

Generative Adversarial Ranking Nets

Yinghua Yao*

EVA.YH.YAO@GMAIL.COM

*Center for Frontier AI Research, A*STAR, Singapore
and*

*Institute of High Performance Computing, A*STAR, Singapore*

Yuangang Pan†

YUANGANG.PAN@GMAIL.COM

*Center for Frontier AI Research, A*STAR, Singapore
and*

*Institute of High Performance Computing, A*STAR, Singapore*

Jing Li

J.LEE9383@GMAIL.COM

*Center for Frontier AI Research, A*STAR, Singapore
and*

*Institute of High Performance Computing, A*STAR, Singapore*

Ivor W. Tsang

IVOR.TSANG@GMAIL.COM

*Center for Frontier AI Research, A*STAR, Singapore
and*

*Institute of High Performance Computing, A*STAR, Singapore*

Xin Yao

XINYAO@LN.EDU.HK

Department of Computing and Decision Sciences, Lingnan University, Hong Kong

Editor: Matthew Hoffman

Abstract

We propose a new adversarial training framework – generative adversarial ranking networks (GARNet) to learn from user preferences among a list of samples so as to generate data meeting user-specific criteria. Verbosely, GARNet consists of two modules: a ranker and a generator. The generator fools the ranker to raise generated samples to the top; while the ranker learns to rank generated samples at the bottom. Meanwhile, the ranker learns to rank samples regarding the interested property by training with preferences collected on real samples. The adversarial ranking game between the ranker and the generator enables an alignment between the generated data distribution and the user-preferred data distribution with theoretical guarantees and empirical verification. Specifically, we first prove that when training with full preferences on a discrete property, the learned distribution of GARNet rigorously coincides with the distribution specified by the given score vector based on user preferences. The theoretical results are then extended to partial preferences on a discrete property and further generalized to preferences on a continuous

*. The first version of this work was done when the author was a PhD student at Southern University of Science and Technology (SUSTech), China and University of Technology Sydney (UTS), Australia.

†. The corresponding author.

property. Meanwhile, numerous experiments show that GARNet can retrieve the distribution of user-desired data based on full/partial preferences in terms of various interested properties (i.e., discrete/continuous property, single/multiple properties). Code is available at <https://github.com/EvaFlower/GARNet>.

Keywords: Generative Adversarial Network, Controllable Generation, User Preferences, Adversarial Ranking, Relativistic f -Divergence

1. Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Arjovsky et al., 2017; Huang et al., 2022) are a popular generative model to approximate a data distribution via adversarial training between two networks, wherein a generator learns to generate data alike real data, whereas a discriminator learns to distinguish real data from synthetic data coming from the generator. Although unconditional generative modeling presents an intriguing technical challenge, it lacks the ability to control the generation of data and has limited practical relevance. In contrast, controllable generative models can generate data to match some user-desired properties, meeting the needs of many real-world applications (Engel et al., 2018). Recent success has been driven in the field of using some predefined attributes/classes or an evaluator that can evaluate specific properties for samples to direct the generation process of GANs (Mirza and Osindero, 2014; Asokan and Seelamantula, 2020; Engel et al., 2018; Gupta and Zou, 2019; Samanta et al., 2020).

However, in many real-world applications, users may not be able to explicitly articulate the data they are interested in via the desired class/attribute labels or high values from the property evaluator (Engel et al., 2018). Instead, more naturally, users would often implicitly describe their preferences for data by ranking two or more samples in terms of a certain interested property (Fürnkranz and Hüllermeier, 2017). In particular, GANs are supposed to generate *more (or even exclusively) user-desired data (i.e., data for which users have high preferences)* by learning from preferences, which can have many promising applications. Here are some examples we propose. (1) In terms of image retrieval, it is more precise to adopt visual comparisons than conventional textual feedback to match the user’s mental model of the desired content (Yu and Kovashka, 2020). Thus, data generated based on preferences is a good representation of user needs, which can be used to retrieve user-desired images from the database. (2) In terms of drug design, the experts may not be able to clearly describe the structure of desired drugs for target diseases but can assign their preferences over candidate drug structures based on their expertise or lab experiments. The process of drug discovery can be formulated as learning user-desired drug data distribution with the preferences. (3) In terms of reinforcement learning (RL) without an off-the-shelf reward function, it is feasible to provide human preferences between pairs of trajectories of the agent (Christiano et al., 2017; Wirth et al., 2017). Then, the generative model can produce user-desired demonstrations by learning from the preferences. The RL tasks can be solved as imitation learning with these generated demonstrations.

There were a few works (Yu and Kovashka, 2020; Yao et al., 2022) applying user preferences to direct the generation of GANs. However, they required heavy interactions with humans to obtain preferences for generated samples during training so as to generate user-desired data. Moreover, these studies only considered the simplest pairwise preferences, failing to handle more general partial preferences. In this study, we consider teaching

GANs to learn a user-preferred data distribution directly from a more general form of user preferences, i.e., a collection of partial preferences over the training samples, without extra annotations on generated samples during the training. Incorporating such preferences into the training of GANs poses several challenges. First, learning a user-preferred data distribution means that the distribution alignment is imposed between a portion of the training data and generated data, while GANs match the generated data distribution to the whole training data distribution (Goodfellow et al., 2014; Mao et al., 2017). Therefore, GANs’ original framework cannot achieve the goal of this study. Second, the discriminator in GANs is designed to distill label information on individual samples (Goodfellow et al., 2014; Mirza and Osindero, 2014; Odena et al., 2017). So it cannot deal with preference data defined upon a list of samples, which is usually formulated as listwise ranking problems (Lin, 2010). Third, the key to eliciting adversarial classification in GANs is to assign the generated samples with “fake” labels when training the discriminator and “real” labels when training the generator, respectively (Goodfellow et al., 2014; Dai et al., 2017). Nevertheless, constructing the supervision for the generated samples adversarially in the context of users’ preferences (a.k.a, listwise ranking (Cao et al., 2007; Pan et al., 2022)) has not been explored.

To tackle the challenges, we propose a new distribution matching mechanism between generated data and high-ranked data while considering all data in the training. Meanwhile, in order to learn from user preferences, GANs need to handle a list of samples. Motivated by learning to rank (Cao et al., 2007), we tailor-design a ranker to replace the discriminator in GANs. Therefore, we reform the adversarial learning framework of GAN and propose Generative Adversarial Ranking Nets (GARNet) to learn from user preferences. Specifically, our GARNet consists of a ranker and a generator. While a vanilla discriminator performs a binary classification task by taking an individual sample as input, our proposed ranker learns to rank samples in the context of partial ranking lists regarding the interested property. In addition, the ranker assigns the generated samples to the lowest position. As the generator targets the user-preferred distribution, it is trained to synthesize samples that confuse the ranker to be ranked higher than the real samples. Competition between the ranker and the generator drives both two modules to improve themselves until the generator converges to a user-preferred data distribution. The main contributions are summarized as follows:

- We propose Generative Adversarial Ranking Nets (GARNet) for data generation guided by user preferences. In particular, we design an adversarial ranking game between a ranker and a generator, which enables a distribution alignment between generated data and a portion of the training data (i.e., user-desired data) (Corollary 7).
- We prove that the objective function of the ranker in GARNet defines a relativistic f -divergence between the user-preferred data distribution and the generated data distribution, while the generator is to minimize the divergence to learn the user-preferred data distribution. Meanwhile, we prove that the distribution learned by GARNet is rigorously determined by the score vector given by the users (Proposition 5).
- We empirically show our GARNet can learn a user-preferred data distribution (determined by the given score vector) from preferences in terms of discrete and continuous properties. We further show that our GARNet can better retrieve the distribution of user-desired data from partial preferences than various baselines. The potential of GARNet is also validated in the application of imbalanced learning.

2. Background and Problem Statement

In this section, we first review existing work that applies labels or property evaluators to control generation with desired properties. Then we formulate our problem setting – controllable generation by preferences.

2.1 Property-oriented Generation by Labels/Evaluators

Many works (Mirza and Osindero, 2014; Odena et al., 2017; Miyato and Koyama, 2018; Sohn et al., 2015; Li et al., 2020; Pumarola et al., 2020; Ho and Salimans, 2022; Ding et al., 2020; Wang et al., 2021; Triess et al., 2022) proposed conditional variants of generative models that introduce conditioning variables \mathcal{C} to enable user control for the generation, i.e., modeling the conditional distribution $p(x|\mathcal{C})$. However, the condition \mathcal{C} involving discrete class labels and attribute labels (Zhu et al., 2017) makes a restricting user control (discussed in Section 2.2.1). The conditions involving continuous labels can have a fine-grained control, but they require complete knowledge about the properties, implying that there exists a universal evaluation criterion to annotate data with absolute values in terms of the properties, which could be hard to access in some real-world applications (Christiano et al., 2017).

A few works (Engel et al., 2018; Gupta and Zou, 2019; Samanta et al., 2020) proposed to apply an accessible property evaluator $h(x)$, which can provide evaluations for data samples w.r.t. a specified property to control the generation of generative models. However, these evaluators may not exist in real-world applications (Christiano et al., 2017) since they also require complete knowledge about the properties. In addition, their optimization for desired properties requires extensive feedback loops where all generated samples are sent to the evaluator $h(x)$ for evaluation (Gupta and Zou, 2019), which would incur heavy cost especially when the evaluation process is expensive. Further, an additional problem is that they need to carefully balance the goal of generation quality and that of producing data with desired properties (Engel et al., 2018; Samanta et al., 2020).

2.2 Our Problem: Property-oriented Generation by Preferences

Let $\mathcal{X} = \{x_n\}_{n=1}^N$ denote a training dataset with N samples and $\mathcal{S} = \{s_m\}_{m=1}^M$ denote a collection¹ of M preferences defined on subsets of \mathcal{X} . In particular, each preference $s \in \mathcal{S}$ is an ordered list, namely,

$$s := s^1 > s^2 > \dots > s^l, \text{ and } s^i \in \mathcal{X}, \forall i = 1, 2, \dots, l, \quad (1)$$

where $2 \leq l \ll N$ denotes the number of samples contained in s . l is usually varied for different preferences. Note that with $l \ll N$, annotating preferences simply relies on partial knowledge about the properties (Fürnkranz and Hüllermeier, 2017) because it only involves comparisons among small subsets of the dataset. Our target is to learn a generative model $P_g(x)$ that is equal to the user-preferred data distribution $P_u(x)$ from partial preferences \mathcal{S} .

1. Regarding data collection, the preferences \mathcal{S} are collected among multiple users following the way in rank aggregation (Lin, 2010; Saquil et al., 2018; Pan et al., 2018).

Table 1: Main mathematical notations in this paper. Elements in a set (e.g., \mathcal{X}) are indexed using subscripts (e.g., x_i), whereas objects ranked at the i -th position in a ranking list s are indexed using superscripts s^i .

Notation	Explanation
\mathcal{X}	training samples $\mathcal{X} = \{x_n\}_{n=1}^N$
\mathcal{Y}	class labels $\mathcal{Y} = \{y_1, y_2, \dots, y_T\}$ endowed with an order $y_1 > y_2 > \dots > y_T$
\mathcal{O}	score space for a continuous property
$y(x)$	function that maps sample x to its class label
$o(x)$	function that maps sample x to its underlying ground-truth ranking score
$x_i > x_j$	sample x_i is preferred over sample x_j , $x_i, x_j \in \mathcal{X}$
s	preferences, $s := s^1 > s^2 > \dots > s^l$, $s^i \in \mathcal{X}$
\mathcal{S}	a collection of preferences $\mathcal{S} = \{s_m\}_{m=1}^M$ over training data \mathcal{X}
$\pi(s)$	ground-truth score vector for preference s
$s^{(R)}$	target preference to train the ranker, i.e., $s^{(R)} := s^1 > \dots > s^l > x_g$
$s^{(G)}$	target preference to train the generator, i.e., $s^{(G)} := x_g > s^1 > \dots > s^l$
P_u	user-preferred data distribution
P_g	distribution of the generative model
P_i	distribution of $\mathcal{X}_i = \{x y(x) = y_i\}$
D_f	relativistic f -divergence
R	ranker
G	generator

To be specific, $P_u(x)$ should allocate high density to global high-ranked data while low or even zero density to global low-ranked data².

For example, as shown in Fig. 1, we assume that shoe images with different strengths in terms of the *open* attribute are distributed evenly. The collected pairwise preferences reveal that users prefer more *open* shoes. Accordingly, the generative model is expected to learn a distribution that assigns a larger density on shoe images with larger *open* values. Note that the generative model has the advantage of generating novel images different from the training data (Song and Ermon, 2019).

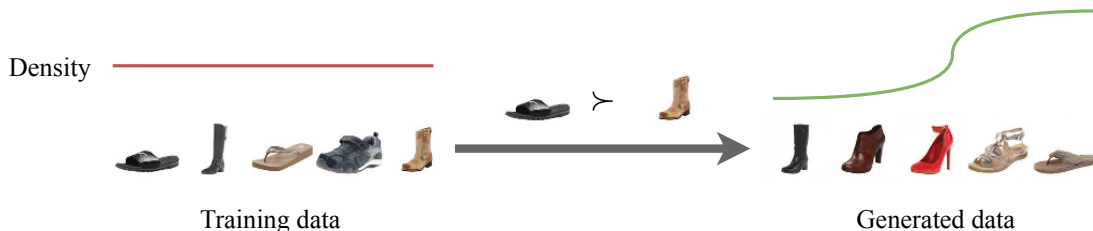


Figure 1: Illustration of user-preferred data distribution learning guided by user preferences (on the *open* attribute).

2. User preferences help to derive a global ranking over all the data (Cao et al., 2007; Lu and Boutilier, 2011). Global high-ranked data denote those data ranked high in the global ranking. Global low-ranked data denote those data ranked low in the global ranking.

2.2.1 CONTROLLED BY PREFERENCES VS. CONDITIONED ON LABELS

Class labels or attribute labels are argued to have limited description capacity (Parikh and Grauman, 2011) since such category-level labels cannot capture intra-category differences. In contrast, user preferences can be used to describe fine-grained information (Parikh and Grauman, 2011; Chen et al., 2013; Raman and Joachims, 2014). For example, people feel tangled when they are asked to decide whether a face image is smiling or not, as there always exist many images that are difficult to categorize, apart from a small amount of obvious smiling images and non-smiling images. Instead, people can compare a set of images and rank them in terms of the strength of the *smile* attribute.

On the other hand, since generation conditioned on class labels is dominated by those classes with sufficient training observations, a class with limited samples would be overlooked by the generative model. Instead, generation controlled by preferences can stress the modeling for “preferred class”. We will empirically show that preferences-guided distributional generation can serve as a remedy to improve extremely imbalanced class learning, while generation conditioned on labels fails in this case (See Section 7.4).

3. GARNet for Preferences w.r.t. a Discrete Property

First of all, we consider a scenario in which the ranking relation among training samples is based on a discrete property. In particular, each training sample belongs to one of a finite set of classes which possess a natural order.

Definition 1 (Preferences w.r.t. a discrete property) *Let $\mathcal{X} = \{x_n\}_{n=1}^N$ be a set of training instances and $\mathcal{Y} = \{y_1, y_2, \dots, y_T\}$ be a set of class labels endowed with an order $y_1 > y_2 > \dots > y_T$, where T is the number of ordered classes. Assume each training instance x_i is assigned with a label $y(x_i) \in \mathcal{Y}$, where $y(\cdot)$ is a function that outputs the label for a sample. The preference over a group of samples $\{x_{i_1}, x_{i_2}, \dots\} \subseteq \mathcal{X}$ is defined as*

$$\begin{aligned} & x_{i_1} > x_{i_2} > x_{i_3} > \dots, \\ \text{if } & y(x_{i_1}) > y(x_{i_2}) > y(x_{i_3}) > \dots \end{aligned} \tag{2}$$

We ignore the preference lists containing samples with the same labels since such preference lists can be simply transformed into the preferences (Eq. (2)) by removing redundant samples (Pan et al., 2022).

Example. Suppose users prefer small digits on the MNIST dataset. Then for digits 0 to 9, the order would be $0 > 1 > 2 > \dots > 9$, according to which users express their preferences over MNIST images.

In the following part of this section, we propose a framework called Generative Adversarial Ranking Nets (GARNet) for the distribution learning guided by the above-mentioned preferences (problem definition in Section 2.2). Our GARNet consists of a ranker R (parameterized by θ_R) and a generator G (parameterized by θ_G). In particular, an adversarial ranking process is defined between the generator and the ranker. Namely, they are trained against the goal of ranking defined between the real data and the generated data (See Fig. 2). The competition between the generator and the ranker drives the generated distribution $P_g(x)$ (simplified as P_g) aligned with the user-preferred data distribution $P_u(x)$ (simplified as P_u).

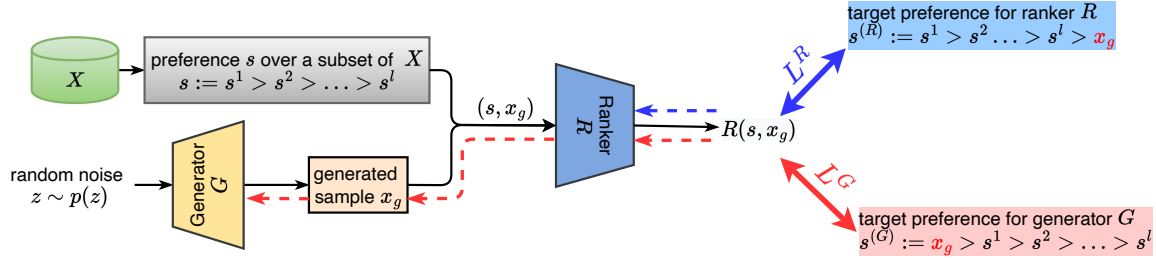


Figure 2: Framework of GARNet.

3.1 Given Full Preferences

We open the description of our GARNet with the case given a collection of full preferences $\mathcal{S} = \{s_m\}_{m=1}^M$. That is, the length of each preference l is equal to the number of ordered classes T . Namely

$$s := s^1 > s^2 \dots > s^l, l = T \text{ where } s^i \in \mathcal{X}, \forall i = 1, 2, \dots, l. \quad (3)$$

A full preference s would include samples from all ordered classes. Consequently,

$$y(s^i) \equiv y_i, \forall s \in \mathcal{S} \text{ and } \forall i = 1, 2, \dots, T. \quad (4)$$

Denoting P_i as the distribution of data $\mathcal{X}_i = \{x | y(x) = y_i, x \in \mathcal{X}\}$, we have $s^i \sim P_i$.

3.1.1 LISTWISE RANKING

The ranker R learns a ranking function from the preferences \mathcal{S} . We employ ListNet (Cao et al., 2007), which is a cross-entropy based Learning-to-Rank (L2R) function that maximizes:

$$\mathcal{L}_{L2R}(\pi(s), R(s)) = \sum_{i=1}^l \sigma(\pi(s))_i \log \sigma(R(s))_i. \quad (5)$$

With a slight abuse of notation, $s = (s^1, s^2, \dots, s^l)$. $\pi(s)$ denotes the ground-truth score vector for the preference s , which can be explicitly or implicitly given by humans (Cao et al., 2007). $R(\cdot)$ denotes the score vector for the preference s predicted by the ranker R . In specific, the ranker R acts as a nonlinear feature extractor with a scalar output. Given feature matrix $w = [w_1, w_2, \dots, w_l]$ of l samples, the ranker R outputs a score vector $R(w) = [R(w_1), R(w_2), \dots, R(w_l)]$, where $R(w_i) \in \mathbb{R}$. $\sigma(\cdot)$ is a softmax function that takes a list of scalar values $r = [r_1, r_2, \dots, r_l]$ as input, i.e., $\sigma(r)_i = \frac{e^{r_i}}{\sum_{j=1}^l e^{r_j}}$. $\sigma(\pi(s))_i$ calculates the top-1 probability of object s^i , i.e., the probability of s^i being ranked on the top given the scores of all the objects $\pi(s)$. Similarly, $\sigma(R(s))_i$ gives the top-1 probability of object s^i given the score vector $R(s)$.

3.1.2 GENERATIVE ADVERSARIAL RANKING

Given a preference s and a generated sample $x_g = G(z) \sim P_g(x)$, where $z \sim \mathcal{Z}$ is a random noise, the ranker R incorporates the generated sample x_g into the ranking list s and outputs

Algorithm 1 Generative Adversarial Ranking Nets

-
- 1: **Input:** Training data $\mathcal{X} = \{x_n\}_{n=1}^N$, user preferences $\mathcal{S} = \{s_m\}_{m=1}^M$, batch size B , score vector π , ranker R and generator G .
 - 2: **Output:** Generator G for user-preferred data distribution, i.e., $P_g(x) = P_u(x)$.
 - 3: **repeat**
 - 4: Sample a mini-batch of preferences $\{s_i\}_{i=1}^B$ from \mathcal{S} .
 - 5: Get fake samples $\{x_{gi}\}_{i=1}^B$ from the generator G , i.e., $x_{gi} = G(z_i)$ where z_i is a random noise.
 - 6: Following Eq. (6a), construct target preferences $\{s_i^{(R)}\}_{i=1}^B$ for the ranker R .
 - 7: Train the ranker R according to Eq. (8a).
 - 8: Following Eq. (6b), construct target preferences $\{s_i^{(G)}\}_{i=1}^B$ for the generator G .
 - 9: Train the generator G according to Eq. (8b).
 - 10: **until** convergence
-

a new ranking score vector as $R(s, x_g)$. Motivated by vanilla GAN (Goodfellow et al., 2014) which designs a target real/fake label for each generated sample to trigger the adversarial game between the discriminator and the generator, we construct a target preference for a generated list of samples consisting of s and x_g , i.e., (s, x_g) , as adversarial supervision. To promote the competition between the ranker R and the generator G , we design the target preference in two different ways, namely,

$$\text{target preference for } R \quad s^{(R)} := s^1 > \dots > s^T > \mathbf{x}_g, \quad (6a)$$

$$\text{target preference for } G \quad s^{(G)} := \mathbf{x}_g > s^1 > \dots > s^T. \quad (6b)$$

The generator fools the ranker to grade the generated samples as the best (Eq. (6b)); while the ranker learns to rank them at the lowest position (Eq. (6a)). Then, we define the objective for the ranker R and the generator G as follows, respectively,

$$\sup_{R: \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_{\substack{s \sim \mathcal{S} \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s^{(R)}), R(s^{(R)}) \right) \right], \quad (7a)$$

$$\sup_{G: \mathcal{Z} \rightarrow \mathcal{X}} \mathbb{E}_{\substack{s \sim \mathcal{S} \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s^{(G)}), R(s^{(G)}) \right) \right]. \quad (7b)$$

Thus, the training loss for the ranker and the generator with the mini-batch data can be formulated as follows, respectively:

$$L^R = \sum_{i=1}^B \mathcal{L}_{L2R} \left(\pi \left(s_i^{(R)} \right), R \left(s_i^{(R)} \right) \right), \quad (8a)$$

$$L^G = \sum_{i=1}^B \mathcal{L}_{L2R} \left(\pi \left(s_i^{(G)} \right), R \left(s_i^{(G)} \right) \right), \quad (8b)$$

where B is the batch size. The training algorithm is summarized in Algorithm 1.

GANs' theories usually show GANs estimate a divergence between the generated data distribution and the real data distribution, to support that they are good estimators of the

target data distribution. Thus similarly, we will show our GARNet is a good estimator of user-preferred data distribution by deriving a new divergence. In terms of most GAN variants, the objective function of the discriminator at optimum is a divergence (Jolicoeur-Martineau, 2020; Goodfellow et al., 2014; Arjovsky et al., 2017). In the following, we prove that training the ranker in GARNet (Eq. (7a)) is equivalent to estimating a divergence between the user-preferred data distribution $P_u(x)$ and the generated data distribution $P_g(x)$. The generator in GARNet (Eq. (7b)) is trained to minimize the divergence so as to achieve $P_g(x) = P_u(x)$.

Definition 2 (Divergence) *Let $P \in \mathbb{S}$ and $Q \in \mathbb{S}$ be probability distributions where \mathbb{S} is the set of all probability distributions with common support. A function $D : (\mathbb{S}, \mathbb{S}) \rightarrow [0, +\infty)$ is a divergence if it respects the following two conditions:*

$$\begin{aligned} D(P, Q) &\geq 0 \\ D(P, Q) &= 0 \iff P = Q. \end{aligned} \tag{9}$$

Theorem 3 (Relativistic f -divergence (Jolicoeur-Martineau, 2020)) *Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a concave function such that $f(0) = 0$, f is differentiable at 0, $f'(0) \neq 0$, $\sup_v f(v) = M > 0$, and $\arg \sup_v f(v) > 0$. Assume P and Q are two distributions with support \mathcal{X} , we have that*

$$D_f(P, Q) = \sup_{C: \mathcal{X} \rightarrow \mathbb{R}} 2 \mathbb{E}_{\substack{x \sim P \\ \tilde{x} \sim Q}} [f(C(x) - C(\tilde{x}))] \tag{10}$$

is a relativistic f -divergence.

Theorem 3 demonstrates the optimal discriminator in relativistic GAN (Jolicoeur-Martineau, 2019) estimates a relativistic f -divergence between the real data distribution and the generated data distribution. Its discriminator that is adapted from the concave function $f(z) = \log(\text{sigmoid}(z)) + \log(2)$ defines a pairwise ranking loss between a real sample and a fake sample, and we find that

$$\begin{aligned} f(C(x) - C(\tilde{x})) &= \log\left(\frac{1}{1 + e^{-(C(x) - C(\tilde{x}))}}\right) + \log(2) \\ &= \log\left(\frac{e^{C(x)}}{e^{C(x)} + e^{C(\tilde{x})}}\right) + \log(2) \\ &\stackrel{\textcircled{1}}{=} \mathcal{L}_{LR}(\pi(s), C(s)) + \log(2), \end{aligned} \tag{11}$$

$\textcircled{1}$ is valid if $\pi(s) = [0, -\infty]$ and $s := x > \tilde{x}$ where $x \sim P$ and $\tilde{x} \sim Q$.

Inspired by the connection between the relativistic f -divergence (Eq. (10)) and the tailored loss of our GARNet (Eq. (6a), Eq. (7a)), we argue that the optimal ranker of our GARNet also implicitly estimates a relativistic f -divergence, but between the user-preferred data distribution and the generated data distribution, which will be proven in Theorem 4. Accordingly, we introduce a new relativistic f -divergence between the user-preferred data distribution P_u and the generated data distribution P_g , which generalizes the targeted data distribution $P(x)$ in Theorem 3 to the mixture distribution $P_u = \sum_{i=1}^T q_i P_i$ with a user-specified mixing ratio where $q_1 > q_2 > \dots > q_T$ and $\sum_i q_i = 1$. In particular, P_i is the

distribution of $\mathcal{X}_i = \{x|y(x) = y_i, x \in \mathcal{X}\}$. The mixture distribution P_u allocates a larger density q_i to the higher-ranked class i , consistent with our problem setting in Section 2.2.

Theorem 4 (Relativistic f -divergence between P_u and P_g) *Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a concave function such that $f(0) = 0$, f is differentiable at 0, $f'(0) \neq 0$, $\sup_v f(v) = M > 0$, and $\arg \sup_v f(v) > 0$. Let P_u be the mixture distribution of the whole ordered data, i.e., $P_u = \sum_{i=1}^T q_i P_i$, where $q_1 > q_2 > \dots > q_T$ and $\sum_i q_i = 1$. Let P_g be the distribution of generated data x_g . We have that*

$$D_f(P_u, P_g) = \sup_{R: \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i R(s^i) - R(x_g) \right) \right], \quad (12)$$

is a relativistic f -divergence between P_u and P_g .

According to Definition 2, we prove that $D_f(P_u, P_g)$ is a divergence between P_u and P_g following the three steps:

#1 Prove that $D_f(P_u, P_g) \geq 0$.

#2 Prove that $P_u = \sum_{i=1}^T q_i P_i = P_g \implies D_f(P_u, P_g) = 0$.

#3 Prove that $D_f(P_u, P_g) = 0 \implies P_u = \sum_{i=1}^T q_i P_i = P_g$.

The details are left in Appendix A.

Instead of targeting general preferences that require $T + 1$ hyperparameters for the ground-truth score vector $\pi(s^{(R)})$ in Eq. (7a), we focus our problem setting on preferences where the first T terms of $\pi(s^{(R)})$ form an arithmetic progression, which depends only on the first term and the common difference, making it generally applicable to preferences of various lengths. Particularly, we demonstrate that when choosing the first T terms of $\pi(s^{(R)})$ as an arithmetic progression, the optimal ranker of our GARNet approximates a relativistic f -divergence between the user-preferred data distribution and the generated data distribution shown in Theorem 4.

Proposition 5 *Given the training dataset $\mathcal{X} = \{x_n\}_{n=1}^N$ and a collection of full preferences $\mathcal{S} = \{s_m\}_{m=1}^M$, where each preference $s := s^1 > s^2 > \dots > s^T \in \mathcal{S}$ and $s^i \sim P_i$. Let $P_u = \sum_{i=1}^T q_i P_i$ be the mixture distribution of the whole ordered data where $q_i = \sigma(\pi(s^{(R)}))_i$. Let P_g be the distribution of generated data x_g . Given a fixed generator G , the optimal ranker R^* of our GARNet (Eq. (7a)) approximates the relativistic f -divergence between P_u and P_g , i.e., $D_f(P_u, P_g)$, if $\pi(s^{(R)}) = [a + (T - 1)d, a + (T - 2)d, \dots, a, b]$ and $\frac{1}{e^{a-b}} \rightarrow 0$.*

Proof First of all, given the definition of $\pi(s^{(R)})$, we have

$$\lim_{\frac{1}{e^{a-b}} \rightarrow 0} q_{T+1} = \lim_{\frac{1}{e^{a-b}} \rightarrow 0} \sigma \left(\pi \left(s^{(R)} \right) \right)_{T+1} = \lim_{\frac{1}{e^{a-b}} \rightarrow 0} \frac{e^b}{e^b + \sum_{j=1}^T e^{a+(T-j)d}} \stackrel{\textcircled{1}}{=} 0, \quad (13a)$$

$$\lim_{\frac{1}{e^{a-b}} \rightarrow 0} \sum_{i=1}^T q_i = \lim_{\frac{1}{e^{a-b}} \rightarrow 0} 1 - q_{T+1} \stackrel{\textcircled{2}}{=} 1. \quad (13b)$$

where ①, ② follow the Squeeze theorem (Stewart et al., 2020) given $0 < q_{T+1} < \frac{1}{e^{a-b}}$ and $\frac{1}{e^{a-b}} \rightarrow 0$. Note in practice, we set $a - b = 10$ as $\frac{1}{e^{a-b}} \rightarrow 0$.

According to Eq. (5), we have

$$\begin{aligned}
 \mathcal{L}_{L2R} \left(\pi(s^{(R)}), R(s^{(R)}) \right) &= \sum_{i=1}^{T+1} \sigma(\pi(s^{(R)}))_i \log \sigma(R(s^{(R)}))_i & (14) \\
 &\stackrel{\textcircled{1}}{=} \left(\sum_{i=1}^T q_i \log \frac{e^{R(s^i)}}{e^{R(x_g)} + \sum_{j=1}^T e^{R(s^j)}} \right) + q_{T+1} \log \frac{e^{R(x_g)}}{e^{R(x_g)} + \sum_{j=1}^T e^{R(s^j)}} \\
 &= \log \frac{e^{\sum_{i=1}^T q_i R(s^i)}}{\left(e^{R(x_g)} + \sum_{j=1}^T e^{R(s^j)} \right)^{\sum_{i=1}^T q_i}} + q_{T+1} \log \frac{e^{R(x_g)}}{e^{R(x_g)} + \sum_{j=1}^T e^{R(s^j)}} \\
 &\stackrel{\textcircled{2}}{=} \log \frac{1}{e^{R(x_g) - \sum_{i=1}^T q_i R(s^i)} + \sum_{j=1}^T e^{R(s^j) - \sum_{i=1}^T q_i R(s^i)}}.
 \end{aligned}$$

① follows the definition of the target preference $s^{(R)}$ in Eq. (6a). ② is valid due to Eq. (13a) and Eq. (13b).

Meanwhile, when the ranker is approaching the optima R^* for a fixed generator G , i.e., $\sigma(R^*(s^{(R)})) = \sigma(\pi(s^{(R)}))$, we have $R^*(s^{(R)}) = \pi(s^{(R)}) + \delta$ due to the translation invariance of softmax (Laha et al., 2018), i.e., $\sigma(r + \delta) = \sigma(r)$. Therefore, the first T terms of $R^*(s^{(R)})$ is also arithmetic progression with a common difference of $-d$ same as that of $\pi(s^{(R)})$. Then, we have

$$\begin{aligned}
 R^*(s^j) - \sum_{i=1}^T q_i R^*(s^i) &= [R^*(s^1) - (j-1)d] - \sum_{i=1}^T q_i [R^*(s^1) - (i-1)d] & (15) \\
 &= R^*(s^1) - \left(\sum_{i=1}^T q_i \right) R^*(s^1) - \left(j-1 + \sum_{i=1}^T q_i(1-i) \right) d \stackrel{\textcircled{1}}{=} \left(\sum_{i=1}^T q_i i - j \right) d = c_j.
 \end{aligned}$$

where c_j is a constant $\forall j = 1, 2, \dots, T$, exclusively determined by the pre-specified score vector $\pi(s)$. ① is valid due to Eq. (13b).

Therefore, the objective for the ranker R can be approximated as follows:

$$\begin{aligned}
 \text{Eq. (7a)} &\stackrel{\textcircled{1}}{\approx} \sup_{R:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s\sim\mathcal{S} \\ x_g\sim P_g}} \left[\log \frac{1}{c + e^{-(\sum_{i=1}^T q_i R(s^i) - R(x_g))}} \right] \\
 &\stackrel{\textcircled{2}}{=} \sup_{R:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s^1\sim P_1 \\ \dots \\ s^T\sim P_T \\ x_g\sim P_g}} \left[\log \frac{1}{c + e^{-((\sum_{i=1}^T q_i R(s^i)) - R(x_g))}} \right] \\
 &\stackrel{\textcircled{3}}{=} \sup_{R:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s^1\sim P_1 \\ \dots \\ s^T\sim P_T \\ x_g\sim P_g}} \left[\log \frac{1}{c + e^{-((\sum_{i=1}^T q_i R(s^i)) - R(x_g))}} + \log(c + 1) \right] \\
 &\stackrel{\textcircled{4}}{=} \sup_{R:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s^1\sim P_1 \\ \dots \\ s^T\sim P_T \\ x_g\sim P_g}} \left[f \left(\sum_{i=1}^T q_i R(s^i) - R(x_g) \right) \right],
 \end{aligned} \tag{16}$$

① is reasonable because $\sum_{j=1}^T e^{R(s^j) - \sum_{i=1}^T q_i R(s^i)}$ in Eq. (14) would degenerate to a constant $c = e^{c_1 + c_2 + \dots + c_T}$ following Eq. (15), being independent of the ranker when the ranker is approaching the optima. ② is valid as $y(s^i) \equiv y_i$ according to Eq. (4). ③ is valid due to the addition of a constant. ④ is obtained by denoting $f(z) = \log \frac{1}{c + e^{-z}} + \log(c + 1)$.

Note f is a concave function; $f(0) = 0$; f is differentiable at 0; $f'(0) \neq 0$; $\sup_x f(x) = M > 0$; and $\arg \sup_x f(x) > 0$. According to Theorem 4, the optimal ranker of our GARNet approximately estimates a relativistic f divergence. ■

Remark 6 *In theory, it seems feasible to directly adopt the following objective to learn P_u according to Theorem 4.*

$$\sup_{R:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s^1\sim P_1 \\ \dots \\ s^T\sim P_T \\ x_g\sim P_g}} \left[f \left(\sum_{i=1}^T q_i R(s^i) - R(x_g) \right) \right], \tag{17a}$$

$$\sup_{G:\mathcal{X}\rightarrow\mathbb{R}} \mathbb{E}_{\substack{s^1\sim P_1 \\ \dots \\ s^T\sim P_T \\ x_g\sim P_g}} \left[f \left(R(x_g) - \sum_{i=1}^T q_i R(s^i) \right) \right], \tag{17b}$$

where the concave function f can be defined as $f(z) = \log(\text{sigmoid}(z)) + \log(2)$ like Jolicoeur-Martineau (2019). However, in practice, the above objective cannot provide sufficient gradient for the generator G to learn properly. Specifically, in the early training, the generator G is poor, and the ranker R can easily assign lower values to the generated sample x_g than real samples. $\sum_{i=1}^T q_i R(s^i) - R(x_g)$ would be large and then gradient vanish occurs.

Based on Proposition 5, we set a and b to be 10 and 0, respectively. The common difference d for the arithmetic progression in $\pi(s^{(R)})$ determines the mixture distribution $P_u = \sum_{i=1}^T q_i P_i$. Namely, the mixture distribution P_u would be close to the distribution of top-1 data when d is sufficiently large, which refers to our Corollary 7; while P_u would be close to a distribution where the whole ordered data is uniformly distributed when d is very small.

Corollary 7 *The optimal ranker of GARNet (Eq. (7a)) defines a relativistic f -divergence between the distribution of top-1 data P_1 and the generated data distribution P_g , i.e., $D_f(P_1, P_g)$ when $e^{-d} \rightarrow 0$.*

Proof According to the definition of q_1 in Proposition 5, we have

$$\begin{aligned} q_1 &= \frac{e^{a+(T-1)d}}{\left(\sum_{i=1}^T e^{a+(T-i)d}\right) + e^b} \\ &= \frac{e^{a+(T-1)d}}{e^{a+(T-1)d} (1 + e^{-d} + \dots + e^{-(T-1)d}) + e^{b-(a+T-1)d}} \\ &= \frac{1}{1 + e^{-d} + \dots + e^{-(T-1)d} + e^{b-(a+T-1)d}} > \frac{1}{1 + T e^{-d}} \end{aligned} \quad (18)$$

Since $\frac{1}{1+T e^{-d}} < q_1 < 1$, we have $q_1 \rightarrow 1$ following the Squeeze theorem (Stewart et al., 2020) when $e^{-d} \rightarrow 0$. The proof is completed. \blacksquare

Proposition 8 *Given the optimal ranker R^* as demonstrated in Proposition 5, the generator of GARNet (Eq. (7b)) is minimizing the divergence between P_u and P_g , i.e., $D_f(P_u, P_g)$.*

Proof Given R^* , the objective for the generator can be formulated as:

$$\sup_{G: \mathcal{Z} \rightarrow \mathcal{X}} \mathbb{E}_{\substack{s \sim \mathcal{S} \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s^{(G)}), R^*(s^{(G)}) \right) \right]. \quad (19)$$

We update the parameters of generator θ_G as follows:

$$\theta_G^{t+1} = \theta_G^t + \nabla_{\theta_G} \mathbb{E}_{\substack{s \sim \mathcal{S} \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s^{(G)}), R^*(s^{(G)}) \right) \right], \quad (20)$$

which maximizes the ranker's output for the generated samples $R^*(x_g)$. Then, $D_f(P_u, P_g)$ will be minimized. \blacksquare

Proposition 5 and Proposition 8 demonstrate that the distribution learned by our GARNet (Eq. (7)) is determined by the pre-specified score vector $\pi(s)$. We present a case study on MNIST (Lecun et al., 1998) to justify it by setting d to 0.1, 0.5, 1 and 5, respectively. As shown in Fig. 3, the proportion of generated data from GARNet is almost consistent with the top-1 probability calculated by the specified score vector, i.e., $q = \sigma(\pi(s))$.

In terms of preferences over a discrete property, the most desirable data is supposed to belong to the global top-1 class, i.e., $y(x) = y_1$. As clarified in Corollary 7, we can simply set up a sufficiently large d (e.g., 5) to achieve this goal.

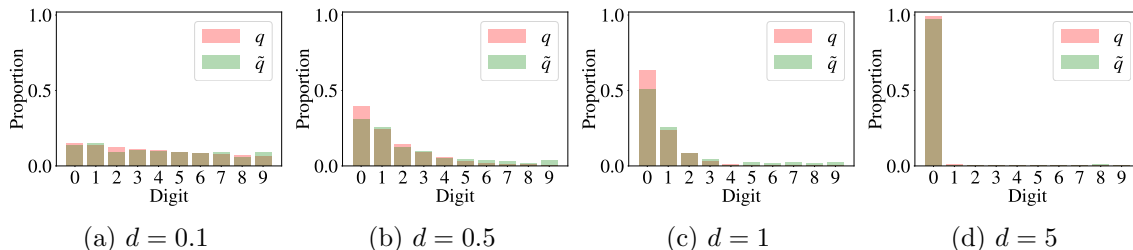


Figure 3: GARNet learns the distribution P_u with different pre-specified score vectors $\pi(s)$ from full preferences w.r.t. a discrete property on MNIST (preferring small digits, i.e., $0 \succ 1 \dots \succ 9$). d is the common difference for $\pi(s^{(R)})$ (Proposition 5). $q = \sigma(\pi(s))$ calculates the ground-truth top-1 probability of each digit class. \tilde{q} calculates the proportion of different digit classes for generated data from GARNet. The digit values for the generated data are evaluated by a pretrained classifier for digit classification.

3.2 Given Partial Preferences

In real-world situations, full preferences are not always available. A more realistic scenario is to provide partial preferences of various lengths as training dataset (Lin, 2010). Namely,

$$s := s^1 > s^2 > \dots > s^l, \quad l \leq T. \quad (21)$$

Similarly, we construct the adversarial ranking process between the ranker and generator:

$$\text{ranker} \quad s^{(R)} := s^1 > \dots > s^l > \mathbf{x}_g; \quad (22a)$$

$$\text{generator} \quad s^{(G)} := \mathbf{x}_g > s^1 > \dots > s^l. \quad (22b)$$

We adopt the objective (Eq. (7)) to learn the user-preferred distribution $P_u(x)$ via the adversarial ranking (Eq. (22)). Thanks to the consistency of the score function (Lemma 9), our GARNet can still learn the desired distribution from the partial preferences.

Lemma 9 (Consistency of the score function (Xia et al., 2008)) *Given a collection of partial preferences $\mathcal{S} = \{s_m\}_{m=1}^M$, where $2 \leq |s_m| \leq T$, the ListNet loss (Eq. (5)) adopted in our GARNet can recover the optimal score function for the ranker if \mathcal{S} contains sufficient preferences and the samples \mathcal{X} are well covered³.*

According to Lemma 9, the optimal ranker can recover $s^1 > s^2 > \dots > s^T > \mathbf{x}_g$. Therefore, GARNet can still learn a user-preferred data distribution from partial preferences, which assigns greater density to higher-ranked data. We present a case study on MNIST (Lecun et al., 1998) with partial preferences ($2 \leq l \leq 10$). We set the ground-truth score vector $\pi(s^{(R)})$ for Eq. (22a) as $[a + (l-1)d', a + (l-2)d', \dots, a, b]$ similar in Proposition 5. As shown in Fig. 4, the proportion of generated data from GARNet is larger when the digit class is ranked higher, which is consistent with user preferences. Especially, when d' is sufficiently large, GARNet only generates high-ranked data, namely, digit zero and one.

3. It doesn't mean to include every sample into the preferences but requires the samples included to be approximately uniformly distributed in the score space.

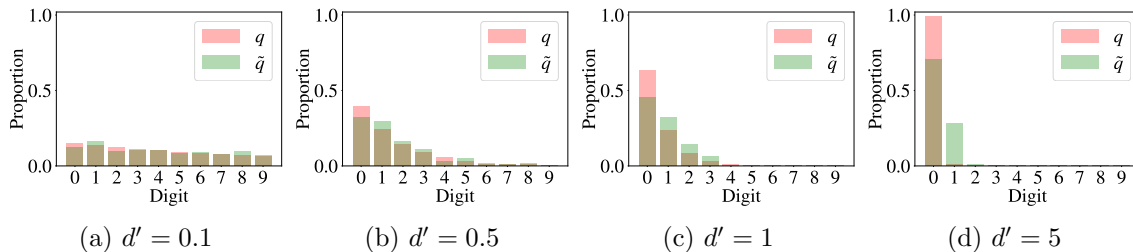


Figure 4: GARNet learns the distribution P_u with different pre-specified score vectors $\pi(s)$ from partial preferences ($2 \leq l \leq 10$) w.r.t. a discrete property on MNIST (preferring small digits). d' is the common difference of the ground-truth score vector $\pi(s^{(R)})$ for Eq. (22a). \tilde{q} calculates the proportion of different digit classes for generated data from GARNet. q is copied from the results in Fig. 3 for references. The digit values for the generated data are evaluated by a pretrained classifier for digit classification.

We found that training on relatively long preferences ($7 \leq l \leq 10$), GARNet can converge to top-1 data distribution when setting $d' = 5$. Note the resultant proportion per digit \tilde{q} is no longer consistent with pre-specified values q as $y(s^i) \equiv y_i, \forall s \in \mathcal{S}$ and $\forall i = 1, 2, \dots, T$ does not hold anymore for partial preferences.

4. GARNet for Preferences w.r.t. a Continuous Property

We consider a scenario that the ranking relation among training samples is based on a continuous property. Particularly, each training sample x_i is associated with an underlying score $o(x_i)$ that represents the user’s preference for x_i in terms of the property.

Definition 10 (Preferences w.r.t. a continuous property) *Given a training dataset $\mathcal{X} = \{x_n\}_{n=1}^N$, where each training instance $x_n \in \mathcal{X}$ has an underlying ground-truth ranking score $o(x_n) \in [A, B] \subseteq \mathcal{O}$ in terms of a certain continuous property, the preference among a list of samples $\{x_{i_1}, x_{i_2}, \dots\} \subseteq \mathcal{X}$ is defined as*

$$\begin{aligned} & x_{i_1} > x_{i_2} > x_{i_3} > \dots, \\ \text{if } & o(x_{i_1}) > o(x_{i_2}) > o(x_{i_3}) > \dots \end{aligned} \quad (23)$$

Example. Suppose users prefer smiling face images. Then users would assign higher preferences to those images with bigger smiles, i.e., the score for sample x in terms of the “smiling” attribute $o(x)$ being larger.

Theorem 11 *Given a collection of preferences in terms of a continuous property over $[A, B]$, see Eq. (23), it can be transformed equivalently to the preferences in terms of finite ordered classes if we consider the samples with close ranking scores as ties.*

Proof According to Heine-Borel theorem (Jeffreys et al., 1999), we have

$$O = \{(o_i - \epsilon_i, o_i + \epsilon_i) | o_i \in [A, B], \epsilon_i > 0, i = 1, 2, \dots, T'\}, \quad (24)$$

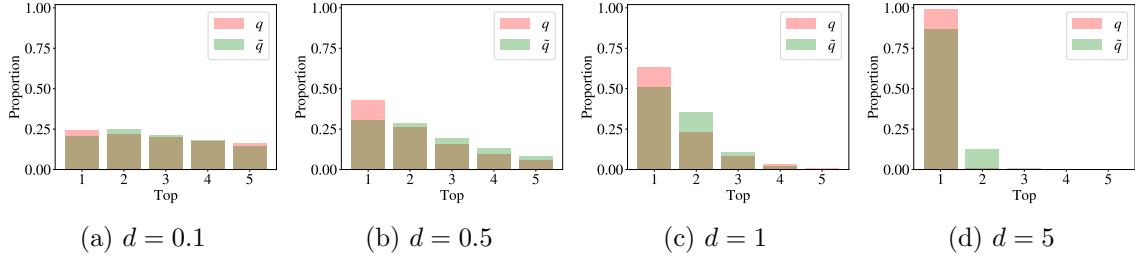


Figure 5: GARNet learns the distribution P_u with different pre-specified score vectors $\pi(s)$ from full preferences w.r.t. a continuous property on LFW (preferring smiling faces). $q = \sigma(\pi(s))$ calculates the top-1 probabilities of each class. \tilde{q} calculates the proportion of different classes for generated data from GARNet. Top 1 class represents images locating in the score interval of those real images ranked top 20% in terms of the *smile* attribute; top 2 class represents images locating in the score interval of those real images ranked between top 20% and top 40%; so on and so forth. The labels for the generated data are obtained based on the score given by a pretrained ranker for face images w.r.t. the *smile* attribute.

which is a finite subcover of $[A, B]$. Without loss of generality, we assume

$$o_1 > o_2 > \dots > o_{T'}.$$
 (25)

Then, we can construct a union of the non-overlapping intervals to cover the score interval $[A, B]$:

$$O = \left(\bigcup_{i=T', \dots, 2} [o_i - \epsilon_i, o_i + \epsilon_i] \right) \cup [o_1 - \epsilon_1, o_1 + \epsilon_1],$$
 (26)

where $o_{T'} - \epsilon_{T'} = A$, $o_1 + \epsilon_1 = B$, $o_i + \epsilon_i = o_{i-1} - \epsilon_{i-1}$, $i = T', \dots, 2$.

Meanwhile, we define a pair of samples within the same interval as a tie following Zhou et al. (2008), namely

$$x_m = x_n$$
 (27)

for any $o(x_m), o(x_n) \in O_i$,

where $O_i = [o_i - \epsilon_i, o_i + \epsilon_i]$, $i = T', T' - 1, \dots, 2$ and $O_1 = [o_1 - \epsilon_1, o_1 + \epsilon_1]$.

By assigning the same label y_i to all samples within each interval O_i , we can obtain a set of ordered class labels $\{y_i\}_{i=1}^{T'}$ where $y_1 > y_2 > \dots > y_{T'}$, namely

$$y(x_i) \equiv y_i$$
 (28)

if $o(x_i) \in O_i$.

To sum up, the preferences in terms of a continuous property (Definition 10) approximate the preferences in terms of T' ordered classes. ■

Remark 12 *The assumption of Eq. (27) is in line with real-world applications. People usually cannot distinguish a pair of samples when the difference is small and would annotate them as ties (Zhou et al., 2008), where ϵ_i is the indistinguishable score difference for each*

O_i . Nevertheless, for $o(x_1) \in O_{i-1}$ and $o(x_2) \in O_i$, x_1 and x_2 are indistinguishable when $o(x_1) - o(x_2)$ is small. Fortunately, assigning them as two adjacent ordered classes y_{i-1} and y_i is still reasonable because the order gap between y_{i-1} and y_i is not significant when T' is sufficiently large.

According to Theorem 11, learning from the preferences in terms of a continuous property can be discretized as learning from the preferences over a finite number of ordered classes. Therefore, we can adopt GARNet which defines an adversarial ranking goal (Eq. (6)) to learn the desired distribution from full preferences in terms of the continuous property. We present a case study on LFW (Huang et al., 2008) w.r.t. the *smile* attribute in Fig. 5 to justify this. Simply, we discrete the continuous ranking score w.r.t. the *smile* attribute over real images into five ordered classes. That is, images ranked top 20% in terms of the *smile* attribute are assigned as top 1 class; images ranked between top 20% and top 40% is assigned as top 2 class; so on and so forth. To label a generated sample, we first apply a pretrained ranker for ranking face images w.r.t. the *smile* attribute to predict the sample’s score and then determine the label as the corresponding class according to the score interval in which the score is located. As shown in Fig. 5, the proportion of generated data from GARNet is almost consistent with the top-1 probability calculated by the specified score vector, i.e., $q = \sigma(\pi(s))$.

Similarly, we can adopt GARNet (Eq. (22)) to learn the desired distribution from partial preferences in terms of the continuous property (see Section 7.2 for more empirical studies). In practice, we do not explicitly discretize the continuous ranking scores to a finite number of classes and define a so-called top-1 class as the most desirable data. Instead, we suppose the generative model is more consistent with user preferences if the generated data has higher ranking scores.

5. GARNet for a Mixture of Preferences

We extend our GARNet to a more general circumstance where a mixture of user preferences are collected from distinct groups of users. Particularly, we consider a simple situation where each group of users rank the samples in terms of a specific attribute, i.e., $\{\mathcal{S}_u\}_{u=1}^U$. The objective of GARNet in this context can be formulated as follows:

$$\sup_{R: \mathcal{X} \rightarrow \mathbb{R}} \sum_{u=1}^U \mathbb{E}_{\substack{s_u \sim \mathcal{S}_u \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s_u^{(R)}), R_u(s_u^{(R)}) \right) \right], \quad (29a)$$

$$\sup_{G: \mathcal{Z} \rightarrow \mathcal{X}} \sum_{u=1}^U \mathbb{E}_{\substack{s_u \sim \mathcal{S}_u \\ x_g \sim P_g}} \left[\mathcal{L}_{L2R} \left(\pi(s_u^{(G)}), R_u(s_u^{(G)}) \right) \right]. \quad (29b)$$

For simplicity, we extend the scalar output of the ranker R to a U dimension vector so as to learn from $\{\mathcal{S}_u\}_{u=1}^U$ simultaneously. R_u is the u -th output of the ranker.

Conditional GARNet When provided a mixture of preferences where one group of preferences might be conflicting with another, GARNet can achieve a compromise performance, which will be empirically verified in the experiment. However, for exactly opposing attributes, like being both “open” and “not open”, a practical way is to model them using

two separate GARNet models or extending our GARNet to its conditional version, denoted as CGARNet, conditioned on each group of preferences. The extension is similar as extending GAN to conditional GAN (Mirza and Osindero, 2014). Both the ranker and the generator can be conditioned on an attribute label.

6. Discussion on GANs Extended to Preference-guided Generation

In this section, we discuss how current GANs variants can be extended to preference-guided generation and their defects. To simplify the discussion, here we aim to learn the distribution that is most consistent with user preferences, i.e., the distribution of top-1 data P_1 .

Preferences guided generative adversarial learning To apply vanilla GAN (denoted as GAN-0) (Goodfellow et al., 2014) for this application, an intuitive way is to first construct a dataset consisting of the user-desired samples and then perform regular GAN training on this dataset to obtain a generator that purely outputs the desired samples. Two possible strategies can be applied:

(1) *Selecting desired samples using partial preferences directly* (GAN-1): A new dataset is constructed by selecting top-1 samples from each partial preference s_m . Since the top-1 samples in partial preferences are not necessarily the global top-ranked samples, this strategy produces biased training data that inevitably involves undesired samples. As a result, the derived distribution is not precisely user-desired. On the other hand, it would suffer from insufficient data issues when the amount of training data is small.

(2) *Selecting desired samples with a ranking proxy* (GAN-2): Specifically, we train a global ranking proxy using partial preferences (Cao et al., 2007) and then apply the proxy to select the global high-ranked samples as a new training dataset. For example, the top 30% of the training samples are selected as the desired dataset. In spite of its feasibility, this strategy may also incur insufficient data issues especially when the desired data in the original training dataset is limited.

Feedback GAN (FBGAN) (Gupta and Zou, 2019) is based on the second strategy and remedies its drawbacks by introducing the high-ranked generated samples into previous training data. Since the ratio of the high-ranked samples in the training data gradually increases along with the training epoch, all training data will ideally converge to the user-desired samples. However, FBGAN may suffer from severe data quality issues since plainly treating the generated samples as the training data will degrade the generation quality.

GANs plus a ranking module Another idea is to introduce a ranker as an extra critic (Saquil et al., 2018) to promote the generation of user-desired samples while the original GAN’s discriminator is kept as a critic to guarantee the generation quality, which is dubbed as GAN-RK.

Similarly, the ranker learns to rank samples from partial preferences and outputs high ranking scores for user-desired samples. The discriminator can be a classifier that distinguishes real from fake (Goodfellow et al., 2014) or define a distribution discrepancy (Arjovsky et al., 2017). For the generator, apart from the goal of aligning the generated distribution with the real distribution guided by the discriminator, an extra goal is in pursuit of generating samples with high-ranking scores judged by the ranker. However, both goals are conflicting since one requires the generator to synthesize samples alike the whole data

samples while the other requires the generation of partial data samples, i.e., samples with high-ranking scores. Thus, learning the desired data distribution cannot be well achieved.

Generator with Naive Ranker Another naive approach is to just have a ranker that learns from user preferences on real data to guide a generator, which is called GR. However, such an approach would suffer from poor data quality issues since there is no modeling for data authenticity in this model.

7. Experiments

We empirically verify that our GARNet can learn a data distribution that is best consistent with user preferences. Furthermore, we demonstrate the potential of GARNet in improving imbalanced class learning.

Dataset: (1) MNIST dataset (Lecun et al., 1998) consists of 28×28 images with digit zero to nine. We use its training set (50K images) for experiment. We suppose that users prefer smaller digits of MNIST. An image with a smaller digit value will be ranked higher. For instance, the partial ranking list over four digits 1, 3, 2, 9 is $s := 1 > 2 > 3 > 9$. As there are 10 digits in total, the maximal length of the preferences l can be 10. In particular, the length of the preferences included in the training data varies from 7 to 10 so that GARNet can converge to top-1 data distribution.

(2) Labeled Faces in the Wild (LFW) dataset (Huang et al., 2008) consists of 13,143 celebrity face images from the wild. LFW-10 consists of a subset of 2,000 images from LFW along with about 500 pairwise comparisons for one attribute. We take the pairwise comparisons in terms of the *smile* attribute as the user preferences for the training data. Since LFW10 has limited pairwise comparisons, we exploit a pretrained ranker to augment more preferences for the training data. In addition, as LFW10 has limited training images, i.e., 1K images, we take 13,143 images from LFW as our training data also. Specifically, we pretrain a ranker to learn to rank images in terms of the *smile* attribute from given pairwise comparisons of LFW10. Then we use the ranker to output ranking scores for all training samples of LFW and construct pairwise preferences based on the scores. If face image x_a has larger *smile* than face image x_b , then the preference is $s := x_a > x_b$.

(3) UT-Zap50K dataset (Yu and Grauman, 2014) contains 50,025 shoe images from Zappos.com. It contains pairwise comparisons over several attributes. We use all pairs, i.e., UT-Zap50K-1, UT-Zap50K-2, and UT-Zap50K-lexi as training data. The pairwise comparisons in terms of *comfort* (4,483 pairwise comparisons), *open* (4,351) and *sporty* (4,085) attributes are taken as the user preferences for the training data, respectively. We exploit a pretrained ranker to augment more preferences for the training data. We pretrain a ranker to rank images in terms of a specified attribute from given pairwise comparisons and then construct pairwise preferences based on the scores evaluated by the ranker. If shoe image x_a is more *comfort/open/sporty* than shoe image x_b , then the preference is $s := x_a > x_b$.

Baselines: We consider GAN (Goodfellow et al., 2014), FBGAN (Gupta and Zou, 2019), GAN-RK (Saquil et al., 2018) (GAN plus an additional ranker) and GR (a generator with a ranker that is only trained with partial preferences) as our baselines. In terms of GAN, it is trained with three kinds of subsets: the entire data, the subset made up of local high-ranked samples (top-1 samples in the partial preferences), and the subset made up of global high-ranked samples (e.g., top 10% for MNIST; top 50% for LFW and UT-Zap50K)

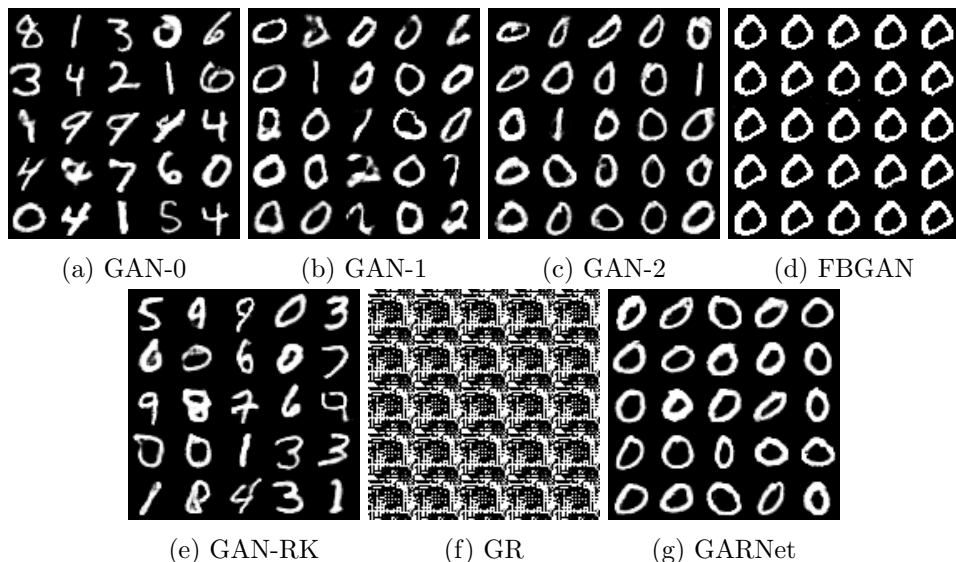


Figure 6: Comparison w.r.t. the generation of user-preferred digits (small digits) on MNIST.

selected by a surrogate ranker. These three variants are denoted as GAN-0, GAN-1, and GAN-2, respectively.

Network & Hyperparameters: All methods are implemented based on WGAN-GP architecture (DCGAN version) (Gulrajani et al., 2017) unless specifically mentioned. For the training we use the Adam optimizer (Kingma and Ba, 2015) with learning rate $2 * 10^{-4}$ and $\beta_1 = 0.5, \beta_2 = 0.999$. According to Proposition 5 and Corollary 7, we set the ground-truth score vector for $s^{(R)}$ as $\pi(s^{(R)}) = [10 + 5(l - 1), 10 + 5(l - 2), \dots, 10, 0]$ for all datasets, which can make GARNet learn a data distribution that is best consistent with user preferences as $q_1 \approx 1$. We simply set the ground-truth score vector for $s^{(G)}$ as $\pi(s^{(G)}) = [10 + 5(l - 2), 10 + 5(l - 3), \dots, 10, 5, 0]$. For MNIST, the batch size used is 50. The training iteration is set to 100K. For LFW and UT-Zap50K, the batch size is 64. The training iteration is 200K. The training images are resized to 32×32 unless specifically mentioned. To rank images in terms of a certain attribute, we adopt a pairwise ranking loss in Burges et al. (2005). The pretrained VGG16 (Simonyan and Zisserman, 2015) is used to map an image to its ranking score.

Evaluation Metrics: (1) For MNIST, mean digit (MD) is adopted as a performance measure for the generation of preferred digits. The digit values for the generated data will be evaluated by a pretrained classifier for digit classification. For each method, we randomly generate 50K digits and calculate their MD.

(2) For LFW and UT-Zap50K, mean score (MS) is adopted as a performance measure for the generation of user-preferred face images. The scores are evaluated by a pretrained ranker that learns to rank images in terms of the target attribute from the given preferences. Frechet Inception Distance (FID) (Heusel et al., 2017) measures the visual quality w.r.t. the high-ranked real face images. For each method, we randomly select 50K generated images and conduct the evaluation.

7.1 Preferences w.r.t. Discrete Digits

In this section, we apply GARNet to generate the user-preferred digits on MNIST using the partial preferences.

Table 2: Comparison of various methods on MNIST w.r.t. mean digit (MD, \downarrow). Best results are in bold. FBGAN suffers from mode collapse (Fig. 6d). GR generates meaningless results (Fig. 6f), so its MD is not collected.

Method	Real	GAN-0	GAN-1	GAN-2	GAN-RK	GR	FBGAN	GARNet
MD	4.45	4.49	0.75	0.15	4.63	-	0.00	0.00

According to Fig. 6 and Table 2, we highlight that only our GARNet successfully learns the distribution of the top-ranked digits from user preferences. Table 2 shows that except for GAN-0, GAN-RK, and GR, other methods have smaller MD compared to the training data (denoted as real). It means that those generative models can learn from user preferences to some extent. However, only FBGAN and GARNet can converge to generate top-ranked digits with MD 0.00. On the other hand, GAN-0 is the vanilla GAN that is trained with all training data, so its MD is close to that of real images. GAN-RK fails to learn from user preferences because of the conflict between the discriminator and ranker. GR cannot get any meaningful generations because it does not involve a distribution matching between real and generated samples. Therefore, its MD is not applicable.

Fig. 6 shows that: (1) GARNet generates top-ranked digits (digit 0), with high quality as shown in Fig. 6g. (2) According to Fig. 6b, GAN-1 still generates digits that are not ranked top, like digit two, since the constructed training subset contains undesired samples. (3) GAN-2 achieves better performance than GAN-1 but still fails to converge to the distribution of the top-ranked digit. (4) FBGAN suffers from mode collapse of which the generated digit zeros are exactly the same since its selection of high-ranked generated data does not consider maintaining sample diversity.

7.2 Preferences w.r.t. a Continuous Attribute

As described in experiments on MNIST, GAN-1, GAN-2, FBGAN and our GARNet can learn from the user preferences. Therefore, we only compare GARNet with these baselines on generating user-preferred facial images on LFW w.r.t. the *smile* attribute as well as generating user-preferred shoe images on UT-Zap50K w.r.t. the *comfort*, *open* and *sporty* attribute, respectively.

We quantitatively evaluate the performance of learning from user preferences by mean score (MS) and the generation quality by FID in Table 3. Meanwhile, we plot the score density for generated samples of various methods and training samples in Fig. 7. The evaluation results on LFW (*smile*) show that: GAN-1, GAN-2, and GARNet shift to a distribution of data with larger scores, which means they can learn from user preferences and generate face images with large *smiles*. (1) Our GARNet learns the best user-preferred distribution, indicated by a best MS, along with the best image quality, indicated by a lowest FID score. (2) GAN-1 and GAN-2 have poorer quality than GARNet. (3) FBGAN

suffers from mode collapse and its generated images have very poor quality (see Appendix), which is verified by its large FID score. Similar results can be concluded on UT-Zap50K.

For further visual quantification, we place GARNet’s generated samples and training samples (a.k.a. real samples) w.r.t. their scores on the *smile* axis in Fig. 8a. GARNet mainly (over 95%) covers those real images ranked top 50% in terms of the *smile* attribute. GARNet even generates images with higher scores than the maximum scores in the LFW dataset, which shows its potential for generating data beyond existing rankings. We place GARNet’s generated samples and training samples (a.k.a. real samples) w.r.t. their scores on the *comfort* axis in Fig. 8b. Over 75% generated images cover those real images ranked top 50% in terms of the *comfort* attribute. Results on the *open* and *sporty* attributes can be seen in Appendix.

Table 3: Comparison on LFW face data and UT-Zap50K shoe data w.r.t. performance measure (MS, \uparrow) and quality score (FID, \downarrow). The best results are highlighted in bold. Since FBGAN suffers from mode collapse and poor image quality issues (large FID; see its generated images in Appendix), its MS is not collected.

Dataset	MS					FID			
	Real	GAN-1	GAN-2	FBGAN	GARNet	GAN-1	GAN-2	FBGAN	GARNet
LFW (smile)	0.59	1.41	1.14	-	2.29	47.55	37.40	89.36	22.22
UT-Zap50K (comfort)	-5.06	-2.66	-0.91	-	-0.90	49.20	77.99	253.53	39.16
UT-Zap50K (open)	-0.36	0.93	0.99	-	3.75	63.92	112.93	265.49	46.78
UT-Zap50K (sporty)	4.09	5.28	4.48	-	7.07	41.02	84.59	431.44	32.87

7.3 User Control on a Mixture of Preferences

To demonstrate that GARNet can generalize to a mixture of user preferences, i.e., preferences on multiple attributes, we conduct the experiments in terms of the *comfort* and *open* attributes as the training data. To achieve a better generation, we apply a more advanced GAN architecture (Karras et al., 2020). Meanwhile, the training images are resized to 64×64 for better resolution.

In Fig. 9, we visualize real images and generated images by ordering them using the pretrained rankers defined over the *open* and *comfort* attributes, respectively. It is observed that GARNet generates shoe images that are more comfort or more open. In addition, the least *comfort* or the least *open* shoe images are not generated by GARNet as they are disliked by users.

It is worth mentioning that GARNet achieves a compromise between a pair of moderate conflicting attributes. Intuitively, *open* and *comfort* are conflicting attributes. For example, sport shoes are thought comfortable but may be slightly open. As shown in Figure 9, learning from preferences w.r.t. the *open* and *comfort* attributes, GARNet converges to avoid generating samples ranked lowest in either of two attributes but would generate relatively close and comfortable shoes or relatively uncomfortable and open shoes.

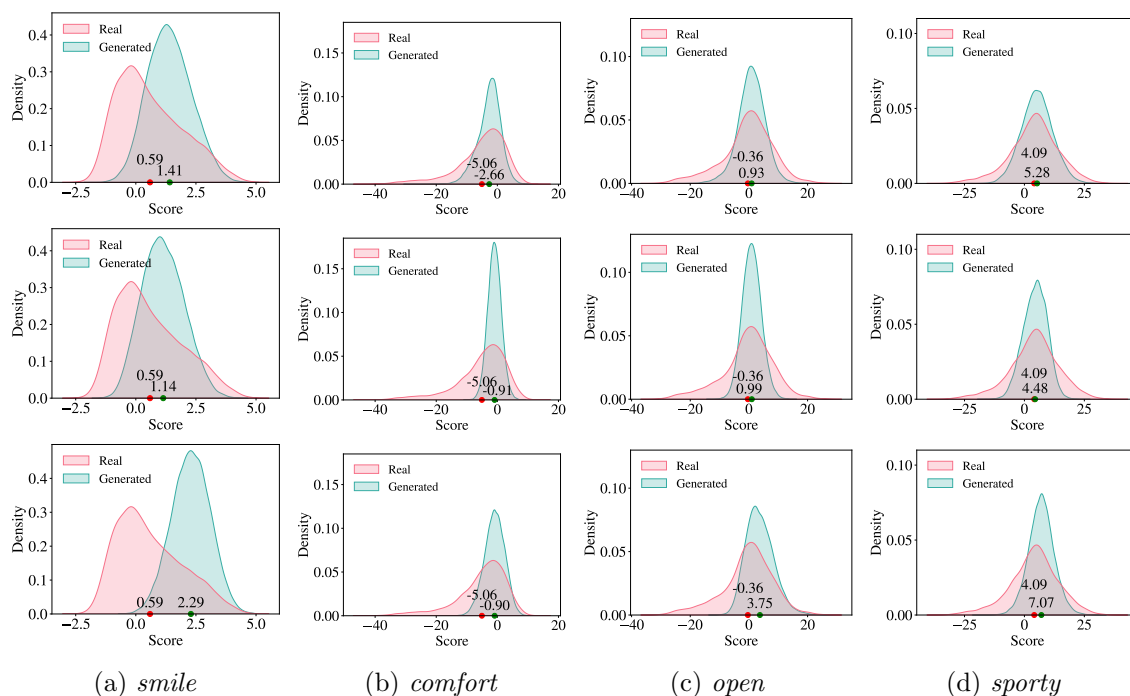


Figure 7: Comparison of GAN-1 (Row 1), GAN-2 (Row 2) and our GARNet (Row 3) in terms of the score density. The green point denotes the mean score of generated samples while the red point denotes that of real samples. Note the x-axis denotes ranking scores of samples instead of ordered classes in Figure 5. Similar observations that a preferred class has a larger proportion can be found if we coarsen the ranker scores into limited ordered classes.

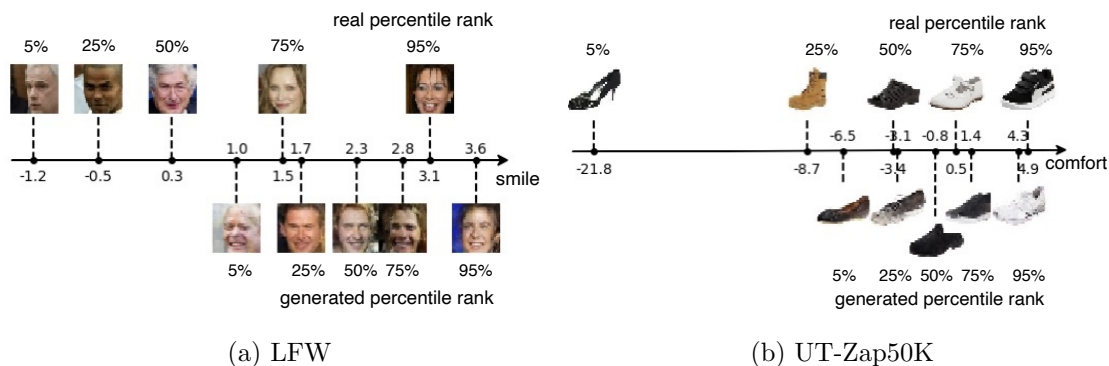


Figure 8: Real images (32×32 , above the axis) vs. generated images (GARNet, below the axis) w.r.t. the score axis of attributes. The percentile rank of a given score is the percentage of scores in its frequency distribution that are less than that score. Real (generated) percentile rank means calculated among real (generated) images.

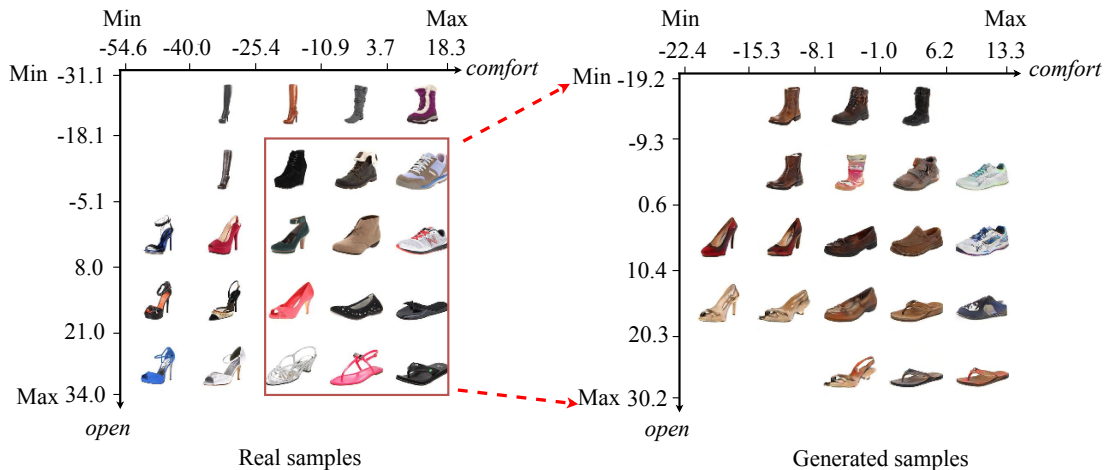


Figure 9: GARNet for multiple attributes. Images (64×64) are placed w.r.t. the “comfort” and “open” score. The results are obtained by an advanced GAN variant, StyleGAN2-ADA (Karras et al., 2020), thus having better quality.

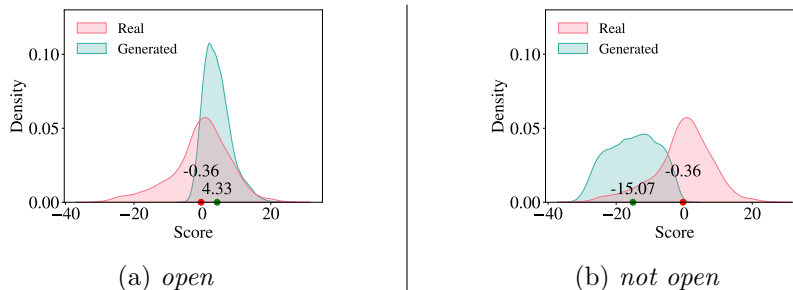


Figure 10: The generated samples (32×32) and the score density ($open \uparrow$, score \uparrow) for CGARNet conditioned on *open* and *not open* attributes, respectively. The green point denotes the mean score of generated samples while the red point denotes that of real samples.

7.3.1 CONDITIONAL GARNET

We apply our conditional GARNet (CGARNet) for exact opposing attributes, i.e., *open* and *not open*.

Fig. 10 shows that: (1) When conditioned on the *open* attribute, CGARNet generates shoe images with large *open* attribute values, with a score density that locates on a region of large *open* score values. (2) When conditioned on the *not open* attribute, CGARNet generates shoe images with small *open* attribute values, with a score density that locates on a region of small *open* score values.

7.4 GARNet on preferences VS. GANs on Labels

In this section, we show that preference-guided generation outperforms generation conditioned on labels in extremely imbalance class learning. Note conditional generation can generate samples for the minority data, showing a promising application for imbalanced

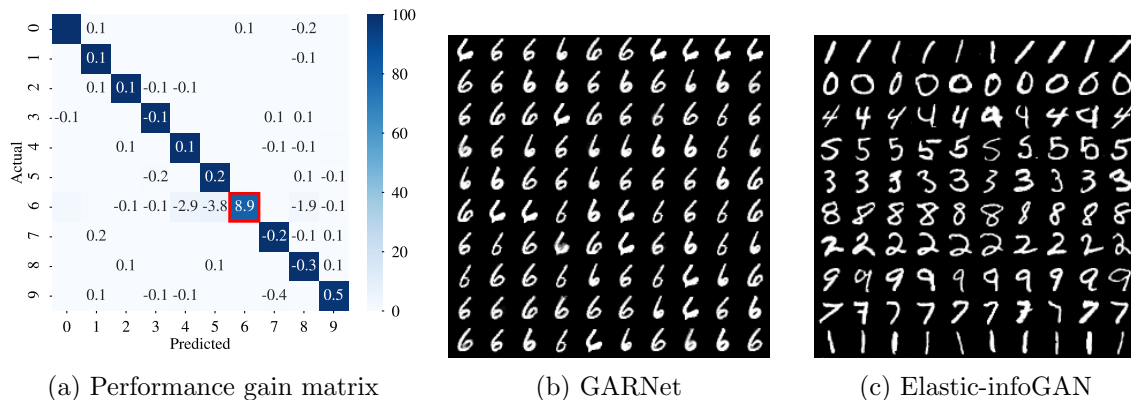


Figure 11: (a) GARNet boosts imbalance classification on MNIST with digit six as the minority class. The gain matrix (zero is not presented) is obtained by $C' - C$, where C' and C are the confusion matrix calculated on GARNet boosted data and original MNIST data, respectively. The color denotes the confusion matrix (%) on original MNIST data. (b-c) Visual results of GARNet and Elastic-infoGAN on MNIST with extremely imbalanced data.

data classification by promoting the minority via data augmentation in a pre-processing manner.

Experimental setup: On the MNIST dataset (Lecun et al., 1998), we randomly pick up 0.5% samples of the digit six to constitute the minority class. All samples of the rest classes are retained. Note the imbalance ratio reaches to minor/major $\approx \frac{1}{200}$. We construct partial preferences simply by preferring digit six than any other class. We compare the classification performance of a CNN classifier (CNN) without and with data augmentation by GARNet.

Baselines: Elastic-infoGAN (Ojha et al., 2020) is a state-of-art generative model for imbalanced data generation. Standard GAN trained only with minor class is excluded as a baseline since there are insufficient samples ($0.5\% * 4,951 \approx 25$) for training a generative model.

Fig. 11b shows that our GARNet successfully generates the user-desired data, i.e., digit six. Though there are limited digit six images in the training set, we can construct sufficient partial preferences between the minor class and the major classes. Therefore, our GARNet can learn a direction of user preferences and generate the desired data. In contrast, Fig. 11c shows that Elastic-infoGAN fails to generate digit six since it relies on class labels and the samples of digit six are too limited.

Therefore, our GARNet can be used to improve imbalance class learning via data augmentation but Elastic-infoGAN cannot. Fig. 11a shows the combined confusion matrix of two settings: with augmentation by GARNet and without augmentation. The color indicates the confusion matrix of CNN with imbalanced data, where every class has a high purity except digit six which is a bit confusing with digit four, five, and eight. By replenishing the digit six with GARNet, we train the same CNN architecture (CNN-GARNet). It shows that the purity of digit six increases by 8.9%, demonstrating GARNet can efficiently augment the minority data in imbalanced scenarios.

8. Conclusion

This paper presents a novel adversarial ranking framework, GARNet, to learn from user preferences. In particular, we prove that GARNet is equivalent to optimizing a divergence between the user-preferred data distribution (determined by the given score vector) and the generated data distribution, which theoretically guarantees GARNet as a good estimator of the desired data distribution. Meanwhile, we empirically show GARNet can obtain corresponding distributions when different given score vectors are specified. Numerous experiments demonstrate GARNet can learn the distribution that best matches user preferences compared to various baselines. A study of imbalanced class learning validates the advantage of preference-guided GARNet over GAN conditioned on labels.

GARNet is modeled under the homogeneity assumption of rank aggregation that only one ground truth full ranking exists (Lin, 2010), which is in line with the actual situation in many real-world applications – peer grading (Raman and Joachims, 2014), to image rating (Liang and Grauman, 2014), document recommendation (Sellamanickam et al., 2011), opinion analysis (Kim et al., 2015), and bioinformatics (Chatterjee et al., 2018). It is possible to break such an assumption when the preferences are sampled from heterogeneous user groups, i.e., a mixture of preferences: (1) There is one group of users dominating the ranking. Then, the homogeneity assumption is still valid by simply taking the samples from the majority of preferences as clean preferences and samples from the rest ones as noisy preferences. Robust rank aggregation (Pan et al., 2022) can be incorporated into our GARNet so as to handle these noisy preferences, which is left for future exploration. (2) The preferences from different groups of users are balanced. The homogeneity assumption on users is no longer valid and it is more reasonable and realistic to assume a heterogeneity of users for an idealized scenario (Zhao et al., 2016), which is called mixture rank aggregation. We conduct a preliminary exploration for a simple case where the group affiliation of each preference is known. We leave the general case of mixture ranking for future work.

Acknowledgements

This work was supported by the A*STAR Centre for Frontier AI Research, the A*STAR Career Development Fund (Grant No.C222812019), the A*STAR GAP project (Grant No.I23D1AG079), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), NSFC (Grant No.62250710682), and the Program for Guangdong Provincial Key Laboratory (Grant No. 2020B121201001).

Appendix A. Proof of Theorem 4

Proof Let $R^* = \arg \sup_{R: \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i R(s^i) - R(x_g) \right) \right]$ be the best choice of R .

#1 Prove that $D_f(P_u, P_g) \geq 0$.

Let $R^w(x) = c' \forall x \in \mathcal{X}$ (worst possible choice of R), where c' is a constant.

$$D_f(P_u, P_g) = \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f\left(\sum_{i=1}^T q_i R^*(s^i) - R^*(x_g)\right) \right] \geq \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f\left(\sum_{i=1}^T q_i R^w(s^i) - R^w(x_g)\right) \right] \stackrel{\textcircled{1}}{=} 0 \quad (30)$$

① is valid because $\sum_{i=1}^T q_i = 1$.

#2 Prove that $P_u = \sum_{i=1}^T q_i P_i = P_g \implies D_f(P_u, P_g) = 0$.

$$\begin{aligned} D_f(P_u, P_g) &= \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f\left(\sum_{i=1}^T q_i R^*(s^i) - R^*(x_g)\right) \right] \\ &\stackrel{\textcircled{1}}{\leq} \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T}} \left[f\left(\sum_{i=1}^T q_i R^*(s^i) - \mathbb{E}_{x_g \sim P_g} [R^*(x_g)]\right) \right] \\ &\stackrel{\textcircled{2}}{\leq} f\left(\mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T}} \left[\sum_{i=1}^T q_i R^*(s^i) \right] - \mathbb{E}_{x_g \sim P_g} [R^*(x_g)]\right) \\ &= f\left(\sum_{i=1}^T q_i \mathbb{E}_{s^i \sim P_i} [R^*(s^i)] - \mathbb{E}_{x_g \sim P_g} [R^*(x_g)]\right) \\ &\stackrel{\textcircled{3}}{=} f\left(\sum_{i=1}^T q_i \mathbb{E}_{x \sim P_i} [R^*(x)] - \mathbb{E}_{x \sim P_g} [R^*(x)]\right) \\ &= f\left(\int R^*(x) \sum_{i=1}^T q_i P_i dx - \int R^*(x) P_g dx\right) \stackrel{\textcircled{4}}{=} f(0) = 0, \end{aligned} \quad (31)$$

where ① and ② follow Jensen's inequality for a concave function f . ③ is valid due to change of variables. ④ is valid because $P_g = \sum_{i=1}^T q_i P_i$. Since $D_f(P_u, P_g) \geq 0$ (Eq. (30)), we have $D_f(P_u, P_g) = 0$.

#3 Prove that $D_f(P_u, P_g) = 0 \implies P_u = \sum_{i=1}^T q_i P_i = P_g$.

We prove this by contraposition. Namely, we prove that $P_u \neq P_g \implies D_f(P_u, P_g) \neq 0$. Let $\mathcal{H} = \{x | P_u(x) > P_g(x)\}$. Since $P_u \neq P_g$, we have $\mathcal{H} \neq \emptyset$. Let $u = \int_{\mathcal{H}} P_u(x) dx$ and $v = \int_{\mathcal{H}} P_g(x) dx$, we have $(1 - u) = \int_{\mathcal{X} \setminus \mathcal{H}} P_u(x) dx$ and $(1 - v) = \int_{\mathcal{X} \setminus \mathcal{H}} P_g(x) dx$. According to the definition, $u > 0, v > 0$, and $u > v$.

Let $R'(x) = \begin{cases} \Delta & \text{if } x \in \mathcal{H} \\ 0 & \text{else} \end{cases}$, where $\Delta \neq 0$, then we have

$$\begin{aligned}
 L(\Delta) &= \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i R'(s^i) - R'(x_g) \right) \right] \stackrel{\textcircled{1}}{=} \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i (R'(s^i) - R'(x_g)) \right) \right] \\
 &\stackrel{\textcircled{2}}{\geq} \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[\sum_{i=1}^T q_i f(R'(s^i) - R'(x_g)) \right] = \sum_{i=1}^T \mathbb{E}_{\substack{s^i \sim P_i \\ x_g \sim P_g}} [q_i f(R'(s^i) - R'(x_g))] \\
 &\stackrel{\textcircled{3}}{=} \sum_{i=1}^T \mathbb{E}_{\substack{x_u \sim P_i \\ x_g \sim P_g}} [q_i f(R'(x_u) - R'(x_g))] \\
 &= \int_{\mathcal{X}} \int_{\mathcal{X}} f(R'(x_u) - R'(x_g)) \left(\sum_{i=1}^T q_i P_i(x_u) \right) P_g(x_g) dx_u dx_g \tag{32} \\
 &\stackrel{\textcircled{4}}{=} \int_{\mathcal{X}} \int_{\mathcal{X}} f(R'(x_u) - R'(x_g)) P_u(x_u) P_g(x_g) dx_u dx_g \\
 &= \int_{\mathcal{H}} \int_{\mathcal{H}} f(R'(x_u) - R'(x_g)) P_u(x_u) P_g(x_g) dx_u dx_g \\
 &\quad + \int_{\mathcal{H}} \int_{\mathcal{X} \setminus \mathcal{H}} f(R'(x_u) - R'(x_g)) P_u(x_u) P_g(x_g) dx_u dx_g \\
 &\quad + \int_{\mathcal{X} \setminus \mathcal{H}} \int_{\mathcal{H}} f(R'(x_u) - R'(x_g)) P_u(x_u) P_g(x_g) dx_u dx_g \\
 &\quad + \int_{\mathcal{X} \setminus \mathcal{H}} \int_{\mathcal{X} \setminus \mathcal{H}} f(R'(x_u) - R'(x_g)) P_u(x_u) P_g(x_g) dx_u dx_g \\
 &= f(\Delta)u(1-v) + f(-\Delta)v(1-u),
 \end{aligned}$$

① is valid because $\sum_{i=1}^T q_i = 1$. ② follows Jensen's inequality for concave function f . ③ is valid due to the change of variables. ④ is obtained by denoting $P_u = \sum_{i=1}^T q_i P_i$.

Since $u(1-v) > v(1-u)$, $\exists \Delta^* > 0$, s.t. $L(\Delta^*) > 0$ according to Lemma A.3 in Jolicoeur-Martineau (2020). Thus, let $\Delta = \Delta^*$, we have $Eg. (32) > 0$. Therefore,

$$D_f(P_u, P_g) = \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i R^*(s^i) - R^*(x_g) \right) \right] \geq \mathbb{E}_{\substack{s^1 \sim P_1 \\ \dots \\ s^T \sim P_T \\ x_g \sim P_g}} \left[f \left(\sum_{i=1}^T q_i R'(s^i) - R'(x_g) \right) \right] > 0.$$

The proof is completed. ■

Appendix B. Additional Experimental Results

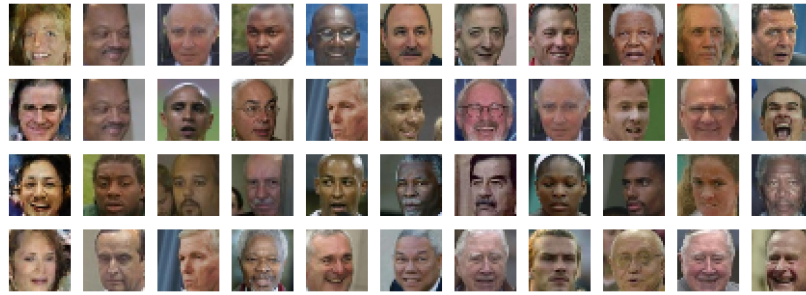
(a) *smile*(b) *comfort*(c) *open*(d) *sporty*

Figure 14: Verification of **GARNet generating novel images** (32×32) (generated images not exactly alike any training images) on LFW (*smile*) and UT-Zap50K (*comfort*, *open* and *sporty*). Nearest neighbors are measured by the ℓ_2 distance between images. The first column is generated samples from GARNet. The other columns are the generated samples' nearest neighbors in the training dataset.

References

- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.
- Siddarth Asokan and Chandra Sekhar Seelamantula. Teaching a GAN what not to learn. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems*, 2020.
- Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning*, 2005.
- Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning*, pages 129–136, 2007.
- Sujoy Chatterjee, Anirban Mukhopadhyay, and Malay Bhattacharyya. A weighted rank aggregation approach towards crowd opinion analysis. *Knowledge-Based Systems*, 149: 47–60, 2018.
- Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202, 2013.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.
- Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017.
- Xin Ding, Yongwei Wang, Zuheng Xu, William J Welch, and Z Jane Wang. Ccgan: Continuous conditional generative adversarial networks for image generation. In *International conference on learning representations*, 2020.
- Jesse H. Engel, Matthew D. Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018.
- Johannes Fürnkranz and Eyke Hüllermeier. Preference learning. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1000–1005. Springer, 2017.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27:2672–2680, 2014.

- Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.
- Anvita Gupta and James Zou. Feedback gan for dna optimizes protein functions. *Nature Machine Intelligence*, 1(2):105, 2019.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.
- Jian Huang, Yuling Jiao, Zhen Li, Shiao Liu, Yang Wang, and Yunfei Yang. An error analysis of generative adversarial networks for learning distributions. *Journal of Machine Learning Research*, 23(116):1–43, 2022.
- Harold Jeffreys, Bertha Jeffreys, and Bertha Swirles. *Methods of mathematical physics*. Cambridge university press, 1999.
- Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*, 2019.
- Alexia Jolicoeur-Martineau. On relativistic f-divergences. In *International Conference on Machine Learning*, pages 4931–4939. PMLR, 2020.
- Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020.
- MinJi Kim, Farzad Farnoud, and Olgica Milenkovic. Hydra: gene prioritization via hybrid distance-score rank aggregation. *Bioinform.*, 31(7):1034–1043, 2015.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. On controllable sparse alternatives to softmax. *Advances in neural information processing systems*, 31, 2018.
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.
- Xiao Li, Chenghua Lin, Chaozheng Wang, and Frank Guerin. Latent space factorisation and manipulation via matrix subspace projection. *International Conference on Machine Learning*, 2020.

- Lucy Liang and Kristen Grauman. Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 208–215, 2014.
- Shili Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010.
- Tyler Lu and Craig Boutilier. Learning mallows models with pairwise preferences. In *International Conference on Machine Learning*, pages 145–152, 2011.
- Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *International Conference on Computer Vision*, pages 2794–2802, 2017.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *International Conference on Learning Representations*, 2018.
- Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651, 2017.
- Utkarsh Ojha, Krishna Kumar Singh, Cho-Jui Hsieh, and Yong Jae Lee. Elastic-infogan: Unsupervised disentangled representation learning in class-imbalanced data. In *Advances in Neural Information Processing Systems*, 2020.
- Yuangang Pan, Bo Han, and Ivor W Tsang. Stagewise learning for noisy k-ary preferences. *Machine Learning*, 107(8):1333–1361, 2018.
- Yuangang Pan, Ivor W Tsang, Weijie Chen, Gang Niu, and Masashi Sugiyama. Fast and robust rank aggregation against model misspecification. *J. Mach. Learn. Res.*, 23:23–1, 2022.
- Devi Parikh and Kristen Grauman. Relative attributes. In *International Conference on Computer Vision*, pages 503–510. IEEE, 2011.
- Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-flow: Conditional generative flow models for images and 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7949–7958, 2020.
- Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In *International Conference on Knowledge Discovery and Data Mining*, pages 1037–1046, 2014.
- Bidisha Samanta, Abir De, Gourhari Jana, Vicenç Gómez, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez-Rodriguez. Nevae: A deep generative model for molecular graphs. *The Journal of Machine Learning Research*, 21(1):4556–4588, 2020.

- Yassir Saquil, Kwang In Kim, and Peter M. Hall. Ranking cgans: Subjective control over semantic image attributes. In *British Machine Vision Conference*, page 131. BMVA Press, 2018.
- Sundararajan Sellamanickam, Priyanka Garg, and Sathiya Keerthi Selvaraj. A pairwise ranking based approach to learning with positive and unlabeled examples. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 663–672, 2011.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28:3483–3491, 2015.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907, 2019.
- James Stewart, Daniel K Clegg, and Saleem Watson. *Multivariable calculus*. Cengage Learning, 2020.
- Larissa T Triess, Andre Bühler, David Peter, Fabian B Flohr, and Marius Zöllner. Point cloud generation with continuous conditioning. In *International Conference on Artificial Intelligence and Statistics*, pages 4462–4481. PMLR, 2022.
- Ze Wang, Seunghyun Hwang, Zichen Miao, and Qiang Qiu. Image generation using continuous filter atoms. *Advances in Neural Information Processing Systems*, 34:17826–17838, 2021.
- Christian Wirth, Riad Akrou, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.
- Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, 2008.
- Yinghua Yao, Yuangang Pan, Ivor W Tsang, and Xin Yao. Differential-critic gan: Generating what you want by a cue of preferences. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014.
- Zac Yu and Adriana Kovashka. Syntharch: Interactive image search with attribute-conditioned synthesis. In *Conference on Computer Vision and Pattern Recognition, CVPR Workshops*, pages 645–654, 2020.

- Zhibing Zhao, Peter Piech, and Lirong Xia. Learning mixtures of plackett-luce models. In *International Conference on Machine Learning*, pages 2906–2914. PMLR, 2016.
- Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. Learning to rank with ties. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–282, 2008.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, pages 2223–2232, 2017.