcal{D}^+; \\ 1 - \frac{q_i \lambda}{p_i \ln 2} & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

Constant  $e$  is the base of the natural logarithms.

Table 2 gives a comparison of the behavior of WKL and BKL for the gradient contribution of an individual document  $d_i$  compared to KL divergence compared to KL divergence based on different  $g_{WKL}$  and  $g_{BKL}$  value ranges and the relative ratio of teacher's and student's predictions  $\frac{p_i}{q_i}$ . Notice that when  $p_i > q_i$ , we consider a teacher model *performs better* than a student if  $d_i \in \mathcal{D}^+$ , and *performs worse* if  $d_i \in \mathcal{D}^-$ . When  $p_i < q_i$ , we consider this teacher model *performs better* than a student if  $d_i \in \mathcal{D}^-$ , and *performs worse* if  $d_i \in \mathcal{D}^+$ .

Table 2 lists the conditions representing three significant misbehavior regions, to be illustrated in Figure 2(a), in which BKL fails to meet the expectation discussed in the top portion of Table 1. WKL is well-behaved as shown from this table and its behavior is formally characterized by the following theorem.

**Theorem 3.** *When a teacher model performs better than a student model in ranking a document for a query,  $g_{WKL} > 0$ . When this teacher model performs worse,  $g_{WKL} < 1$ , and  $g_{WKL} \leq 0$  when  $q_i \geq \max(e \times p_i, \frac{1}{\gamma_1 + 1})$  for  $d_i \in \mathcal{D}^+$  and when  $\frac{p_i}{q_i} \geq e^{\frac{1}{\gamma_2, i}}$  for  $d_i \in \mathcal{D}^-$ .*

**PROOF.** For  $d_i \in \mathcal{D}^+$ , we consider the ratio  $g_{WKL}$  defined in Equation (10) in two cases.

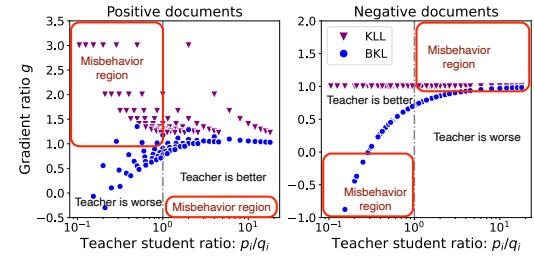
- When  $p_i > q_i$ ,  $\gamma_1 q_i \ln \frac{p_i}{q_i} + 1 - q_i > 0$ . Thus  $g_{WKL} > 0$ .
- When  $p_i < q_i$ ,  $\gamma_1 q_i \ln \frac{p_i}{q_i} + 1 - q_i < 1 - q_i$ . Thus  $g_{WKL} < (1 - q_i)^{\gamma_1} \leq 1$ .

When  $q_i \geq e \times p_i$ ,  $\gamma_1 q_i \ln \frac{p_i}{q_i} + 1 - q_i \leq -\gamma_1 q_i + 1 - q_i \leq 0$  if  $q_i \geq \frac{1}{\gamma_1 + 1}$ .

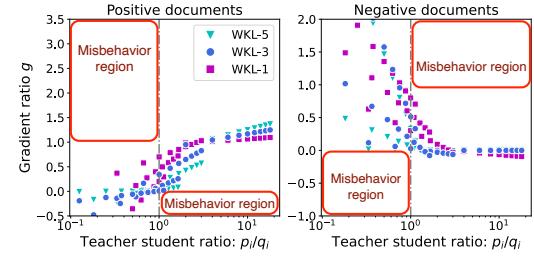
Thus  $g_{WKL} \leq 0$  when  $q_i \geq \max(e \times p_i, \frac{1}{\gamma_1 + 1})$ . For  $d_i \in \mathcal{D}^-$ ,

- When  $p_i > q_i$ ,  $1 - \gamma_2, i \ln \frac{p_i}{q_i} < 1$ . Then  $g_{WKL} < q_i^{\gamma_2, i} \leq 1$ . When  $\frac{p_i}{q_i} \geq e^{\frac{1}{\gamma_2, i}}$ ,  $1 - \gamma_2, i \ln \frac{p_i}{q_i} \leq 0$ . Then  $g_{WKL} \leq 0$ .
- When  $p_i < q_i$ ,  $1 - \gamma_2, i \ln \frac{p_i}{q_i} > 0$ . Then  $g_{WKL} > 0$ .

□



(a) Relative gradient contribution ratio  $g$  of BKL and KLL



(b) Gradient contribution ratio of WKL with  $\gamma_1 = \gamma_2, i = 5, 3$ , or  $1$

**Figure 2: WKL vs. BKL & KLL when teacher is better or worse**

To illustrate the comparison in Table 2 using an example, Figure 2(a) plots the gradient contribution ratio  $g_{BKL}$  in a blue dot and  $g_{KLL}$  in a purple triangle with  $\lambda = 0.1$ . The x-axis is  $\frac{p_i}{q_i}$  varying from 0.01 to 10 at Figure 2(b) plots  $g_{WKL}$  with  $\gamma_1 = \gamma_2, i = 5, 3$ , or 1 marked as WKL-5 (a light blue triangle), WKL-3 (a dark blue dot), and WKL-1 (a purple square), respectively.

The rectangle red boxes marked the misbehavior regions show that the areas where the gradient contribution ratio values do not match the expected behavior described in the top portion of Table 1. From Figure 2(b),  $g_{WKL}$  values are outside the red misbehavior regions, and thus WKL follows KL divergence loss if the teacher model does better than the student, but it restrains gradient update with a conservative size, or deviates in an opposite direction when the teacher is worse.

From Figure 2(a), there are three significant misbehavior regions in red into which BKL gradient ratios fall, meaning BKL fails to meet the expectations as summarized in Table 2. There are two misbehavior regions into which KLL falls.

For example, Figure 2(b) illustrates that for positive documents, when the teacher performs better with  $\frac{p_i}{q_i} > 1$ ,  $g_{WKL} > 0$  or exceeds 1 and WKL allows the student to follow the teacher's parameter update direction. When the teacher under-performs with  $\frac{p_i}{q_i} < 1$ ,

Scenarios	Behavior of WKL	Behavior of BKL
<b>Positive document</b> $d_i$ . $p_i$ is teacher prediction, $q_i$ is student prediction.		
Teacher: better $p_i > q_i$	$g_{WKL} > 0$ Conservatively or aggressively follow	$g_{BKL} > 0$ Conservatively or aggressively follow
Student: better $p_i < q_i$	$g_{WKL} < 1$ to conservatively follow or deviate $g_{WKL} \leq 0$ when $\frac{q_i}{p_i} \geq e$ , $q_i \geq \frac{1}{r_1+1}$ to deviate	$g_{BKL}$ varies from negative to positive <b>Misbehavior:</b> $g_{BKL} > 1$ aggressively follows when $q_i > e^{-1}$
<b>Negative document</b> $d_i$		
Student: better $p_i > q_i$	$g_{WKL} < 1$ to conservatively follow or deviate $g_{WKL} \leq 0$ when $\frac{p_i}{q_i} \geq e^{\frac{1}{r_2,i}}$ to deviate	<b>Misbehavior:</b> When $p_i > q_i$ , $g_{BKL} \approx 1$ Otherwise $0 < g_{BKL} < 1$ to conservatively follow
Teacher: better $p_i < q_i$	$g_{WKL} > 0$ Conservatively/aggressively follow	$0 < g_{BKL} < 1$ if $\frac{p_i}{q_i} > \frac{\lambda}{\ln 2}$ to conservatively follow <b>Misbehavior:</b> $g_{BKL} \leq 0$ if $\frac{p_i}{q_i} \leq \frac{\lambda}{\ln 2}$ to deviate

Table 2: A behavior comparison of relative gradient contributions by document  $d_i$  in  $L_{WKL}$  and  $L_{BKL}$ 

$g_{WKL}$  become close to 0 or even negative, and the student does not learn much from the teacher or its learning deviates from the teacher's learning direction. In comparison from the left portion of Figure 2(b), BKL still forces the student to follow the teacher's direction with  $g_{BKL} > 0$  or even  $> 1$  in most cases when the teacher is worse. Thus WKL's design corrects the misbehavior of BKL.

## 5 EVALUATION RESULTS

### 5.1 Evaluation setup for student models

We apply WKL in refining three student models during training.

- The SPLADE model [8, 10] which computes the weight score  $w_j$  of  $j$ -th token term for a sparse vector of document  $d$  as

$$w_j = \sum_{i \in d} \log(1 + \text{ReLU}(H(h_i)^T E_j + b_j))$$

where document  $d$  consists of a sequence of BERT last layer's embeddings  $(h_1, h_2, \dots, h_n)$ .  $E_j$  is the BERT input embedding of the  $j$ -th token and  $b_j$  is a token level bias.  $H(\cdot)$  is a linear layer with activation and layer normalization.

- Two-stage search pipeline that combines the results of first-stage SPLADE retrieval and the second-stage ColBERT top- $k$  ranking with a fusion [5, 21]. ColBERT's scoring formula is:

$$\sum_{h_i \in M(Q, \Theta)} \max_{h_j \in M(d, \Theta)} H(h_i)^T H(h_j)$$

where each document  $d$  and given query  $Q$  use a multi-vector representation  $M(d, \Theta)$  and  $M(Q, \Theta)$  respectively, and  $h_i, h_j$  are BERT last layer's embeddings and  $H(\cdot)$  is one linear layer with normalization on the output representation.

- Dense single-vector retriever SimLM [43]. It is a state-of-the-art dual-encoder with optimized pretraining [43, 44].

**WKL parameters.** We have considered the following special version of WKL, called CKL, and a preliminary evaluation with two student models can be found in [47]. This section provides more detailed and additional evaluation results with one extra student model. For negative document  $d_i$ , we set  $\gamma_{2,i} = \gamma_1 - \beta_i$ . The exponent weight bias  $\beta_i$  is defined as

$$\beta_i = \alpha \left( \frac{1}{\pi(i)} - \frac{1}{|\mathcal{D}^+|} \sum_{d_j \in \mathcal{D}^+} \frac{1}{\pi(j)} \right). \quad (11)$$

Here  $\pi(i), \pi(j)$  are the rank of negative document  $d_i$  and positive document  $d_j$  respectively. Bias  $\beta_i$  represents the importance of correcting the ranking position of negative document  $d_i$ , compared against the harmonic average position of positive documents. The above use of a rank position is motivated by the previous work which considers the relevance gain by swapping two documents in a ranked order, e.g. LambdaMART [1] and CL-DRD [48]. The above expression satisfies  $|\beta_i| < \alpha$ . Among negative documents, if  $q_i > q_j$ , document  $d_i$  is ranked before  $d_j$ .  $\frac{1}{\pi(i)} > \frac{1}{\pi(j)}$ . Thus  $\beta_i > \beta_j$ . Then  $\gamma_{2,i} < \gamma_{2,j}$ . That meets the requirement specified in Section 3.2.

Exponent bias  $\beta_i$  is updated based on its rank position immediately after each training iteration where  $q_i$  is recomputed, which makes the loss function non-differentiable. Thus during training, we opt to periodically update  $\beta_i$  using the latest student's model performance, and the priority adjustment of each negative document is stable for a block of training iterations. This design allows  $\beta_i$  to be treated as a constant in the loss function. This is a reasonable tradeoff as model refinement that addresses ranking accuracy for a negative document takes a number of iterations and continuous  $\beta_i$  adjustment for such a document may not yield sufficient benefits.

Since  $\gamma_1$  and  $\alpha$  determine the value of  $\gamma_{2,i}$  for every document  $d_i$ , the rest of this section will use two hyperparameters  $\gamma_1$  and  $\alpha$  to adjust the configuration of WKL, and investigate the sensitivity with different choices of  $\gamma_1$  and  $\alpha$  values in model refinement.

**Datasets and metrics.** We use the MS MARCO datasets for full passage ranking [2, 6]. MS MARCO contains 8.8 million passages and 502,940 training queries with binary judgment labels for each query. The development (Dev) query set contains 6980 test queries while the test sets in TREC deep learning (DL) 2019 and 2020 tracks provide 43 and 54 queries, respectively. Following the common practice, we report mean reciprocal rank (MRR@10) for the Dev set and NDCG@10 score for TREC DL test sets. The recall ratio at 1000 is another metric which is the percentage of relevant-labeled results appeared in the final top-1000 results. We also use BEIR which contains 13 publicly available datasets [41] for evaluating the zero-shot performance of the trained models.

Our evaluation implementation uses C++ and Python. The implementation of SPLADE model follows its official release [40] and sparse retrieval code in PISA [30] with some optimization [19, 33]. We follow the SBERT library [37] to implement ColBERT. Two teachers are used during training. For SimLM, we use the code

and checkpoint released in the SimLM project GitHub. The cross encoder teacher adopted for ColBERTv2 and SPLADE is MiniLM-l-6-v2 [31] with 0.407 MRR@10 on MS MARCO Dev on top of SPLADE retrieval. For SimLM, we use a cross encoder teacher from the released SimLM project [43] with 0.438 MRR@10. More information on training and configurations can be found in Appendix B.

## 5.2 Model refinement and parameter choices

$\gamma_1, \alpha$	Dev MRR@10	DL19 NDCG@10	DL20 NDCG@10
<b>SPLADE + top-1000 ColBERT re-ranking</b>			
2,0, 0,0	0.404	0.716	0.709
2,0, 0.5	0.404	0.733	0.725
2,0, 1.0	0.404	0.740	0.728
3,0, 0,0	0.405	0.735	0.717
4,0, 0,0	0.408	0.740	0.734
4,0, 1.0	0.410	0.735	0.731
4,0, 1.5	0.404	0.724	0.740
5,0, 0,0	0.409	0.737	0.722
<b>5,0, 1,0</b>	<b>0.411</b>	<b>0.744</b>	<b>0.741</b>
5,0, 1.5	0.407	0.742	0.731
6,0, 0,0	0.410	0.750	0.724
<b>SPLADE retrieval</b>			
2,0, 0,0	0.3993	0.7435	0.7177
3,0, 0,0	0.4008	0.7348	0.7255
4,0, 0,0	0.4006	0.7309	0.7215
5,0, 0,0	0.4007	0.7456	0.7256
<b>5,0, 1,0</b>	<b>0.4013</b>	<b>0.7445</b>	<b>0.7206</b>
5,0, 1.5	0.4008	0.7192	0.7335
6,0, 0,0	0.4003	0.7317	0.7180
<b>SimLM retrieval</b>			
<b>1,0, 0</b>	<b>0.395</b>	0.708	0.706
2,0, 0	0.394	—	—
3,0, 0	0.393	—	—
4,0, 0	0.392	—	—

**Table 3: Student performance when varying WKL parameters**

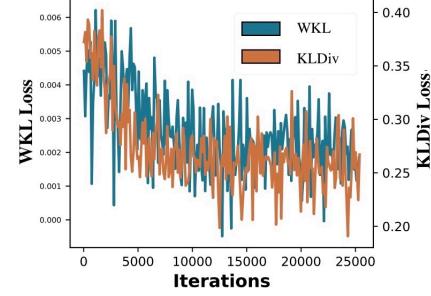
Table 3 shows the relevance performance of the three student models refined by WKL for MS MARCO passage Dev set, TREC DL19, and DL20 under different hyperparameter  $\gamma_1$  and  $\alpha$  values. The result shows that the refinement by WKL boosts the performance of each student model with a well-balanced relevance effectiveness across the tested datasets.

For the two-stage SPLADE/ColBERT pipeline, the model starting point after warmup and before WKL refinement is 0.399 MRR for the Dev set, and WKL boosts to 0.411 MRR. When  $\gamma_1$  is too small, WKL behaves similarly as KL-divergence and when  $\gamma_1$  becomes too big, the gradient will reduce quickly towards 0. Such a value is not preferred. Thus setting with  $\gamma_1 = 5$  and  $\alpha = 1.0$  is a good choice.

For the SPLADE student model, the starting point after warmup allows the SPLADE retriever to reach 0.394 MRR@10. WKL refinement further boosts to 0.401 MRR. The middle portion of Table 3 lists the performance of SPLADE refined with WKL under different hyperparameters  $\gamma_1$  and  $\alpha$  values. As one can see,  $\gamma_1 = 5$  and  $\alpha = 1$  perform decently well and these are our default choice.

For the dense retrieval student model, the released SimLM checkpoint [3] gives 0.344 MRR@10 using the standard MS MARCO. Applying KL divergence further boosts to 0.365 MRR, and after this warmup, WKL delivers 0.395 MRR@10. Without this warmup, WKL delivers 0.381. The bottom portion of Table 3 lists the performance

of SimLM refined with WKL after KL divergence warmup under different hyperparameter  $\gamma_1$  and  $\alpha$  values. The result shows that setting with  $\gamma_1 = 1$  and  $\alpha = 0$  is a good choice for SimLM.



**Figure 3: Training loss during training**

Figure 3 shows that like KL divergence, the WKL loss gradually decreases over the time as training iterates and converges for refining the two-stage pipeline with MS MARCO passages. This matches Theorem 2 in Section 3.3 which shows that WKL loss has a constant lower bound to hit.

## 5.3 A comparison with related work

	Dev	DL19	DL20	BEIR(Avg)	
				MRR@10	NDCG@10
SPLADE++ [9]	0.380	0.732	—	—	0.507
SimLM with title anno. [43]	0.411	0.712	0.697	—	—
SimLM/KL w/o title anno.	0.365	0.685	0.611	—	—
ColBERTv2	0.397	—	—	—	0.499
SLIM++ [22]	0.404	0.714	0.697	—	0.490
CITADEL [23]	0.399	0.703	0.702	—	0.501
ALIGNER [32]	0.403	—	—	—	0.511
SPLADE+ColBERT/BKL [46]	0.407	0.716	0.736	—	0.505
<b>SPLADE+ColBERT/WKL</b>	<b>0.411</b>	<b>0.744</b>	<b>0.741</b>	<b>0.515</b>	—
SPLADE/WKL	0.401	0.745	0.7206	—	—
SimLM/WKL w/o title	0.395	0.708	0.706	—	—

**Table 4: A comparison with baselines**

Table 4 compares the student models in the last three rows with the related baselines in terms of MRR@10 or NDCG@10. The result demonstrates that the refined student models are competitive to the state-of-the-art research. This table lists the published performance of SPLADE++ for sparse retrieval. For multi-vector representations, it lists dense retrievers with multi-vector representations using ColBERTv2, CITADEL, and ALIGNER. It also lists SLIM++ [22] which improves multi-vector representations with a sparse scheme. For two-stage search that combines SPLADE and ColBERT, this table lists the model performance with refinement by BKL [46] and WKL is better than BKL on the tested datasets.

Notice that our student model for dense retriever SimLM yields 0.395 MRR, below 0.411 reported in [3] which evaluates the modified MS MARCO dataset with title annotation. Title annotation is considered unfair in [20] since the original dataset released doesn't utilize title information. All experiments for WKL follow the standard approach of using the original MS MARCO without title annotation,

The column for BEIR in Table 4 lists the average NDCG@10 across 13 datasets. Table 5 gives details using the WKL-refined two-stage pipeline. This refined model performs well compared to the zero-shot performance of BM25 retrieval, sparse SPLADE++,

dense SimLM, and ColBERTv2. BM25/MiniLM uses BM25 retrieval and then re-ranks with the MiniLM cross-encoder.

Dataset	BM25	SPLADE++	SimLM	ColBERTv2	BM25/miniLM	WKL
Search Tasks						
DBpedia	0.313	0.436	0.351	0.446	0.400	<b>0.459</b>
FiQA	0.236	0.349	0.298	0.356	0.309	0.372
NQ	0.329	0.533	0.502	<b>0.562</b>	0.453	<b>0.562</b>
HotpotQA	0.603	0.693	0.568	0.667	0.677	0.692
NFCorpus	0.325	0.345	0.318	0.338	<b>0.364</b>	0.348
T-COVID	0.656	0.725	0.515	0.738	<b>0.766</b>	0.746
Touche-2020	<b>0.367</b>	0.242	0.292	0.263	0.314	0.316
Semantic Relatedness Tasks						
ArguAna	0.315	0.518	0.376	0.463	0.473	<b>0.578</b>
C-FEVER	0.213	0.237	0.171	0.176	<b>0.239</b>	0.231
FEVER	0.753	0.796	0.689	0.780	0.756	0.779
Quora	0.789	0.849	0.797	<b>0.852</b>	0.843	0.746
SCIDOCs	0.158	0.161	0.137	0.154	0.170	0.164
SciFact	0.665	0.710	0.559	0.568	0.697	0.698
Average	0.440	0.507	0.429	0.499	0.497	<b>0.515</b>
BM25 Diff	–	15.24%	-2.60%	13.47%	12.94%	<b>16.92%</b>

Table 5: Zero-shot performance (average NDCG@10) on BEIR

Table 6 compares a few other loss options when refining the SPLADE model. The loss options include MarginMSE loss [16] and KLL defined in Eq. 2. “CL-DRD” is a listwise loss in CL-DRD for curriculum learning [48]. Training for all loss options is conducted under the same training setup in terms of negative samples, the starting warm-up checkpoint, and the machine environment. Recall@1000 for these losses is near identical as 0.983 for the Dev set, and thus it is not listed. This table marks the results with ‘†’ if a baseline result is in statistically significant degradation from WKL. After a warmup, WKL outperforms the tested other loss options with a smaller advantage in the Dev set while having a larger improvement of DL’19 and DL’20 sets. For the SPLADE and ColBERT pipeline, WKL has a larger MRR gain for the Dev set compared to these loss options [47].

	Dev	DL19	DL20
Loss option	MRR@10	NDCG@10	NDCG@10
KLDiv	0.399 <sup>†</sup>	0.656	0.689
MarginMSE	0.397 <sup>†</sup>	0.664	0.678
KLDiv_logL	0.396 <sup>†</sup>	0.669	0.672
CL-DRD	0.400	0.674	0.662
<b>WKL</b>	<b>0.4013</b>	<b>0.7445</b>	<b>0.7206</b>

Table 6: SPLADE refinement under different losses

## 6 CONCLUDING REMARKS

The contribution of this work is to provide a detailed analysis of a generalized KL divergence loss (WKL) in an easy-to-implement weighted format. Our lower bound analysis gives an insight into the behavior characteristic of WKL during model refinement. The relative gradient contribution study reveals that WKL follows the gradient update direction of KL divergence loss when the teacher model performs better than the student model for a training query. When this teacher performs worse, WKL deviates in an opposite update direction or restrains the update size cautiously in the same update direction.

The evaluation shows that WKL can boost the relevance of three student models for the tested datasets. WKL can outperform a few

other loss options after a warmup. This warmup is necessary in the tested cases and thus WKL is useful for model refinement after initial training with another loss.

Our future work is to investigate the use of WKL in more ranking models and experiments. The limitation of this work is that the applicability of WKL is restricted to ranking applications where binary positive and negative labels are assigned per training queries. This considers that it is hard and costly to build a dataset at a large scale for ranker training with multi-level labels in practice. It is interesting to extend WKL in the future for training data with multi-level labels.

**Acknowledgments.** We thank Yifan Qiao for his help and the anonymous referees for their valuable comments. This work is supported in part by NSF IIS-2225942 and has used the computing resource of the ACCESS program supported by NSF. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

## A PROOF OF THEOREM 2

**PROOF.** let  $RHS(i)$  be the  $i$ -th component of in the right-hand side of Inequality (3) or Inequality (4) in Theorem 1.

We further four cases in order to derive a lower constant bound. **Case 1)** We consider the cases of  $\gamma_{2,i} > 0$ .

- We apply a known inequality:  $\ln x \leq x - 1$  when  $x$  is positive and the equality is reached when  $x = 1$ .

$$\sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} \geq \sum_{d_j \in \mathcal{D}^+} p_j (1 - \frac{q_j}{p_j}) \geq \sum_{d_j \in \mathcal{D}^+} p_j - \sum_{d_j \in \mathcal{D}^+} q_j.$$

The lower bound is accomplished when  $p_j = q_j$  for all positive documents. Since

$$\sum_{d_j \in \mathcal{D}^+} p_j + \sum_{d_i \in \mathcal{D}^-} p_i = 1 \text{ and } \sum_{d_j \in \mathcal{D}^+} q_j + \sum_{d_i \in \mathcal{D}^-} q_i = 1,$$

$$\sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} \geq - \sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} q_i.$$

- Based on the above derivation, when  $\gamma_1 \geq 1$  or  $\gamma_1 = 0$ ,

$$\begin{aligned} RHS(1) + RHS(2) &\geq - \sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} q_i - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i, \\ &\geq - \sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} p_i (q_i - q_i^{\gamma_{2,i}} \ln q_i) \\ &\geq - \sum_{d_i \in \mathcal{D}^-} p_i. \end{aligned}$$

When  $0 < \gamma_1 < 1$ ,

$$\begin{aligned} RHS(1) + RHS(2) &= \gamma_1 (- \sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} q_i) - \sum_{d_i \in \mathcal{D}^-} p_i q_i^{\gamma_{2,i}} \ln q_i, \\ &\geq -\gamma_1 \sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} p_i (\gamma_1 q_i - q_i^{\gamma_{2,i}} \ln q_i) \\ &\geq -\gamma_1 \sum_{d_i \in \mathcal{D}^-} p_i. \end{aligned}$$

Notice that in the above derivation, Expression  $q_i - q_i^{\gamma_{2,i}} \ln q_i$  has its lower bound achieved when  $q_i$  is approaching 0. When  $0 < \gamma_1 < 1$  Expression  $\gamma_1 q_i - q_i^{\gamma_{2,i}} \ln q_i$  also has its lower bound achieved when  $q_i$  is approaching 0.

- Now we derive a lower bound for  $RHS(3) = \frac{\gamma}{\log e} \sum_{d_i \in \mathcal{D}^+} q_i \log q_i$ . Since function  $x \log x$  is convex and following Jensen's inequality on a convex function,

$$\frac{\sum_{d_j \in \mathcal{D}^+} q_j \log q_j}{s} \geq \left( \frac{\sum_{d_j \in \mathcal{D}^+} q_j}{s} \right) \log \left( \frac{\sum_{d_j \in \mathcal{D}^+} q_j}{s} \right)$$

where  $s = |\mathcal{D}^+|$ . Let  $z = \sum_{d_i \in \mathcal{D}^-} q_i$ . Then

$$\sum_{d_j \in \mathcal{D}^+} q_j \log q_j \geq (1 - z) \log \left( \frac{1 - z}{s} \right) \geq -\frac{2 \log e}{e}.$$

Expression  $(1 - z) \log \left( \frac{1 - z}{s} \right)$  is bounded by  $-\frac{2 \log e}{e}$  by computing its minimum value.

Adding the above component lower bounds together. When  $\gamma_1 \geq 1$  or  $\gamma_1 = 0$ ,  $L_{WKL} = RHS(1) + RHS(2) + RHS(3) + RHS(4) \geq \sum_{d_i \in \mathcal{D}^-} p_i (-1 + \ln p_i) - \frac{2\gamma_1}{e}$ .

When  $0 < \gamma_1 < 1$ ,

$$L_{WKL} = RHS(1) + RHS(2) + RHS(3) + RHS(4) + RHS(5)$$

$$\geq (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i (-\gamma_1 + \ln p_i) - \frac{2\gamma_1}{e}.$$

**Case 2)** We consider the case of  $\gamma_{2,i} = 0$  and  $\gamma_1 = 0$ . In this case, WKL is the same as KL divergence loss.  $L_{WKL} = L_{KL} \geq 0$ . The lower bound is accomplished  $p_i = q_i$  for all positive and negative documents  $d_i$ .

**Case 3)** Now we consider the cases of  $\gamma_{2,i} = 0$ , and  $\gamma_1 > 0$ . There are two subcases. **Subcase 3.1)** When  $\gamma_1 > 1$ , from Inequality (3),

$$\begin{aligned} L_{WKL} &\geq \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i \ln q_i \\ &\quad + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i \\ &\geq \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j \\ &\geq -\frac{2\gamma_1}{e}. \end{aligned}$$

**Subcase 3.2)** When  $0 < \gamma_1 < 1$ , from Inequality (4),

$$\begin{aligned} L_{WKL} &\geq r_1 \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} - \sum_{d_i \in \mathcal{D}^-} p_i \ln q_i + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j \\ &\quad + (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j + \sum_{d_i \in \mathcal{D}^-} p_i \ln p_i \\ &\geq r_1 \sum_{d_j \in \mathcal{D}^+} p_j \ln \frac{p_j}{q_j} + \sum_{d_i \in \mathcal{D}^-} p_i \ln \frac{p_i}{q_i} \\ &\quad + \frac{\gamma_1}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j + (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j \\ &\geq \gamma_1 \left( -\sum_{d_i \in \mathcal{D}^-} p_i + \sum_{d_i \in \mathcal{D}^-} q_i \right) + \left( \sum_{d_i \in \mathcal{D}^-} p_i - \sum_{d_i \in \mathcal{D}^-} q_i \right) \\ &\quad - \frac{2\gamma_1}{e} + (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j \\ &\geq (1 - \gamma_1) \sum_{d_i \in \mathcal{D}^-} p_i - (1 - \gamma_1) - \frac{2\gamma_1}{e} + (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} p_j \ln p_j \\ &= (1 - \gamma_1) \sum_{d_j \in \mathcal{D}^+} (-p_j + p_j \ln p_j) - \frac{2\gamma_1}{e}. \end{aligned}$$

□

## B TRAINING STEPS AND CONFIGURATIONS

Training for each student model involves two steps: Step 1 is to warm up the student model with knowledge distillation following a fixed teacher model. Step 2 is to use the proposed WKL loss or other loss options to refine the student retriever model and the student re-ranker model separately. When we compare different loss functions for the refinement, we always start from the same model after warm-up and refine it using the same set of training triplets and the same teacher model. In this way, we rule out the potential influence caused by different implementation details in performance comparison.

The cross encoder teacher adopted for ColBERTv2 and SPLADE is MiniLM-l-6-v2 [31]. For SimLM, we use a cross encoder from the released SimLM project [43]. Following the setting of SPLADE++, we use co-Condenser [4] as the pretrained starting checkpoint and adopt sentenceBERT [14] as the ranker to select hard negatives. This warm-up step chooses margin-MSE [16] as a loss for knowledge distillation for both retriever and re-ranker. To train the retrieval model, we also add additional sparsity regularization with coefficients 0.008 and 0.01 for a query and documents respectively, following SPLADE++. This observation aligns with the results reported in TAS-B [17]. The above warm-up step allows the SPLADE retriever to reach 0.394 MRR@10 and the ColBERT re-ranker to deliver 0.399 MRR@10.

In Step 2 for model refinement, we use the WKL loss for knowledge distillation or another loss function to compare. We index the corpus with a warm-up retrieval model using PISA [30]. To speedup training, we only retrieve the top 100 documents (passages for MS MARCO) per query for re-ranking during training. Negative sampling uses the top 20 documents per training query after re-ranking as candidate hard negatives. During model refinement, we sample negative examples from these 20 documents so that the total number of positive and negative documents is a fixed constant, limited by the available GPU memory. For our machine environment, this fixed constant is 6. Namely, if there are 2 positive documents for a query, we sample at most 4 negative documents.

In terms of training machine resources and parameters, we use four NVIDIA V100 GPUs to warm up and refine SPLADE with the training batch size as 128 queries and to warm up and refine ColBERT with a batch size of 32 queries. This training resource usage is reasonable compared to what has been used in the previous work [17, 34, 35]. Learning rates 2e-5 and 1e-5 are used in the warm-up step and the refinement step, respectively. We update the exponent weight bias  $\beta_i$  discussed above every 2000 training batches, as more frequent update does not lead to an improvement. When training the student retriever, to avoid the expensive re-indexing time during this update, we re-evaluate the top 50 documents per training query as an approximation using the model checkpoint saved after every 2000 batches. The above refinement with WKL for training takes less than 20 epochs to converge.

The selected default WKL parameters are  $(\gamma, \alpha) = (5, 1)$  for ColBERT and SPLADE, and  $(\gamma, \alpha) = (1, 0)$  for SimLM. Section 5 examines the choices and sensitivities of these WKL parameters for the tested models.

## REFERENCES

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

[2] Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. *ArXiv* abs/1611.09268 (2016).

[3] SimLM Checkpoint. [n. d.]. <https://huggingface.co/intfloat/simlm-base-msmarco-finetuned>. ([n. d.]).

[4] Co-Condenser. 2021. <https://huggingface.co/Luyu/co-condenser-marco>. (2021).

[5] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *SIGIR* (2009).

[6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2020 Deep Learning Track. *ArXiv* abs/2102.07662 (2020).

[7] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Term Weighting For First Stage Passage Retrieval. *SIGIR* (2020).

[8] Thibault Formal, C. Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *ArXiv* abs/2109.10086 (2021).

[9] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. *SIGIR* (2022).

[10] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. *SIGIR* (2021).

[11] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACM, Online and Punta Cana, Dominican Republic, 981–993. <https://doi.org/10.18653/v1/2021.emnlp-main.75>

[12] Luyu Gao, Zhuyun Dai, and J. Callan. 2020. Understanding BERT Rankers Under Distillation. *Proceedings of SIGIR* (2020).

[13] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. *NAACL* (2021).

[14] MS-MARCO Hard-Negatives. 2022. <https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives>. (2022).

[15] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv* abs/1503.02531 (2015).

[16] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. *ArXiv* abs/2010.02666 (2020).

[17] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy J. Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. *SIGIR* (2021).

[18] O. Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *SIGIR* (2020).

[19] Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. *SIGIR* (2022).

[20] Carlos Lassance and Stephane Clinchant. 2023. The Tale of Two MSMARCO - and Their Unfair Comparisons. In *Proceed. of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, USA, 2431–2435.

[21] Hang Li, Shuai Wang, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and G. Zuccon. 2022. To Interpolate or not to Interpolate: PRF, Dense and Sparse Retrievers. *SIGIR* (2022).

[22] Minghan Li, Sheng-Chieh Lin, Xueguang Ma, and Jimmy Lin. 2023. SLIM: Sparsified Late Interaction for Multi-Vector Retrieval with Inverted Indexes. In *SIGIR*. arxiv and sigir 2023:2302.06587 [cs.IR]

[23] Minghan Li, Sheng-Chieh Lin, Barlas Onguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen tau Yih, and Xilun Chen. 2022. CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval. *arXiv* 2211.01267 (2022). arXiv:2211.10411 [cs.IR]

[24] Jimmy J. Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *ArXiv* abs/2106.14807 (2021).

[25] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy J. Lin. 2021. In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval. In *REPL4NLP*.

[26] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollar. 2017. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2017), 318–327.

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

[28] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and M. Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *TACL* (2021).

[29] Antonio Mallia, O. Khattab, Nicola Tonello, and Torsten Suel. 2021. Learning Passage Impacts for Inverted Indexes. *SIGIR* (2021).

[30] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 2019. PISA: Performant indexes and search for academia. *Proceedings of the Open-Source IR Replicability Challenge* (2019).

[31] MiniLM-L-6-v2. 2022. <https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2>. (2022).

[32] Yujie Qian, Jinyu Lee, Sai Meher Karthik Duddu, Zhuyun Dai, Siddhartha Brahma, Iftekhar Naim, Tao Lei, and Vincent Y. Zhao. 2022. Multi-Vector Retrieval as Sparse Alignment. *arXiv* 2211.01267 (2022). arXiv:2211.01267 [cs.CL]

[33] Yifan Qiao, Yingrui Yang, Shanxiu He, and Tao Yang. 2023. Representation Sparsification with Hybrid Thresholding for Fast SPLADE-based Document Retrieval. *ACM SIGIR'23* (2023).

[34] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACM, Online and Punta Cana, Dominican Republic, 2825–2835.

[35] Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. *NAACL'22* ArXiv abs/2112.01488 (2022).

[36] Christopher Sciavolino, Zexuan Zhong, Jinyu Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACL, 6138–6148.

[37] Sentence-Transformers. 2022. <https://www.sbert.net/>. (2022).

[38] Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Dixin Jiang. 2023. LexMAE: Lexicon-Bottlenecked Pretraining for Large-Scale Retrieval. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=PfpEtB3-csK>

[39] Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Kai Zhang, and Dixin Jiang. 2022. UnifieR: A Unified Retriever for Large-Scale Retrieval. *ArXiv* abs/2205.11194 (2022).

[40] SPLADE. 2022. <https://github.com/naver/splade>. (2022).

[41] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *NeurIPS*.

[42] Thibaut Thonet, Yagmur Gizem Cinar, Eric Gaussier, Minghan Li, and Jean-Michel Renders. 2022. Listwise Learning to Rank Based on Approximate Rank Indicators. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (Jun. 2022), 8494–8502.

[43] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Dixin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with Representation Bottleneck for Dense Passage Retrieval. *ACL* (2023).

[44] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training Retrieval-oriented Transformers via Masked Auto-Encoder. *EMNLP* (2022).

[45] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[46] Yingrui Yang, Shanxiu He, Yifan Qiao, Wentai Xie, and Tao Yang. 2023. Balanced Knowledge Distillation with Contrastive Learning for Document Re-Ranking. In *Proc. of 2023 ACM SIGIR Inter. Conf. on Theory of Information Retrieval* (Taipei, Taiwan) (ICTIR '23), 247–255.

[47] Yingrui Yang, Yifan Qiao, Shanxiu He, and Tao Yang. 2024. Weighted KL-Divergence for Document Ranking Model Refinement. *ArXiv* and *SIGIR* 2024. arXiv:2406.05977

[48] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum Learning for Dense Retrieval Distillation. *SIGIR* (2022).

[49] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. *CoRR* abs/2104.08051 (2021). <https://arxiv.org/abs/2104.08051>

[50] Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial Retriever-Ranker for dense text retrieval. *ICLR* (2022).