Information Retrieval Journal (2023) 26:1
https://doi.org/10.1007/s10791-023-09419-0

An in-depth study on adversarial learning-to-rank

Hai-Tao Yu[1] · Rajesh Piryani[1] · Adam Jatowt[2] · Ryo Inagaki[1,4] · Hideo Joho[1] · Kyoung-Sook Kim[3]

Received: 17 November 2021 / Accepted: 4 October 2022 / Published online: 28 February 2023

Abstract
In light of recent advances in adversarial learning, there has been strong and continuing interest in exploring how to perform adversarial learning-to-rank. The previous adversarial ranking methods [e.g., IRGAN by Wang et al. (IRGAN: a minimax game for unifying generative and discriminative information retrieval models. Proceedings of the 40th SIGIR pp. 515–524, 2017)] mainly follow the generative adversarial networks (GAN) framework (Goodfellow et al. in Generative adversarial nets. Proceedings of NeurIPS pp. 2672–2680, 2014), and focus on either pointwise or pairwise optimization based on the rule-based adversarial sampling. Unfortunately, there are still many open problems. For example, how to perform listwise adversarial learning-to-rank has not been explored. Furthermore, GAN has many variants, such as f-GAN (Nowozin et al. in Proceedings of the 30th international conference on neural information processing systems, pp. 271–279, 2016) and EBGAN (Zhao et al. in Energy-based generative adversarial network. International conference on learning representations (ICLR), 2017), a natural question arises then: to what extent does the adversarial learning strategy affect the ranking performance? To cope with these problems, firstly, we show how to perform adversarial learning-to-rank in a listwise manner by following the GAN framework. Secondly, we investigate the effects of using a different adversarial learning framework, namely f-GAN. Specifically, a new general adversarial learning-to-rank framework via variational divergence minimization is proposed (referred to as IRf-GAN). Furthermore, we show how to perform pointwise, pairwise and listwise adversarial learning-to-rank within the same framework of IRf-GAN. In order to clearly understand the pros and cons of adversarial learning-to-rank, we conduct a series of experiments using multiple benchmark collections. The experimental results demonstrate that: (1) Thanks to the flexibility of being able to use different divergence functions, IRf-GAN-pair shows significantly better performance than adversarial learning-to-rank methods based on the IRGAN framework. This reveals that the learning strategy significantly affects the adversarial ranking performance. (2) An in-depth comparison with conventional ranking methods shows that although the adversarial learning-to-rank models can achieve comparable performance as conventional methods based on neural networks, they are still inferior to LambdaMART by a large margin. In particular, we pinpoint that the weakness of adversarial learning-to-rank is largely attributable to the gradient estimation based on sampled rankings which significantly diverge from ideal rankings. Careful examination of this weakness is highly recommended for developing adversarial learning-to-rank approaches.

Extended author information available on the last page of the article

Keywords Learning-to-rank · Adversarial optimization · Variational divergence minimization · Reparameterization

1 Introduction

Nowadays, massive volumes of search requests are submitted daily to online search engines. In order to be able to accurately and efficiently provide desired information to users, one needs to solve many problems which still continue to constitute open challenges in both academic and industrial communities. A key problem is *ranking*, which has attracted significantly increasing attention in recent years across many fields, such as document retrieval and recommender systems. In this paper, we focus on the field of document retrieval. In particular, given a query and a set of documents to be ranked, the desired ranking model (or function) assigns a score to each document, then a ranked list of documents can be obtained by sorting the documents in descending order of scores. In general, the document with the highest score is assigned a rank of 1. In other words, the rank position of a document represents its relevance with respect to the query.

The modern approach is to use machine learning technologies to train the ranking model, and the research area called *learning-to-rank* (Liu, 2011) has emerged and become popular. In general, in the *training* phase, a number of queries are provided. Each query is associated with a set of documents to be ranked, of which the standard relevance labels (either binary or multi-graded judgments) are also included. Moreover, each query-document pair is represented through a feature vector. The objective of learning is to create a ranking model, which produces rankings of documents that best agree with the rankings derived from the relevance labels. In the phase of *testing*, given a new query, the learned ranking model is used to rank documents associated with this query. The advantages of learning-to-rank are straightforward. On one hand, compared with the traditional score-based models (such as TF-IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 1994)), it is a fully automatic learning process based on training data and can easily incorporate a large number of features. For example, a total of 700 different features are used in the datasets released for the Yahoo! learning to rank challenge (Chapelle & Chang, 2010). On the other hand, the rich learning frameworks (such as *support vector machines* and *deep neural networks*) enable the flexible development of powerful ranking models. The information retrieval (IR) community has experienced a rapid advancement and development of learning-to-rank methods, such as *pointwise* methods, *pairwise* methods and *listwise* methods. We refer the reader to Sect. 2.1 for the detailed differences among the three categories of learning-to-rank methods.

Recently, due to the breakthrough successes of generative adversarial network (GAN) (Goodfellow et al., 2014) and its variants (e.g., Zhang et al., 2019; Nowozin et al., 2016; Zhao et al., 2017) in learning generative models from complicated real-world data, the adversarial optimization framework has attracted increased attention. For example, significant efforts (Wang et al., 2017; He et al., 2018; Park et al., 2019; Wang et al., 2017, 2019; Lin et al., 2018) have been made to develop meaningful adversarial optimization methods for addressing learning-to-rank problems. Wang et al. (2017) proposed a unified framework for fusing generative and discriminative IR in an adversarial setting (IRGAN). The subsequent work (Park et al., 2019) further generated more difficult adversarial negative examples from the adversarially sampled negative examples. Different from the aforementioned studies, He et al. (2018) explored how to add adversarial perturbations

on model parameters so as to enhance the robustness of a recommender model and thus improve its generalization performance. The studies (Lin et al., 2018; Wang et al., 2017, 2019) investigated the effectiveness of adversarial optimization in image search.

Despite the success achieved by the aforementioned adversarial methods for learning-to-rank, there are still many open issues. First, the previous methods (Wang et al., 2017; Park et al., 2019) focus on optimizing either pointwise or pairwise ranking functions. In particular, they rely on rule-based sampling methods for generating adversarial samples. As a result, it becomes impossible to sample a list of documents. According to the previous studies (Cao et al., 2007; Qin et al., 2010; Yu et al., 2019), the listwise methods commonly show superior performance over the pointwise and pairwise methods. Natural questions arise then: *is it possible to perform listwise adversarial learning-to-rank*? *If possible, will the listwise adversarial learning-to-rank methods show superior performance over the pointwise and pairwise adversarial methods*? Second, in the context of web search, the previous methods (Wang et al., 2017; Park et al., 2019) were only evaluated against MQ2008-semi dataset. Compared with the widely used datasets for learning-to-rank evaluation, e.g., MSLRWEB30K (31, 531 queries), MQ2008-semi with only 784 queries is arguably small and outdated, which makes it hard to demonstrate well the robustness of an adversarial learning-to-rank method. Third, the previous methods (Wang et al., 2017; Park et al., 2019) merely explored how to adapt the GAN framework for ranking. However, GAN has many variants, such as training generative adversarial network using variational divergence minimization (f-GAN) (Nowozin et al., 2016) and the energy-based generative adversarial network model (EBGAN) (Zhao et al., 2017), thus a natural question arises: what is the effect of deploying a different adversarial learning strategy on adversarial learning-to-rank? Motivated by the aforementioned issues, we conducted an in-depth study of adversarial learning-to-rank by exploring how to perform adversarial learning-to-rank in a listwise manner and by adapting a different adversarial learning framework. The main contributions of this paper are summarized as follows:

1. Instead of being limited to either pointwise or pairwise adversarial learning-to-rank, we explore how to perform listwise adversarial learning-to-rank. Specifically, we cast the generation of rankings with respect to a query as sampling from a distribution. We appeal to the reparameterization trick to enhance the efficiency of sampling rankings. Thanks to this, the optimization of generator can be conducted *in situ*. To the best of our knowledge, this paper is the first to show how to perform listwise adversarial learning-to-rank.
2. In order to understand the effects of different adversarial learning strategies on adversarial learning-to-rank, we propose a new learning-to-rank framework via variational divergence minimization (IRf-GAN), which generalizes the previous adversarial ranking objective in Wang et al. (2017); Park et al. (2019). The pointwise, pairwise and listwise versions of IRf-GAN are explored, respectively. Thanks to IRf-GAN, we are able to understand well the differences and connections between different adversarial ranking methods.
3. We conduct a series of experiments using two benchmark collections. On one hand, IRf-GAN shows better performance than the typical adversarial ranking method IRGAN. On the other hand, we thoroughly investigate the impacts of factors, such as training order, activation function and divergence function on different adversarial ranking methods. Furthermore, we shed new light on the possible reason why adversarial learning-to-rank models are still inferior to conventional ranking methods, especially LambdaMART.

The remainder of this paper is structured as follows. In Sect. 2, we review the relevant prior literature. In Sect. 3, we describe the mathematical formulation of variational divergence minimization framework and listwise adversarial learning-to-rank. In Sect. 4, we introduce the experimental settings in detail. We conduct an in-depth analysis of the experimental results in Sect. 5. In Sect. 6, we provide an ablation study. Finally, we conclude the paper in Sect. 7.

2 Related work

In this section, we detail the related work on learning-to-rank and the techniques of adversarial learning as well as their applications for solving ranking problems.

2.1 Learning-to-rank

Learning-to-rank refers to a broad range of approaches that aim to tackle ranking problems using machine learning techniques. Conventional learning-to-rank approaches can be classified into three categories: *pointwise, pairwise, and listwise*. The key distinctions are the underlying hypotheses, loss functions, the input and output spaces. The pointwise approaches do not take into account the relative order among documents that are associated with the same query, and define loss functions on the basis of each individual document. The typical pointwise approaches include regression-based (Cossock & Zhang, 2006), classification-based (Nallapati, 2004), and ordinal regression-based algorithms (Chu & Ghahramani, 2005; Chu & Keerthi, 2005). The pairwise approaches care about the relative order between two documents. The goal of learning is to maximize the number of correctly ordered document pairs. The assumption is that the optimal ranking of documents can be achieved if all the document pairs are correctly ordered. Towards this end, many representative methods have been proposed (Burges et al., 2005; Joachims, 2002; Freund et al., 2003; Shen & Joshi, 2005; Yuan et al., 2016). The listwise approaches take all the documents associated with the same query in the training data as the input. In particular, there are two types of loss functions when performing listwise learning. For the first type, the loss function is related to a specific evaluation metric [e.g., nDCG (Järvelin & Kekäläinen, 2002) and ERR (Chapelle et al., 2009)]. Due to the non-differentiability and non-decomposability of the commonly used metrics, the methods of this type either try to optimize the upper bounds as surrogate objective functions (Chapelle et al., 2007; Xu & Li, 2007; Yue et al., 2007) or approximate the target metric using some smooth functions (Guiver & Snelson, 2008; Taylor et al., 2008; Qin et al., 2010). For the second type, the loss function is not explicitly related to a specific evaluation metric. The loss function reflects the discrepancy between the predicted ranking and the ground-truth ranking. Example algorithms include (Cao et al., 2007; Xia et al., 2008; Burges et al., 2006). Although no particular evaluation metrics are directly involved and optimized here, it is possible that the learned ranking function can achieve good performance in terms of evaluation metrics. We refer the reader to the work (Liu, 2011; Li, 2011) for a detailed review. In view of the successful applications of deep learning and the availability of well-maintained libraries for efficient development of deep learning techniques (e.g., PyTorch and TensorFlow), it becomes a popular choice to use deep neural networks as the basis to construct a scoring function when evaluating the above conventional learning-to-rank approaches, which are

referred to as *neural conventional learning-to-rank*. We note that this is also our focus in this work.

Inspired by the recent advances in neural networks, there have been significant interests in designing end-to-end ranking models based on neural networks (Huang et al., 2013; Shen et al., 2014; Guo et al., 2016; Hu et al., 2014; Pang et al., 2016; Wan et al., 2016), which are referred to as *neural learning-to-rank* (N-LTR). The term *end-to-end* means that the parameters in the feature extraction phase (one end) and the parameters of the ranking model (the other end) are jointly trained within a single architecture. For example, the ranking models, such as DSSM (Huang et al., 2013) and CDSSM (Shen et al., 2014), map both queries and documents into the same semantic space based on deep neural networks. The semantic space is assumed that the relevance score between a query and a document is proportional to the cosine similarity between their corresponding vectors. The follow-up studies (Guo et al., 2016; Hu et al., 2014; Pang et al., 2016; Wan et al., 2016; Bello et al., 2019) look into the inherent characteristics of information retrieval. The DRMM model by Guo et al. (2016) takes into account more factors, such as query term importance, exact matching signals, and diverse matching requirement. The methods like (Hu et al., 2014; Pang et al., 2016; Wan et al., 2016) first look at the local interactions between two texts, then design different network architectures to learn more complicated interaction patterns for relevance matching. We refer the reader to Onal et al. (2018); Guo et al. (2019) for an overview of N-LTR models.

After the above N-LTR methods came the “BERT revolution" for text ranking. A number of studies (Yilmaz et al., 2019; Nogueira & Cho, 2019; MacAvaney etal, 2019) explored how to fine-tune the bidirectional encoder representations from transformers (BERT) model (Devlin et al., 2019) to advance the ranking performance, e.g., for the passage re-ranking task based on MS MARCO. To reconcile efficiency and contextualization, ColBERT (Khattab & Zaharia, 2020) was proposed by introducing a late interaction architecture that independently encodes the query and the document. In this work, we focus on datasets consisting of feature vectors and refer the reader to the work (Lin et al., 2020) for a comprehensive overview of BERT-based learning-to-rank.

2.2 Adversarial learning for ranking

Recently, adversarial learning has attracted the deep learning community due to its remarkable contribution in estimating generative models. The concept of generative adversarial learning is introduced by Goodfellow et al. (2014). The primary vision of GAN is to learn the data probability distribution from a true training dataset (Chen et al., 2020). GANs were designed as substitutes to generative models and have the potential to model the true data effectively (Deshpande & Khapra, 2018). The development in generative adversarial learning has inspired various research tasks including network training (Arjovsky et al., 2017; Nowozin et al., 2016), computer vision applications such as image to image translation, high resolution image generation, video synthesis (Oza et al., 2020; Sheng et al., 2019), object identification (Wang et al., 2017), semantic segmentation (Xu & Wang, 2021) and visual tracking (Song et al., 2018). The benefit of adversarial learning is that generator learns to create similar image statistics to those of training examples so that the discriminator is not able to distinguish among the original and generated image (Song et al., 2018). Reed et al. (2016) designed an effective GAN model and training strategy that translates text to generate the realistic bird and flower images from human-written descriptions. Similar to this, Dong et al. (2017) proposed a GAN model that has the

capability to extract the semantic information from two modalities (i.e. text and image) and create the new realistic images from the combined semantics.

GANs have also achieved impressive performance on high dimensional configurations like word sequences, text generation and IR applications (Deshpande & Khapra, 2018). The generator in GANs attempts to model the distribution of training data and adversarial settings that appear to be a natural fit for IR. The relevant document for new queries can be retrieved through the learned distribution (Deshpande & Khapra, 2018). Zhang et al. (2017) proposed a TextGAN framework to generate realistic sentences through adversarial learning. They use an LSTM network as generator and a CNN as discriminator. The authors apply a kernelized discrepancy metric to match the distribution of latent features of real and synthetic sentences. In other research work (Lamb et al., 2016), in order to maintain the long term dependencies, researchers have included an extra discriminator for training a sequence-to-sequence language model.

Recently, significant efforts (Wei et al., 2017; Zou et al., 2019; Zeng et al., 2018; Feng et al., 2018; Singh et al., 2019; Montazeralghaem et al., 020; Xu et al., 2020; Yao et al., 2020; Wang et al., 2017; He et al., 2018; Park et al., 2019; Wang et al., 2017, 2019; Lin et al., 2018; Liu et al., 2020) have been made to develop new methods by deploying popular techniques, such as *reinforcement learning* and *adversarial learning*, to solve ranking problems, where *policy gradient* (the detailed definition is given in Sect. 3.1.1.1) is a core component during the optimization process. For example, inspired by the generative adversarial networks (GAN) (Goodfellow et al., 2014) and its variants, Wang et al. (2017) proposed IRGAN, which extends the GAN framework to learn a scoring function in an adversarial manner. The subsequent work (Park et al., 2019) further generates more difficult adversarial negative examples from the adversarially sampled negative examples. Different from the aforementioned studies, He et al. (2018) explored how to add adversarial perturbations on model parameters so as to enhance the robustness of a recommender model and thus improve its generalization performance. The studies (Lin et al., 2018; Wang et al., 2017, 2019) investigated the effectiveness of adversarial optimization in image search. Regarding learning-to-rank based on adversarial learning, we are not aware of any listwise approaches to date. As a result, this makes it difficult to fully understand the effectiveness of adversarial learning-to-rank. For example, a natural question might be about the effect of listwise sampling (rather than pointwise or pairwise sampling) on the performance. Moreover, given different kinds of adversarial ranking methods, there is no work that uniformly compares these methods and thoroughly investigates their corresponding shortcomings.

3 Adversarial learning-to-rank

In this section, we show how to perform adversarial learning-to-rank by adapting two different adversarial optimization frameworks, such as GAN and f-GAN. For the ranking framework that follows GAN's optimization strategy (referred to as IRGAN), we focus on how to perform adversarial learning-to-rank in a listwise manner. In Sect. 3.2, we propose a new adversarial learning-to-rank framework by following f-GAN 's optimization strategy (referred to as IRf-GAN). Then we describe the pointwise, pairwise and listwise instances of IRf-GAN.

3.1 IRGAN: adversarial learning-to-rank inspired by GAN

Let $\mathcal{Q}$ and $\mathcal{D}$ be the query space and the document space, respectively. We use $\Phi : \mathcal{Q} \times \mathcal{D} \rightarrow \mathcal{Z} := \mathbb{R}^d$ to denote the mapping function for generating a feature vector for a document with respect to a specific query, where $\mathbb{R}$ denotes the set of real numbers and $\mathcal{Z}$ represents the d-dimensional feature space. We use $\mathcal{T} := \mathbb{R}$ to denote the space of the ground-truth labels each document receives. Thus for each query, we have a list of m documents represented as feature vectors $\mathbf{x} = (x_1, ..., x_m) \in \mathcal{X} := \mathcal{Z}^m$ and a corresponding list $\mathbf{y}^* = (y_1^*, ..., y_m^*) \in \mathcal{Y} := \mathcal{T}^m$ of ground-truth labels. The subscript i like in x_i or y_i^* denotes the i-position in the list. In practice, we get independently and identically distributed (i.i.d) samples $\mathcal{S} = \{(\mathbf{x}_j, \mathbf{y}_j^*)\}_{j=1}^n$ from an unknown joint distribution $P(\cdot, \cdot)$ over $\mathcal{X} \times \mathcal{Y}$. A ranking π on m documents $\mathbf{x} = (x_1, ..., x_m)$ is defined as a permutation of $\mathbf{x}$. $\pi(i)$ / $\pi(x_i)$ yields the *rank* of the i-th document within $\mathbf{x}$. $\pi^{-1}(r)$ yields the index within $\mathbf{x}$ of the document at rank r, and we have $\pi^{-1}(\pi(i)) = i$ or $\pi^{-1}(\pi(x_i)) = i$. Since we are interested in sorting documents in descending order according to their relevance, we think of higher positions with smaller rank values as more favorable. A ground-truth ranking refers to the ideal ranking of documents that are sorted according to their real relevance to the query under consideration. We note that there are multiple ideal rankings for a query when we use graded relevance labels due to label ties.

We use $h : \mathbf{x} \rightarrow \mathbb{R}^m$ to denote the real-valued scoring function, which assigns each document a score. The scores of the documents associated with the same query, i.e., $\mathbf{y} = h(\mathbf{x}) = (h(x_1), h(x_2), ..., h(x_m))$, are used to sort the documents. Inspired by Goodfellow et al. (2014) and Wang et al. (2017), we are able to formulate the process of learning-to-rank as a game between two opponents: a *generator* and a *discriminator*. The generator aims to generate (or select) rankings that look like the ground-truth ranking, which may fool the discriminator. On the other hand, the discriminator aims to make a clear distinction between the ground-truth rankings and the ones generated by its opponent generator. The proposed framework for adversarial learning-to-rank is given as:

$$J_{IRGAN}(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \mathbb{E}_{\pi \backsim P_{true}(\pi|q_n)}[\log D_{\phi}(\pi \mid q_n)] + \mathbb{E}_{\pi \backsim P_{\theta}(\pi|q_n)}[\log(1 - D_{\phi}(\pi \mid q_n))] \tag{1}$$

where the generator G is denoted as $P_{\theta}(\pi \mid q_n)$ that aims to minimize the objective. On one hand, the generator fits the true distribution over all possible rankings $\pi \backsim P_{true}(\pi \mid q)$. On the other hand, it randomly generates rankings in order to fool the discriminator. The discriminator is denoted as $D_{\phi}(\pi \mid q_n)$, which estimates the probability of a ranking being either the ground-truth ranking or not. The objective of the discriminator is to maximize the log-likelihood of correctly distinguishing the ground-truth ranking from artificially generated rankings.

3.1.1 Listwise IRGAN

The previous methods (Wang et al., 2017; Park et al., 2019) have explored both pointwise and pairwise adversarial approaches for ranking based on the formulation of Eq. 1. In the following, we detail how to perform adversarial learning-to-rank method in a listwise manner. We elaborate on the optimization of each opponent as below.

3.1.1.1 Generator optimization of listwise IRGAN Given the currently optimized discriminator, we learn the generator via performing the minimization of Eq. 1:

$$\begin{aligned}\theta^* &= \min_{\theta} \sum_{n=1}^{N} \mathbb{E}_{\pi \backsim P_{true}(\pi|q_n)}[\log D_\phi(\pi \mid q_n)] + \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[\log(1 - D_\phi(\pi \mid q_n))] \\ &= \min_{\theta} \sum_{n=1}^{N} \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[\log(1 - D_\phi(\pi \mid q_n))]\end{aligned} \tag{2}$$

The term $\mathbb{E}_{\pi \backsim P_{true}(\pi|q_n)}[\log D_\phi(\pi \mid q_n)]$ is dropped because it does not depend on θ. A closer look at Eq. 2 reveals that $\log(1 - D_\phi(\pi \mid q_n))$ is usually in a large magnitude due to the fact that the probability of a ranking being correctly ordered (i.e., $D_\phi(\pi \mid q_n)$) is not always zero. As a result, the training process may become unstable, which is also pointed out by Wang et al. (2017). To cope with this issue, we drop the logarithm operation as suggested by Wang et al. (2017). Therefore, when $D_\phi(\pi \mid q_n)$ is very close to 0, the term $1 - D_\phi(\pi \mid q_n)$ approaches to 1 and the gradient works normally. Then Eq. 2 is rewritten as:

$$\theta^* = \min_{\theta} \sum_{n=1}^{N} \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[1 - D_\phi(\pi \mid q_n)] \tag{3}$$

Note that directly computing the gradient w.r.t. θ over the expectation in Eq. 3 is intractable, since the space of rankings is exponential in cardinality, especially in the listwise case. To overcome this issue, we appeal to the policy gradient. Specifically, the score function estimator (i.e., the REINFORCE algorithm) (Williams, 1992) makes use of the identity $\nabla_\theta p(z) = p(z)\nabla_\theta \log p(z)$, and the gradient can be derived as follows:

$$\begin{aligned}&\nabla_\theta \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[1 - D_\phi(\pi \mid q_n)] \\ &= \sum_{i=1}^{|\Omega_n|} \nabla_\theta P_\theta(\pi_i \mid q_n) \cdot (1 - D_\phi(\pi_i \mid q_n)) \\ &= \sum_{i=1}^{|\Omega_n|} P_\theta(\pi_i \mid q_n) \cdot \nabla_\theta \log P_\theta(\pi_i \mid q_n) \cdot (1 - D_\phi(\pi_i \mid q_n)) \\ &= \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[\nabla_\theta \log P_\theta(\pi \mid q_n) \cdot (1 - D_\phi(\pi \mid q_n))] \\ &\simeq \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log P_\theta(\pi_k \mid q_n) \cdot (1 - D_\phi(\pi_k \mid q_n))\end{aligned} \tag{4}$$

Namely, the gradient of the expected value over rankings sampled from $P_\theta(\pi \mid q_n)$ (i.e., the first step) can be derived as the expectation of the gradient of the log probability of each

sampled ranking multiplied by $1 - D_\phi(\pi \mid q_n)$ with respect to the corresponding ranking (i.e., the second step), where $\mid \Omega_n \mid$ denotes the size of the space of all possible rankings with respect to q_n. For the last step of Eq. 4, we compute the gradient with Monte-Carlo approximation, where π_k represents the k-th sample ranking generated by the current generator $P_\theta(\pi \mid q_n)$.

In this work, *the core of generator is designed to be a scoring function h_θ, which defines a distribution over the space of possible rankings for the query under consideration*. Furthermore, instead of generating new document feature vectors, we cast the generation of rankings with respect to a query as sampling from said distribution. Since we focus on how to learn effective scoring functions, we leave the case of generating new document feature vectors as a future work. We note that the previous methods (Wang et al., 2017; Park et al., 2019) also rely on sampling rather than generating new document feature vectors. Following the Plackett-Luce model (Plackett, 1975), given the documents $\mathbf{x}$ that are associated with the same query q_n and the relevance scores predicted by the generator $h_\theta(\mathbf{x}) = (h_\theta(x_1), ..., h_\theta(x_m))$, the probability of observing the ranking π is formulated as

$$P_\theta(\pi \mid q_n) = \prod_{i=1}^{m} \frac{\exp(h_\theta(x_{\pi^{-1}(i)}))}{\sum_{j=i}^{m} \exp(h_\theta(x_{\pi^{-1}(j)}))} \tag{5}$$

In other words, the generation of π can be described as the following sequential process: First, the probability of selecting a document x_k to the first rank position is proportional to $\frac{\exp(h_\theta(x_k))}{\sum_{i=1}^{m} \exp(h_\theta(x_i))}$. Second, once the first document is selected, the second document to fill the next rank position will be selected from the remaining documents in the same way. Third, the process repeats until m documents are selected. Inspired by the work of Bruch et al. (2020), we resort to the Gumbel-softmax trick (Jang et al., 2017; Maddison et al., 2017) in order to enhance the efficiency of sampling rankings with h_θ. Specifically, we associate an i.i.d sample drawn from *Gumbel*(0, 1) to each document for the query under consideration (i.e., $\mathbf{g} = g_1, ..., g_m$ for $\mathbf{x} = x_1, ..., x_m$). We then sort $\hat{\mathbf{y}} = \mathbf{g} + h_\theta(\mathbf{x})$ in an decreasing order. The corresponding re-ranking of $\mathbf{x}$ is regarded as a sample ranking of the generator.

3.1.1.2 Discriminator optimization of listwise IRGAN

Given the ground-truth rankings and the ones sampled with the current generator $P_{\theta^*}(\pi \mid q_n)$, the optimal parameters for the discriminator can be obtained as:

$$\phi^* = \max_\phi \sum_{n=1}^{N} \mathbb{E}_{\pi \sim P_{true}(\pi|q_n)}[\log D_\phi(\pi \mid q_n)] + \mathbb{E}_{\pi \sim P_{\theta^*}(\pi|q_n)}[\log(1 - D_\phi(\pi \mid q_n))] \tag{6}$$

In this work, *the core of discriminator is again designed to be a scoring function* h_ϕ. The formulation of discriminator again builds upon the aforementioned Plackett-Luce model. The probability of observing the given ranking π is formulated as:

$$D_\phi(\pi \mid q_n) = \prod_{i=1}^{m} \frac{\exp(h_\phi(x_{\pi^{-1}(i)}))}{\sum_{j=i}^{m} \exp(h_\phi(x_{\pi^{-1}(j)}))} \tag{7}$$

Algorithm 1 shows the overall structure of the proposed listwise adversarial learning-to-rank method, where the generator and discriminator are trained alternately.

Algorithm 1 Listwise IRGAN

```
Initialize generator and discriminator with random weights;
while count ≤ epoch_threshold do
    for d-steps do
        Train discriminator with Eq-6 based on the rankings generated with
        the current generator and the rankings derived from ground-truth
        labels
    end for
    for g-steps do
        Generate K sample rankings with the generator, then update the
        parameters with Eq-4.
    end for
    count++;
end while
```

In our empirical study of listwise IRGAN, we have found that: (1) Compared with the pointwise IRGAN, the performance would be significantly improved when performing adversarial ranking in a listwise manner. (2) Depending on the specific dataset, the listwise IRGAN may achieve significantly improved performance in comparison to the pairwise IRGAN.

3.2 IRf-GAN: adversarial learning-to-rank inspired by f-GAN

Inspired by f-GAN (Nowozin et al., 2016), we are able to formulate the adversarial ranking objective based on on the variational divergence minimization framework as follows:

$$J_{IRf-GAN}(\theta,\phi) = \min_{\theta}\max_{\phi}\sum_{n=1}^{N}(\mathbb{E}_{\pi\sim P_{true}(\pi|q_n)}[g_f(D_\phi(\pi \mid q_n))]- \mathbb{E}_{\pi\sim P_\theta(\pi|q_n)}[f^*(g_f(D_\phi(\pi \mid q_n)))]) \tag{8}$$

where f denotes the f-divergence, and f^* is the Fenchel conjugate of f. g_f is an output activation function which is specific to the adopted f-divergence. Table 1 shows our adopted instantiations of f-divergence, the corresponding conjugates and activation functions, where $\mathbb{R}\_$ represents the set of real negative numbers, t and v represent the input variables of f^* and g_f, respectively.

Table 1 The f-divergences adopted in this work

f-divergence f	Conjugate f^*	Domain	Output activation g_f
Kullback-Leibler	$exp(t-1)$	$\mathbb{R}$	v
GAN	$-log(1-exp(t))$	$\mathbb{R}\_$	$-log(1+exp(-v))$

Analogous to IRGAN, π represents a single document or a pair of documents or a ranking of documents respectively with respect to the cases of performing pointwise or pairwise or listwise adversarial learning. $P_\theta(\pi \mid q_n)$ represents the generator that aims to minimize the objective. On the one hand, the generator fits the true distribution over all documents per query $\pi \backsim P_{true}(\pi \mid q_n)$. On the other hand, it randomly samples documents in order to fool the discriminator. $D_\phi(\pi \mid q_n)$ represents the discriminator. In the case of a pointwise model, the discriminator estimates the probability of a document being either truly relevant or not. In the case of a pairwise model, the discriminator estimates the probability that a pair of documents is consistent with the ground-truth order or not. In the case of a listwise model, the discriminator estimates the probability of a ranking being either the ground-truth ranking or not. We refer to the adversarial learning-to-rank framework given by Eq. 8 as IRf-GAN.

Analogous to the superiority of f-GAN over GAN, the implementation of IRf-GAN enables us to obtain deeper insights into adversarial learning-to-rank from the following aspects. First, IRf-GAN generalizes the adversarial ranking objective proposed in Wang et al. (2017); Park et al. (2019). In other words, the adversarial ranking objective proposed in Wang et al. (2017); Park et al. (2019) is a special case of IRf-GAN by using the GAN divergence function correspondingly. Because f-GAN and GAN appeal to different optimization strategies, the performance differences between IRGAN and IRf-GAN indicate the effect of deploying a different optimization strategy on adversarial learning-to-rank. Second, given the flexibility of f-GAN that a rich family of f-divergences can be deployed, we can investigate well the effectiveness of using variational divergence minimization framework for adversarial learning-to-rank by deploying different f-divergences.

By finding a saddle-point of the min-max objective in IRf-GAN, we are able to jointly learn the parameters of generator and discriminator, which are essentially two scoring functions. We elaborate on the optimization of each opponent as below.

3.2.1 Generator optimization

Given the currently optimized discriminator, we learn the generator via performing the minimization of Eq. 8,

$$\begin{aligned}\theta^* &= \min_\theta \sum_{n=1}^{N} (\mathbb{E}_{\pi \backsim P_{true}(\pi|q_n)}[g_f(D_\phi(\pi \mid q_n))] - \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[f^*(g_f(D_\phi(\pi \mid q_n)))]) \\ &= \min_\theta \sum_{n=1}^{N} (-\mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[f^*(g_f(D_\phi(\pi \mid q_n)))])\end{aligned} \tag{9}$$

By making use of the identity $\nabla_\theta p(z) = p(z)\nabla_\theta \log p(z)$, the gradient can be derived as follows:

$$\begin{aligned}&\nabla_\theta \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[f^*(g_f(D_\phi(\pi \mid q_n)))] \\ &\quad = \mathbb{E}_{\pi \backsim P_\theta(\pi|q_n)}[\nabla_\theta \log P_\theta(\pi \mid q_j) \cdot (f^*(g_f(D_\phi(\pi \mid q_n))))] \\ &\quad \approx \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log P_\theta(\pi_k \mid q_n) \cdot (f^*(g_f(D_\phi(\pi_k \mid q_n))))\end{aligned} \tag{10}$$

For the last step, the gradient is computed with Monte-Carlo approximation, where π_k represents the k-th sample.

Analogous to IRGAN, *the core of generator is again designed to be a scoring function h_θ with parameters θ*. In the case of a pointwise model, the generator is formulated via a softmax function,

$$P_\theta(\pi_k \mid q_n) = \frac{\exp\left(h_\theta(\pi_k)\right)}{\sum_{\pi_j} \exp\left(h_\theta(\pi_j)\right)} \tag{11}$$

In the case of a pairwise model, we appeal to the Bradley-Terry model (Bradley & Terry, 1952). Given the scoring function h_θ, the probability that document x_i "wins" over document x_j (i.e., x_i is regarded as more relevant than x_j) is expressed as $P_{BT}(x_i > x_j) = \frac{1}{1+e^{-(h_\theta(x_i)-h_\theta(x_j))}}$. When generating document pairs, we first determine the order between two documents based on a Bernoulli trial, then randomly select a specified number of document pairs among success trials. In the case of a listwise model, the generator is formulated in the same way as the case of IRGAN in Sect. 3.1.1.1.

3.2.2 Discriminator optimization

Given the samples obtained via ground-truth labels and the ones sampled with the current generator $P_{\theta^*}(\pi \mid q_n)$, the optimal parameters for the discriminator can be obtained as:

$$\phi^* = \max_\phi \sum_{n=1}^{N} (\mathbb{E}_{\pi \sim P_{true}(\pi|q_n)}[g_f(D_\phi(\pi \mid q_n))] - \mathbb{E}_{\pi \sim P_{\theta^*}(\pi|q_n)}[f^*(g_f(D_\phi(\pi \mid q_n)))]) \tag{12}$$

Similar to the generator, *the core of discriminator is again designed to be a scoring function h_ϕ* with parameters ϕ, and we compute the policy gradient during the optimization process. In the case of a pointwise model, the discriminator's output is straightforwardly used without any further transformation. In the case of a pairwise model, the formulation again builds upon the Bradley-Terry model (Bradley & Terry, 1952). The probability that a pair of documents is consistent with the standard order is given as $D_\phi(\pi =< x_i, x_j > \mid q_n) = \frac{1}{1+e^{-(h_\phi(x_i)-h_\phi(x_j))}}$. In the case of a listwise model, the discriminative score is expressed as the probability of observing a given ranking π based on Plackett-Luce model as the listwise case of IRGAN in Sect. 3.1.1.2.

Algorithm 2 shows the overall structure of the proposed adversarial learning-to-rank based on variational divergence minimization. In particular, motivated by the success of the alternating gradient method with a single inner step (Nowozin et al., 2016), for each training query, we train the generator and the discriminator alternately rather than using a double-loop like in Wang et al. (2017); Park et al. (2019).

Algorithm 2 Adversarial learning-to-rank based on variational divergenc minimization

```
Initialize generator and discriminator with random weights.
count = 0
while count < epoch_threshold do
    for each training query do
        Train discriminator with Eq. 12 based on the samples generated
        with the current generator and the ones derived from ground-trut
        labels.
        Generate K samples with the generator, then update the
        parameters based on Eq. 10.
    end for
    count++;
end while
```

In our empirical study of IRf-GAN, we have found that: (1) The pointwise, pairwise and listwise models may reach different levels of performance. (2) When compared with the adversarial ranking framework of IRGAN, the pairwise IRf-GAN can achieve significantly improved performance, which shows that the learning strategy significantly affects the adversarial ranking performance.

4 Experimental setup

In our experiments, we used two benchmark datasets, where each query-document pair is represented with a feature vector. The basic statistics of each dataset without performing any filtering are listed in Table 2. The ground-truth labels take 5 values from 0 (irrelevant) to 4 (perfectly relevant). It is noteworthy that ranking data is typically long-tailed. The non-relevant documents account for the largest ratio, the number of relevant documents becomes rather small with the increase of relevance level.

Table 2 Statistics of the benchmark datasets

	MSLRWEB30K		Yahoo (Set1)	
#Queries	31,531		29,921	
#Docs	3,771,125		709,877	
#Features	136		700	
#Avg relevant docs per query	58.0		17.5	
#Docs per query (Min; Avg; Max)	(1; 119.6; 1,251)		(1; 23.7; 139)	
#Ground-truth labels & distribution	0	1,940,952	0	185,192
	1	1,225,770	1	254,110
	2	504,958	2	202,700
	3	69,010	3	54,473
	4	30,435	4	13,402

Given the above raw datasets, we find that there are 982 and 1135 queries in MSLR-WEB30K and Yahoo (Set1) respectively, which have no relevant documents at all. We filtered out these queries due to the fact that they provide no training signal. Furthermore, we limit the minimum number of documents (including both relevant or non-relevant documents) per query as 10, since the one in a search engine result page (SERP) is usually set as 10. Given the processed datasets, we use the training data to learn the ranking model, use the validation data to select the hyper parameters based on nDCG@5, and use the testing data for evaluation. To reduce the possible impact of overfitting on performance comparison based on MSLRWEB30K, we use all the five folds and perform 5-fold cross validation. We use nDCG to measure the performance, which takes into account both rank position and relevance level. The results are reported for different cutoff values 1, 3, 5, 10 and 20 to show the performance of each method. In this work, a number of representative approaches are compared: LambdaMART (Wu et al., 2010) [the implementation included in LightGBM (Ke et al., 2017)], ListNet (Cao et al., 2007), ListMLE (Xia et al., 2008) and WassRank (Yu et al., 2019) are used to represent the conventional ranking approaches.

We implemented and trained both the proposed method and the baseline approach by Wang et al. (2017) (i.e., IRGAN-Point and IRGAN-Pair) using PyTorch v1.8, where one Nvidia Titan RTX GPU with 24 GB memory is used.[1] By taking the network and learning settings in (Wang et al., 2017) as a reference, we used a uniform 5-layer feed-forward neural network, where the size of a hidden layer is set as 100. We tested different activation functions (*ReLU*, *GELU*), and the best result is reported. We always utilized the same seed when training the models from scratch. We used the L2 regularization with a decaying rate of 1×10^{-3} and the Adam optimizer with a learning rate of 1×10^{-3}. The number of epochs over training data is 100. For IRGAN, the pointwise, pairwise and listwise versions are denoted as IRGAN-Point, IRGAN-Pair and IRGAN-List, respectively. The inner loop for training both generator and discriminator is set as $1 : 1$. The temperature is set as 0.5. The number of sampled documents or document pairs is tested with 5 and 20, respectively. Analogously, the pointwise, pairwise and listwise versions of IRf-GAN are denoted as IRf-GAN-Point, IRf-GAN-Pair and IRf-GAN-List, respectively. The following divergence functions are tested: Kullback–Leibler (KL) and GAN. For an adversarial ranking method, both generator and discriminator can be used to rank the documents.

5 Result and analysis

In Tables 3 and 4, we show the overall performance of the tested approaches on each dataset, respectively. For adversarial ranking methods, they are differentiated as follows: IRGAN-Pair (D-GD-ReLU) refers to the pairwise adversarial ranking method following the IRGAN framework. Its best performance is achieved with the setting of D-GD-ReLU: using discriminator (i.e., D) for final ranking; the training order (i.e., GD) between generator and discriminator the generator is trained first; ReLU is the adopted activation function. IRf-GAN-List(D-GD-GELU-KL) refers to the listwise adversarial ranking method following the IRf-GAN framework. Its best performance is achieved with the setting of D-GD-GELU-KL: using discriminator (i.e., D) for final ranking. The training order (i.e.,

[1] The main source code for reproducing our experimental results is available via: https://github.com/wildltr/ptranking.

Table 3 The performance of involved approaches on MSLRWEB30K

Methods	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@20
ListNet	0.4665	0.4495	0.4534	0.4708	0.4942
ListMLE	0.4637	0.4474	0.4507	0.4668	0.4881
WassRank	0.4684	0.4460	0.4473	0.4611	0.4814
LambdaMART	**0.4933**	**0.4743**	**0.4776**	**0.4948**	**0.5166**
IRGAN-Point (G-DG-GELU)	0.4132	0.4064	0.4116	0.4290	0.4523
IRGAN-Pair (D-GD-ReLU)	0.4452	0.4277	0.4321	0.4476	0.4687
IRGAN-List (D-DG-GELU)	0.4163	0.3940	0.3961	0.4139	0.4407
IRf-GAN-Point (G-DG-GELU-KL)	0.3985	0.3879	0.3924	0.4096	0.4350
IRf-GAN-Pair (D-DG-GELU-KL)	*0.4693*	*0.4488*	*0.4501*	*0.4631*	*0.4794*
IRf-GAN-List (D-GD-GELU-KL)	0.4267	0.4028	0.4039	0.4206	0.4458

The best result is indicated in bold, while the second and third best results are denoted in underline and italics, respectively

Table 4 The performance of involved approaches on Yahoo (Set1)

Methods	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@20
ListNet	*0.6699*	0.6597	0.6734	0.7193	0.4874
ListMLE	0.6609	0.6516	0.6661	0.7156	0.4863
WassRank	0.6714	*0.6593*	*0.6726*	*0.7188*	*0.4873*
LambdaMART	**0.7078**	**0.6985**	**0.7112**	**0.7520**	**0.5068**
IRGAN-Point (G-DG-ReLU)	0.5328	0.5583	0.5870	0.6537	0.4461
IRGAN-Pair (D-GD-ReLU)	0.6000	0.6042	0.6245	0.6819	0.4670
IRGAN-List (G-GD-ReLU)	0.6588	0.6451	0.6584	0.7058	0.4764
IRf-GAN-Point (G-DG-ReLU-KL)	0.5916	0.5961	0.6186	0.6749	0.4601
IRf-GAN-Pair (G-DG-ReLU-KL)	0.6647	0.6512	0.6652	0.7131	0.4834
IRf-GAN-List (D-GD-GELU-GAN)	0.6498	0.6478	0.6625	0.7079	0.4778

The best result is indicated in bold, while the second and third best results are denoted in underline and italics, respectively

GD) between generator and discriminator means that the generator is trained first. The adopted activation function is GELU, while KL is used as the f-divergence function.

We first look at the performances of conventional learning-to-rank approaches, namely LambdaMART, ListNet, ListMLE and WassRank. We can observe that: (1) LambdaMART achieves significantly better performance than other approaches on MSLRWEB30K, which are consistent with previous studies (Wang et al., 2018; Yu et al., 2019; Ai et al., 2018). The main reasons are that: the objective optimized by LambdaMART is a coarse upper bound of nDCG (Wang et al., 2018). Benefiting from GBDT in the form of an ensemble of weak prediction models and the algorithmic and engineering optimizations of LightGBM, LambdaMART shows more promising results. (2) The neural network based approaches (i.e., ListNet, ListMLE and WassRank) rely upon different loss functions, and show different performance. In particular, WassRank has slightly better performance. This echoes the findings in prior study by Yu et al. (2019), which have shown that the adopted loss function by WassRank is more

Table 5 An in-depth comparison between IRGAN and IRf-GAN on MSLRWEB30K

Methods	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@20
IRGAN-Point (G-DG-GELU)	0.4132	*0.4064*	*0.4116*	0.4290	0.4523
IRGAN-Pair (D-GD-ReLU)	0.4452	**0.4277**	**0.4321**	**0.4476**	**0.4687**
IRGAN-List (D-DG-GELU)	0.4163	0.3940	0.3961	0.4139	0.4407
IRf-GAN-Point (G-DG-GELU-GAN)	0.3877	0.3786	0.3842	0.4027	0.4288
IRf-GAN-Pair (D-DG-ReLU-GAN)	**0.4487**	0.4259	0.4227	*0.4266*	0.4382
IRf-GAN-List (D-GD-ReLU-GAN)	*0.4256*	0.4033	0.4047	0.4216	*0.4467*

The best result is indicated in bold, while the second and third best results are denoted by underline and italics, respectively

Table 6 An in-depth comparison between IRGAN and IRf-GAN on Yahoo (Set 1)

Methods	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@20
IRGAN-Point (G-DG-ReLU)	0.5328	0.5583	0.5870	0.6537	0.4461
IRGAN-Pair (D-GD-ReLU)	0.6000	0.6042	0.6245	0.6819	0.4670
IRGAN-List (G-GD-ReLU)	**0.6588**	0.6451	0.6584	*0.7058*	*0.4764*
IRf-GAN-Point (G-DG-GELU-GAN)	0.5757	0.5853	0.6090	0.6662	0.4533
IRf-GAN-Pair (G-DG-ReLU-GAN)	0.6554	*0.6440*	*0.6572*	0.7070	**0.4788**
IRf-GAN-List (D-GD-GELU-GAN)	*0.6498*	**0.6478**	**0.6625**	**0.7079**	0.4778

The best result is indicated in bold, while the second and third best results are denoted in underline and italics, respectively

consistent with the evaluation metric (e.g., nDCG) enabling it be more effective in capturing position importance. According to the latest work by Qin et al. (2021), neural network based approaches can perform competitively well with LambdaMART, where a number of strategies are deployed, such as feature transformation, data augmentation and listwise context. Due to time constraints, we leave it as a future work to test the effectiveness of these strategies.

We next look at the performance of each adversarial ranking approach in Tables 3 and 4. We can observe that depending on the specific implementation (namely pointwise, pairwise and listwise), the generative and discriminative components may reach different levels of performance. Specifically, for the adversarial ranking framework of IRGAN, the discriminator tends to achieve a better performance that the generator. For the adversarial ranking framework of IRf-GAN, the discriminator achieves the best performance on MSLRWEB30K, but shows a lower performance on Yahoo (Set1). We refer the reader to Sect. 2.1 for a detailed analysis of the effects of different factors, such as training order, activation function and divergence function. By comparing the best performance achieved by either generator or discriminator, we can see that: IRf-GAN outperforms IRGAN across two benchmark datasets, and IRf-GAN-pair achieves the best performance among all adversarial methods. The most possible explanation is that: IRf-GAN generalizes the adversarial ranking objective optimized by IRGAN. Based on f-divergence minimization, IRf-GAN provides us the flexibility of investigating different divergence functions.

As mentioned in Sect. 3.2, IRGAN is a special case of IRf-GAN by using the GAN divergence function. In order to clearly show the effects of different adversarial

training frameworks on ranking performance, Tables 5 and 6 show a subtle comparison between IRGAN and IRf-GAN, where *only the GAN divergence function is configured* in IRf-GAN.

From Tables 5 and 6, we can observe that: IRf-GAN outperforms IRGAN on MSLRWEB30K in the cases of pairwise and listwise configurations, and achieves better performance on Yahoo (Set 1) in the cases of pointwise and pairwise and configurations. In a nutshell, the results demonstrate that the adversarial learning framework matters significantly when performing adversarial learning-to-rank.

Now, we focus on investigating the effectiveness of adversarial ranking by comparing with the conventional learning-to-rank approaches. Tables 3 and 4 reveal that adversarial ranking is able to achieve comparable performance to traditional learning-to-rank methods based on neural networks. However, it is, by a large margin, inferior to LambdaMART. It is noteworthy that our experimental results echo the findings by Yu et al. (2022), but are not consistent with the prior study (Wang et al., 2017), which showed that adversarial ranking methods can significantly outperform LambdaMART. In order to justify our findings, on the one hand, we note that our adopted datasets, such as MSLRWEB30K and Yahoo (Set1), are more recent and significantly larger than the single dataset (i.e., MQ2008-semi) used by Wang et al. (2017). Compared with MQ2008-semi that has 46 features and only 784 queries, both MSLRWEB30K and Yahoo (Set1) use more features (136 and 700) and more queries (31, 531 and 29, 921). Moreover, both MSLRWEB30K and Yahoo (Set1) use 5-level graded relevance judgements, ranging from 0 (not relevant) to 4 (perfectly relevant). It is reasonable to say that our experimental results based on multiple larger datasets are more reliable. On the other hand, in the following we pinpoint the potential drawbacks of adversarial learning-to-rank approaches, which also indicate the shortcoming of the prior study (Wang et al., 2017).

Given the benchmark collections with complete information (where relevance labels are known for all items), the loss functions of conventional learning-to-rank approaches are commonly defined based on ground-truth labels, such as ListNet, WassRank and LambdaMART. In particular, the fine-grained relevance levels (i.e., {0, 1, 2, 3, 4}) can be further utilized to calibrate the training loss. For adversarial ranking methods, the training data for discriminator (including different cases of pointwise, pairwise and listwise) includes samples generated by the generator and is not a pure set of samples derived based on ground-truth labels. Looking back at the Eqs. 4, 6, 10 and 12, we can observe that: (1) A common core component is that the gradient of expected value over the distribution of rankings is derived as the expectation of the gradient of the log probability of each sampled ranking multiplied by a scalar. The gradient computation is further approximated using Monte Carlo sampling. (2) The key difference among the different adversarial ranking approaches is the way in how to weigh each individual document within the samples.

Going further, the aforementioned observations enable us to investigate the potential weaknesses that adversarial learning-to-rank approaches suffer from. First, document ranking is characterized by having a small number of relevant documents and a large number of non-relevant documents, which is illustrated in Table 2. Therefore, there is a high probability of selecting either non-relevant or less-relevant documents in samples. Take the query from MSLRWEB30K (query-id: 631, label distribution: 0:96, 1:31, 2:3, 3:2, 4:1) for example. The probability of selecting a highly relevant document (i.e., with a label of 4) is rather small. As a result, the samples significantly diverge from the ideal ones directly derived based on ground-truth labels. Taking the listwise adversarial approach for example, the estimated relevance of a document x_i is proportional to counts of pairwise comparisons that x_i "beats" other documents (i.e., x_i is selected at a higher position). Noteworthy, due to

Table 7 The ablation study of IRGAN and IRf-GAN on MSLRWEB30K

Methods	Training order		Activation function		Divergence function	
	GD	DG	ReLU	GELU	KL	GAN
IRGAN-Point (G)		✓		✓	–	–
IRGAN-Point (D)		✓	✓		–	–
IRGAN-Pair (D)	✓		✓		–	–
IRGAN-List (G)	✓			✓	–	–
IRGAN-List (D)		✓		✓	–	–
IRf-GAN-Point (G)		✓		✓	✓	
IRf-GAN-Point (D)		✓		✓		✓
IRf-GAN-Pair (G)		✓		✓	✓	
IRf-GAN-Pair (D)		✓		✓	✓	
IRf-GAN-List (D)	✓			✓	✓	

the fact that there are a large number of queries, the sampled rankings per query commonly include many wrongly ranked document pairs (contradicting the standard order). It is rarely possible that the optimization can converge to the optimal scoring function. This observation echoes the findings in the recent study by Yu et al. (2022). Their work shows that the recent ranking methods based on either reinforcement learning or adversarial learning boil down to policy-gradient-based optimization, which are, by a large margin, inferior to many conventional ranking methods. The failures are largely attributable to the gradient estimation based on sampled rankings, which significantly diverge from ideal rankings. In particular, the larger the number of documents per query and the more fine-grained the ground-truth labels, the greater the impact policy-gradient-based ranking suffers. Second, policy gradient based methods suffer from instability of gradient estimates. Though a number of recent methods (Xu et al., 2020a, b; Shen et al., 2019) have been proposed to mitigate this problem to some extent, which still remains an open challenge.

In view of the subtle differences among adopted loss functions, it is reasonable to say that the performance of conventional learning-to-rank approaches based on a pure set of samples reveals a loose upper bound of the discriminator. Due to this inherent limitation of the discriminator, the optimization of generator will be impacted since the discriminator is used as a weighting factor. As a result, the overall adversarial training process will be affected. The observation that the discriminator tends to show a better performance than the generator also echoes our analysis, which is also observed study by Yu et al. (2022).

6 Ablation study

In this section, we provide in Tables 7 and 8 ablation study's results to highlight the impact of each configuration on adversarial learning-to-rank approaches, such as training order, activation function and divergence function.

For a pointwise/pairwise/listwise adversarial ranking model, the *check mark* indicates settings of the ranking component (either generator or discriminator) that achieve the best performance among the possible configurations. The *underlined check mark* implies the best settings of the remaining ranking component that did not achieve the best performance. Three running cases (IRGAN-Pair(G), IRf-GAN-List(G) on MSLRWEB30K and

Table 8 The ablation study of IRGAN and IRf-GAN on Yahoo (Set1)

Methods	Training order		Activation function		Divergence function	
	GD	DG	ReLU	GELU	KL	GAN
IRGAN-Point (G)		✓	✓		–	–
IRGAN-Point (D)	✓			✓	–	–
IRGAN-Pair (G)	✓		✓		–	–
IRGAN-Pair (D)	✓		✓		–	–
IRGAN-List (G)	✓		✓		–	–
IRGAN-List (D)	✓			✓	–	–
IRf-GAN-Point (G)		✓	✓		✓	
IRf-GAN-Point (D)		✓		✓		✓
IRf-GAN-Pair (G)		✓	✓		✓	
IRf-GAN-Pair (D)		✓	✓		✓	
IRf-GAN-List (D)	✓			✓		✓

IRf-GAN-List(G) on Yahoo (Set1)) are not taken into account due to having failed to converge. Both check mark and underlined check mark can be viewed as a vote when selecting the possible configurations. Check mark has a relatively higher voting weight than underlined check mark.

From Tables 7 and 8, we can observe that: (1) For training order, IRf-GAN shows that DG is a better choice, but IRGAN behaves differently given different datasets. (2) For activation function, there is no consistent setting for achieving the best performance for both IRGAN and IRf-GAN given different datasets. (3) A closer look at the effect of divergence function on IRf-GAN shows that KL is a better choice compared with GAN. To summarize, the factors, such as training order, activation function and divergence function, significantly affect the performance of both IRGAN and IRf-GAN. Careful examination of these factors is highly recommended in the development of adversarial learning-to-rank approaches.

7 Conclusion

In this paper, we aim to provide an in-depth study on adversarial learning-to-rank. First, we propose a new way on how to perform adversarial learning-to-rank in a listwise manner. Second, in order to understand well the effects of using different adversarial learning strategies, we propose a general adversarial ranking framework IRf-GAN based on variational divergence minimization by following (Nowozin et al., 2016). The pointwise, pairwise and listwise versions of IRf-GAN are further formulated. The experimental results based on multiple benchmark datasets demonstrate the superiority of IRf-GAN-pair over adversarial learning-to-rank methods based on IRGAN framework, which indicates that the learning strategy significantly affects the adversarial ranking performance. The ablation study based on training order, activation function and divergence function reveals that careful examinations of these hyper-parameters are highly important. Finally, because adversarial ranking methods boil down to policy gradient based optimization, we indicate that the weakness of adversarial ranking methods is largely attributable to the gradient

estimation based on sampled rankings which significantly diverge from ideal rankings. This largely explains why adversarial learning-to-rank models are still inferior to conventional ranking methods.

Since ranking is a core step in a variety of applications, such as recommender and question answering systems, we believe that our framework provides a new perspective for addressing problems of this kind. Our work also opens up many interesting future research directions. First, it should be quite interesting to further test alternative adversarial training strategies for ranking. Second, we plan to extend our evaluation by exploring different types of datasets, such as MS MARCO consisting of raw text queries and documents. Third, we plan to conduct an in-depth investigation on the convergence of adversarial optimization. Fourth, it is interesting to evaluate the adversarial learning-to-rank methods based on datasets across multiple domains, such as question answering and image retrieval. Finally, we also plan to conduct a comparative study to show the impact of label noise on adversarial ranking models and conventional ranking methods. For example, the label noise can be obtained by masking a specified number of documents as unlabeled documents or randomly swapping the documents' ground-truth labels. Moreover, we just used simple feed-forward neural networks and did not conduct an in-depth investigation on the impact of different neural architectures. From an optimization perspective, there is no guarantee of optimality for a pre-specified architecture like ours. However we do note that the technique of neural architecture search (NAS) (Elsken et al., 2019) can be applied, which is also planed as a future work.

Acknowledgements This work was partially supported by the commissioned project, JPNP20006, by the New Energy and Industrial Technology Development Organization (NEDO).

References

Ai, Q., Bi, K., Guo, J., & Croft, W.B. (2018). Learning a deep listwise context model for ranking refinement. In *Proceedings of the 41st SIGIR* (pp. 135–144).

Arjovsky, M., Chintala, S., & Bottou, L. (2017). (WGAN) Wasserstein generative adversarial network junhong huang. In *ICML* (pp. 1–44).

Bello, I., Kulkarni, S., Jain, S., Boutilier, C., Chi, E., Eban, E., & Meshi, O. (2019). Seq2Slate: Re-ranking and slate optimization with RNNs. In *Proceedings of the workshop on negative dependence in machine learning*.

Bradley, R.A., & Terry, M.E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(34).

Bruch, S., Han, S., Bendersky, M., & Najork, M. (2020). A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th WSDM* (pp. 61–69).

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd ICML* (pp. 89–96).

Burges, C.J.C., Ragno, R., & Le, Q.V. (2006). Learning to rank with nonsmooth cost functions. In *Proceedings of NeurIPS* (pp. 193–200).

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th ICML* (pp. 129–136).

Chapelle, O., & Chang, Y. (2010). Yahoo! learning to rank challenge overview. In *Proceedings of the 2010 international conference on YLRC* (pp. 1–24).

Chapelle, O., Le, Q., & Smola, A. (2007). Large margin optimization of ranking measures. *NIPS workshop on Machine Learning for Web Search*.

Chapelle, O., Metlzer, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th CIKM* (pp. 621–630).

Chen, Y., Zhao, Y., Jia, W., Cao, L., & Liu, X. (2020). Adversarial-learningbased image-to-image transformation: A survey. *Neurocomputing, 411*, 468–486. https://doi.org/10.1016/j.neucom.2020.06.067

Chu, W., & Ghahramani, Z. (2005). Gaussian processes for ordinal regression. *Journal of Machine Learning Research, 6*, 1019–1041.
Chu, W., & Keerthi, S.S. (2005). New approaches to support vector ordinal regression. In *Proceedings of the 22nd ICML* (pp. 145–152).
Cossock, D., & Zhang, T. (2006). Subset ranking using regression. In *Proceedings of the 19th annual conference on learning theory* (pp. 605–619).
Deshpande, A., & Khapra, M.M. (2018). Dissecting an adversarial framework for information retrieval.
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019* (pp. 4171–4186).
Dong, H., Yu, S., Wu, C., & Guo, Y. (2017). Semantic image synthesis via adversarial learning. In Proceedings of the IEEE international conference on computer vision (Vol. 2017-Octob, pp. 5707–5715). https://doi.org/10.1109/ICCV.2017.608
Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research, 20*(55), 1–21.
Feng, Y., Xu, J., Lan, Y., Guo, J., Zeng, W., & Cheng, X. (2018). From greedy selection to exploratory decision-making: diverse ranking with policy- value networks. In *Proceedings of SIGIR* (pp. 125–134).
Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research, 4*, 933–969.
Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of NeurIPS* (pp. 2672–2680).
Guiver, J., & Snelson, E. (2008). Learning to Rank with SoftRank and Gaussian processes. In *Proceedings of the 31st SIGIR* (pp. 259–266).
Guo, J., Fan, Y., Ai, Q., & Croft, W.B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th CIKM* (pp. 55–64).
Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., & Cheng, X. (2019). A deep look into neural ranking models for information retrieval. *Information Processing and Management*.
He, X., He, Z., Du, X., & Chua, T.-S. (2018). Adversarial personalized ranking for recommendation. In *Proceedings of SIGIR* (pp. 355–364).
Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Proceedings of 27th NIPS* (pp. 2042–2050).
Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of CIKM* (pp. 2333–2338).
Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel–Softmax. In *International Conference on Learning Representations (ICLR)*.
Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems, 20*(4), 422–446.
Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th KDD* (pp. 133–142).
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of NeurIPS* (pp. 3149–3157).
Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of SIGIR* (pp. 39–48).
Lamb, A.M., Goyal, A.G.A.P., Zhang, Y., Zhang, S., Courville, A.C., & Bengio, Y. (2016). Professor forcing: A new algorithm for training recurrent networks. *Advances in Neural Information Processing Systems* (pp. 4601–4609).
Li, H. (2011). Learning to Rank for Information Retrieval and Natural Language Processing (Vol. 4) (No. 1). *Synthesis Lectures on Human Language Technologies*.
Lin, J., Nogueira, R., & Yates, A. (2020). Pretrained transformers for text ranking: BERT and beyond. arXiv:2010.06467 .
Lin, K., Yang, F., Wang, Q., & Piramuthu, R. (2018). Adversarial learning for fine-grained image search. In *Proceedings of ICME* (pp. 490–495).
Liu, J., Dou, Z., Wang, X., Lu, S., & Wen, J. (2020). Dvgan: a minimax game for search result diversification combining explicit and implicit features. In *Proceedings of SIGIR* (p. 479–488).
Liu, T.-Y. (2011). *Learning to rank for information retrieval*. Springer.
MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019). CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd SIGIR* (pp. 1101–1104).
Maddison, C.J., Mnih, A., & Teh, Y.W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*.
Montazeralghaem, A., Zamani, H., & Allan, J. (2020). A reinforcement learning framework for relevance feedback. In *Proceedings of SIGIR* (pp. 59-68).

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th SIGIR* (pp. 64–71).

Nogueira, R., & Cho, K. (2019). Passage Re-ranking with BERT. arXiv:1901.04085v4.

Nowozin, S., Cseke, B., & Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 271–279).

Onal, K. D., Zhang, Y., & Altingovde, I. S., Others. (2018). Neural information retrieval: At the end of the early years. *Journal of Information Retrieval, 21*(2–3), 111–182.

Oza, M., Vaghela, H., & Srivastava, K. (2020). Progressive generative adversarial binary networks for music generation. In *International conference on innovative computing and communications: proceedings of ICICC 2019*, volume 1 (Vol. 1087, p. 181).

Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., & Cheng, X. (2016). Text matching as image recognition. In *Proceedings of AAAI conference on artificial intelligence* (pp. 2793–2799).

Park, D.H., & Chang, Y. (2019). Adversarial sampling and training for semi- supervised information retrieval. In Proceedings of the web conference (pp. 1443–1453).

Plackett, R. L. (1975). The analysis of permutations. *Journal of the Royal Statistical Society. Series C, 24*(2), 193–202.

Qin, T., Liu, T.-Y., & Li, H. (2010). A general approximation framework for direct optimization of information retrieval measures. *Journal of Information Retrieval, 13*(4), 375–397.

Qin, Z., Yan, L., Zhuang, H., Tay, Y., Pasumarthi, R.K.,Wang, X., & Najork, M. (2021). Are neural rankers still outperformed by gradient boosted decision trees?. In *Proceedings of ICLR*.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. In *33rd International conference on machine learning, ICML 2016*, 3, 1681-1690. arXiv:1605.05396

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at TREC-3. In *Proceedings of TREC*.

Shen, L., & Joshi, A. K. (2005). Ranking and Reranking with Perceptron. *Machine Learning, 60*(1–3), 73–96.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd WWW* (pp. 373-374).

Shen, Z., Ribeiro, A., Hassani, H., Qian, H., & Mi, C. (2019). Hessian aided policy gradient. In *ICML* (pp. 5729-5738).

Sheng, L., Pan, J., Guo, J., Shao, J., Wang, X., & Loy, C.C. (2019). Unsupervised bi-directional flow-based video generation from one snapshot. arXiv preprint arXiv:1903.00913 .

Singh, A., & Joachims, T. (2019). Policy learning for fairness in ranking. In *Proceedings of NeurIPS* (pp. 5426–5436).

Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., & Yang, M.H. (2018). VITAL: visual tracking via adversarial learning. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 8990-8999). https://doi.org/10.1109/CVPR.2018.00937

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.

Taylor, M., Guiver, J., Robertson, S., & Minka, T. (2008). SoftRank: Optimizing non-smooth rank metrics. In *Proceedings of the 1st WSDM* (pp. 77-86).

Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., & Cheng, X. (2016). Match- SRNN: Modeling the recursive matching structure with spatial RNN. In *Proceedings of IJCAI conference* (pp. 2922-2928).

Wang, B., Yang, Y., Xu, X., Hanjalic, A., & Shen, H.T. (2017). Adversarial cross-modal retrieval. In *Proceedings of International Conference on Multimedia* (pp. 154–162).

Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y.,Wang, B., & Zhang, D. (2017). IRGAN: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th SIGIR* (pp. 515–524).

Wang, X., Li, C., Golbandi, N., Bendersky, M., & Najork, M. (2018). The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th CIKM* (pp. 1313-1322).

Wang, X., Shrivastava, A., & Gupta, A. (2017). A-Fast-RCNN: Hard positive generation via adversary for object detection. In *Proceedings-30th IEEE conference on computer vision and pattern recognition, CVPR 2017*, pp. 3039–3048. https://doi.org/10.1109/CVPR.2017.324

Wang, Z., Xu, Q., Ma, K., Jiang, Y., Cao, X., & Huang, Q. (2019). Adversarial preference learning with pairwise comparisons. In *Proceedings of international conference on multimedia* (pp. 656–664).

Wei, Z., Xu, J., Lan, Y., Guo, J., & Cheng, X. (2017). Reinforcement learning to rank with markov decision process. In *Proceedings of the 40th SIGIR* (pp. 945–948).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3–4), 229–256.
Wu, Q., Burges, C. J., Svore, K. M., & Gao, J. (2010). Adapting boosting for information retrieval measures. *Journal of Information Retrieval, 13*(3), 254–270.
Xia, F., Liu, T.-Y., Wang, J., Zhang, W., & Li, H. (2008). Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th ICML* (pp. 1192-1199).
Xu, D., & Wang, Z. (2021). Semi-supervised semantic segmentation using an improved generative adversarial network. *Journal of Intelligent and Fuzzy Systems, 40*(5), 9709–9719. https://doi.org/10.3233/JIFS-202220
Xu, J., & Li, H. (2007). AdaRank: a boosting algorithm for information retrieval. In *Proceedings of the 30th SIGIR* (pp. 391–398).
Xu, J., Wei, Z., Xia, L., Lan, Y., Yin, D., Cheng, X., & Wen, J.-R. (2020). Reinforcement learning to rank with pairwise policy gradient. In *Proceedings of SIGIR* (pp. 509–518).
Xu, P., Gao, F., & Gu, Q. (2020). An improved convergence analysis of stochastic variance-reduced policy gradient. In *Proceedings of the 35th UAI conference*.
Xu, P., Gao, F., & Gu, Q. (2020). Sample efficient policy gradient methods with recursive variance reduction. In *Proceedings of ICLR*.
Yao, J., Dou, Z., Xu, J., & Wen, J.-R. (2020). RLPer: A reinforcement learning model for personalized search. In *Proceedings of the web conference* (pp. 2298-2308).
Yilmaz, Z.A., Wang, S., Yang, W., Zhang, H., & Lin, J. (2019). Applying BERT to document retrieval with birch. In *Proceedings of EMNLP 2019* (pp. 19-24).
Yu, H.-T., Huang, D., Ren, F., & Li, L. (2022). Diagnostic evaluation of policy-gradient-based ranking. *Electronics, 11*(1), 37
Yu, H.-T., Jatowt, A., Joho, H., Jose, J., Yang, X., & Chen, L. (2019). Wass-Rank: Listwise document ranking using optimal transport theory. In *Proceedings of the 12th WSDM* (pp. 24–32).
Yuan, F., Guo, G., Jose, J., Chen, L., Yu, H.-T., & Zhang, W. (2016). LambdaFM: Learning optimal ranking with factorization machines using lambda surrogates. In *Proceedings of the 25th CIKM* (pp. 227–236).
Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th SIGIR* (pp. 271–278).
Zeng, W., Xu, J., Lan, Y., Guo, J., & Cheng, X. (2018). Multi page search with reinforcement learning to rank. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (pp. 175–178).
Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *Proceedings of the 36th ICML* (Vol. 97, pp. 7354-7363).
Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., & Carin, L. (2017). Adversarial feature matching for text generation. In *ICML* (pp. 4006–4015). PMLR.
Zhao, J., Mathieu, M., & LeCun, Y. (2017). Energy-based generative adversarial network. In *International conference on learning representations (iclr)*.
Zou, S., Li, Z., Akbari, M.,Wang, J., & Zhang, P. (2019). MarlRank: Multi-agent reinforced learning to rank. In *Proceedings of CIKM* (pp. 2073–2076).

Authors and Affiliations

Hai-Tao Yu[1] · Rajesh Piryani[1] · Adam Jatowt[2] · Ryo Inagaki[1,4] · Hideo Joho[1] · Kyoung-Sook Kim[3]

✉ Hai-Tao Yu
yuhaitao@slis.tsukuba.ac.jp

Rajesh Piryani
rajesh.piryani@gmail.com

Adam Jatowt
adam.jatowt@uibk.ac.at

Ryo Inagaki
ryo.inagaki5@gmail.com

Hideo Joho
hideo@slis.tsukuba.ac.jp

Kyoung-Sook Kim
ks.kim@aist.go.jp

[1] Faculty of Library, Information and Media Science, University of Tsukuba, 1-2 Kasuga, Tsukuba, Ibaraki 305-8550, Japan

[2] Department of Computer Science & DiSC, University of Innsbruck, Innrain 52, Innsbruck 6020, Austria

[3] National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan

[4] Graduate School of Library, Information and Media Studies, University of Tsukuba, 1-2 Kasuga, Tsukuba, Ibaraki 305-8550, Japan