Subverting Fair Image Search with Generative Adversarial Perturbations

Avijit Ghosh  
ghosh.a@northeastern.edu  
Northeastern University  
Boston, MA, USA

Matthew Jagielski  
jagielski@google.com  
Google Research  
Mountain View, CA, USA

Christo Wilson  
cbw@ccs.neu.edu  
Northeastern University  
Boston, MA, USA

ABSTRACT

In this work we explore the intersection of fairness and robustness in the context of ranking: when a ranking model has been calibrated to achieve some definition of fairness, is it possible for an external adversary to make the ranking model behave unfairly without having access to the model or training data? To investigate this question, we present a case study in which we develop and then attack a state-of-the-art, fairness-aware image search engine using images that have been maliciously modified using a Generative Adversarial Perturbation (GAP) model [75]. These perturbations attempt to cause the fair re-ranking algorithm to unfairly boost the rank of images containing people from an adversary-selected subpopulation.

We present results from extensive experiments demonstrating that our attacks can successfully confer significant unfair advantage to people from the majority class relative to fairly-ranked baseline search results. We demonstrate that our attacks are robust across a number of variables, that they have close to zero impact on the relevance of search results, and that they succeed under a strict threat model. Our findings highlight the danger of deploying fair machine learning algorithms in-the-wild when (1) the data necessary to achieve fairness may be adversarially manipulated, and (2) the models themselves are not robust against attacks.

CCS CONCEPTS

• Information systems → Retrieval models and ranking; • Security and privacy;

KEYWORDS

Information Retrieval, Fair Ranking, Adversarial Machine Learning, Demographic Inference

1 INTRODUCTION

The machine learning (ML) community has awoken to concerns about the fairness of ML models, i.e., the elimination of unjustified bias against specific groups of people from models. There is now extensive literature documenting unfairness in deployed ML systems [7, 9, 21] as well as techniques for training fair classification [42, 47, 50, 65] and ranking [28, 86, 98] models. Companies are adopting and deploying fair ML systems in many real-world contexts [4, 10, 96].

Most ML systems that strive to achieve demographic fairness are dependent on high-quality demographic data to control for unjustified biases [13, 94]. Recent work has highlighted how critical this dependency is by showing how unintentional errors in demographic data can dramatically undermine the objectives of fair ranking algorithms [39].

Another serious concern in the ML community is model robustness, especially in the face of clever and dedicated adversaries. The field of adversarial ML has demonstrated that seemingly accurate models are brittle when presented with maliciously crafted inputs [24, 89], and that these attacks impact models across a variety of contexts [12, 31, 46, 89]. The existence of adversarial ML challenges the use of models in real-world deployments.

In this work we explore the intersection of these two concerns—fairness and robustness—in the context of ranking: when a ranking model has been carefully calibrated to achieve some definition of fairness, is it possible for an external adversary to make the ranking model behave unfairly without having access to the model or training data? In other words, can attackers intentionally weaponize demographic markers in data to subvert fairness guarantees?

To investigate this question, we present a case study in which we develop and then attack a fairness-aware image search engine using images that have been maliciously modified with adversarial perturbations. We chose this case study because image retrieval based on text queries is a popular, real-world use case for neural models (e.g., Google Image Search, iStock, Getty Images, etc.), and because prior work has shown that these models can potentially be fooled using adversarial perturbations [101] (although not in the context of fairness). To strengthen our case study, we adopt a strict threat model under which the adversary cannot poison training data [48] for the ranking model, and has no knowledge of the ranking model or fairness algorithm used by the victim search engine. Instead, the adversary can only add images into the victim’s database after the image retrieval model is trained.

For our experiments, we develop an image search engine that uses a state-of-the-art MultiModal Transformer (MMT) [37] retrieval model and a fair re-ranking algorithm (FMMR [51]) that aims to achieve demographic group fairness on the ranked list of image
query results without ever explicitly using demographic labels. Under normal circumstances, where the images are unperturbed, our search engine returns demographically balanced sets of images in response to free text queries. We then train a Generative Adversarial Perturbation (GAP) model [75] that learns from pretrained demographic classifiers to strategically insert human-imperceptible perturbations into images. These perturbations attempt to cause FMMR to unfairly boost the rank of images containing people from an adversary-selected subpopulation (e.g., light-skinned men). Figure 1 shows example image search results produced by our search engine in response to the query “Tennis Player”, with and without our attack.

We present results from extensive experiments demonstrating that our attacks can successfully confer significant unfair advantage to people from the majority class (light-skinned men, in our case)—in terms of their overall representation and position in search results—relative to fairly-ranked baseline search results. We demonstrate that our attack is robust across a number of variables, including the length of search result lists, the fraction of images that the adversary is able to perturb, the fairness algorithm used by the search engine, the image embedding algorithm used by the search engine, the demographic inference algorithm used to train the GAP models, and the training objective of the GAP models. Additionally, our attacks are stealthy, i.e., they have close to zero impact on the relevance of search results.

In summary, we show that GAPs can be used to subvert fairness guarantees in the context of fair image retrieval. Further, our attack is successful under a highly restricted threat model, which suggests that more powerful adversaries will also be able to implement successful attacks. We hypothesize that similar attacks may be possible against other classes of ML-based systems that (1) rely on highly parameterized models and (2) make fairness decisions for inputs that are based on data controlled by adversaries.

The goal of our work is not to hinder or deter the adoption of fair ML techniques—we argue that fair ML techniques must be adopted in practice. Rather, our goal is to demonstrate that fairness guarantees can potentially be weaponized so that the research community will be energized to develop mitigations, e.g., by making models more robust, and by adopting high-quality sources of demographic data that are resistant to manipulation. To facilitate mitigation development without arming attackers, we plan to release our code and data to researchers on request.

2 BACKGROUND

2.1 Fairness in Ranking

Algorithmic decision making is permeating modern life, including high-stakes decisions like credit lending [11], bail granting [7], hiring [96], etc. While these systems are great at scaling up processes with human bottlenecks, they also have the unintended property of embedding and entrenching unfair social biases (e.g., sexism and homophobia). In response, there is a growing body of academic work on ways to detect algorithmic bias [9, 40] and develop classes of fair algorithms, for instance classification [42, 47, 50, 65], causal inference [62, 70], word embeddings [18, 20], regression [3, 15], and retrieval/ranking [28, 86, 98]. There is also a growing body of legal work and legislative action around the globe [1, 30, 33, 53, 73] to tackle algorithmic bias.

In this study we focus on fair Information Retrieval (IR) algorithms—a class of algorithms that have received comparatively less attention than classification algorithms in the literature. Initial studies that examined fair IR proposed to solve this in a binary context, i.e., make a ranked list fair between two groups [28, 98]. Subsequent work uses constrained learning to solve ranking problems using classic optimization methods [86]. There are also methods that use pairwise comparisons [16] and describe methods to achieve fairness in learning-to-rank contexts [68, 99].
In industrial settings, researchers at LinkedIn have proposed an algorithm that uses re-ranking in post-processing to achieve representational parity [38]. However, recent work by Ghosh et al. [39] shows how uncertainty due to incorrect inference of protected demographic attributes can undermine fairness guarantees in IR contexts. Fairness methods that do not require explicit demographic labels at runtime are, as of this writing, an emerging area of focus in classification [55] and ranking [51, 79]. One example that has been studied at large-scale is Shopify’s Fair Maximal Marginal Relevance (FMMR) algorithm [51], which we describe in more detail in § 3.2.2. In this study, we examine the robustness of Shopify and LinkedIn’s fair ranking algorithms.

### 2.2 Adversarial Machine Learning

Adversarial ML is a growing field of research that aims to develop methods and tools that can subvert the objectives of ML algorithms. For example, prior research has highlighted that deep learning models are often not robust when presented with inputs that have been intentionally, maliciously crafted [24, 43, 67, 74, 85, 95, 97].

Several proposed defenses against state-of-the-art adversarial ML attacks have been defeated [8, 91], and adversarial examples (i.e., maliciously crafted inputs) have been shown to transfer across models performing similar tasks [60, 92]. The most promising defense method, adversarial training, is computationally expensive and imperfect—it results in decreased standard accuracy while still having a somewhat low adversarial accuracy [35, 54, 84]. As such, adversarial ML presents a significant hurdle to deploying neural models in sensitive real-world systems.

Our work considers adversarial ML attacks on IR systems. Previous work has demonstrated successful attacks on image-to-image search systems [61, 101], allowing an adversary to control the results of targeted image queries using localized patches [19] or universal adversarial perturbations\(^1\) [57]. Other work has demonstrated attacks on text-to-text retrieval systems [77] and personalized ranking systems [46]. Work by Zhou et al. [101] hypothesized that targeted attacks on selected items in a ranked list might be possible using universal adversarial perturbations. None of these works consider compromising text-to-image models or group fairness objectives, as we do in this study.

Prior work has demonstrated adversarial ML attacks against fairness objectives of ML systems at training time. In these attacks, the adversary supplies poisoned training data, which then results in models that either compromise the accuracy of targeted subgroups [48, 64] or exacerbate pre-existing unfairness between subgroups [29, 87]. Specific to classification, there exists theoretical work that shows how to learn fair models when sensitive attributes are noisy [25] or corrupted in a poisoning attack [27], but they do not consider ranking.

Adversarial ML attacks at test time—i.e., after training a model using non-malicious data—that we consider in this work are relatively unexplored in fairness settings. Nanda et al. [71] show that adversarial ML attacks can harm certain subpopulations more than others in classification tasks. However, while this is an important observation, the harms suggested by this work may be difficult to realize in practice, as they only involve disparity between examples that are adversarially corrupted. By contrast, our work shows that test-time attacks can harm fairness for benign data when launched in a ranking setting.

### 3 METHODOLOGY

We now present the plan for our study. First, we introduce our application context and the threat model under which an attacker will attempt to compromise the application. Second, we discuss the IR models and algorithms underlying our fairness-aware image search engine. Third, we discuss our strategy for attacking this search engine using GAPs.

#### 3.1 Context and Threat Model

In this study, we consider the security of a fairness-aware image search engine. This search engine indexes images from around the web (either automatically via a crawler or from user-provided submissions) and provides a free-text interface to query the image database. Examples of image search engines include Google Image Search, iStock and Getty Images, and Giphy. In our case, the image search engine attempts to produce results that are both relevant to a given query and fair, according to some fairness objective. One example fairness objective is demographic representativeness, i.e., for search results that contain images of people.

We consider a malicious image curator (e.g., Imgur, 4chan, or similar) with a large database of perturbed images that are eventually scraped or uploaded into the victim image search engine’s index.\(^2\) Our adversarial image curator’s goal is to perturb the images in their database to subvert the fairness guarantees of the downstream retrieval system. We assume that the adversary does not have any knowledge of the internals of the ranking system (e.g., what retrieval model is used, other images in the index, or which fairness algorithm is used).

This threat model constitutes a strict, but realistic, limitation on our adversary. Notice that this threat model would also apply if the image search engine was compromised, giving the adversary access to underlying models and the entire dataset of images. We consider both adversaries in our experiments. We also note that, if the adversary only seeks to target a small set of queries, they need only control a fraction of the images matching each query, rather than a fraction of the entire image database. This is useful for the adversary in the case that not all queries are equally sensitive.

#### 3.2 Building an Image Search Engine

We now turn our attention to building a realistic image search engine that will serve as the victim for our attacks.

#### 3.2.1 Image Retrieval from Text Queries

The first choice we make for this study is to select an image retrieval model. There are several frameworks for image retrieval in the literature, starting from

\(^1\)A Universal Adversarial Perturbation (UAP) is an adversarial perturbation that generalizes to all classes of a target classifier, i.e., one patch to untargeted attack as many classes as possible.

\(^2\)An adversarial image curator is also the threat model assumed for clean label poisoning attacks [83, 93]. This adversarial image curator may perturb copies of images taken from the web or original images that they author. This setup is also used by [65] as a defensive method against unauthorized models.
tag-based matching [56] to state-of-the-art vision-language transformers [58, 63]. For the purpose of this paper, we used a Multi-Modal Transformer (MMT) [37] based text-image retrieval model. This model consists of two components: a fast (although somewhat lower quality) retrieval step that identifies a large set of relevant images, followed by a re-ranking step that selects the best images from the retrieved set. Concretely, the user provides a string q that queries into a database D of n images. For the retrieval step, the query string is encoded with an embedding function f_q to produce an embedding v_q, and all images in D have pre-computed embeddings from an embedding function f_i. The cosine distance between v_q and all embeddings of D are computed to collect some large set D_q of size n’ ≤ n plausible image matches. These images are then ranked according to a joint model f, which takes both the query and an image as input, returning scores {s_i}^n_i=1 indicating how well each image D_q[i] matches the query. These scores are used to produce the final ranking.

Note that the MMT model is not designed to be “fair” in any normative sense. To achieve fairness, results from the model must be re-ranked, which we describe in the next section. Thus, the MMT model is not the target of our attacks, since it is not responsible for implementing any fairness objectives.

3.2.2 Fairness-aware Re-ranking. The second choice we make for this study is selecting an algorithm that takes the output of the image retrieval model as input and produces a fair re-ranking of the items. In fairness-aware re-ranking, a ranking function f_r(\tilde{D}, q) is post-processed to achieve fairness according to some subgroup labels on the dataset \tilde{D} = \{s_i, x_i\}^n_i=1, where s_i denotes the score of the i^{th} item (the heuristic score according to which the list is sorted) and x_i denotes the item to be ranked.

The re-ranking algorithm we adopt is Fair Maximal Marginal Relevance (FMMR) [51], which was developed and used at Shopify for representative ranking of images. FMMR builds on the Maximal Marginal Relevance [23] technique in IR that seeks to maximize the information in a ranked list by choosing the next retrieved item in the list to be as dissimilar to the current items present in the list as possible. MMR introduces a hyperparameter that allows the operator to choose the trade-off between similarity and relevance. FMMR modifies the “similarity” heuristic from MMR to encode for similarity in terms of demographics, with the idea being that the next relevant item chosen to be placed in the re-ranked list will be as demographically different from the existing images as possible. Similarity is calculated using image embeddings, for which we examine three models: Faster R-CNN [78], InceptionV3 [88], and ResNet18 [45]. We fix the trade-off parameter \lambda at 0.14 as that is the value used by Karako and Manggala [51] in their FMMR paper.

It is notable that FMMR does not require demographic labels of people in images to perform fair re-ranking, since it uses a heuristic that only relies on embeddings. Indeed, FMMR comes from a class of fair ranking algorithms that all use the inherent latent representations of the objects for their re-ranking strategy [51, 79]. That said, since FMMR attempts to maximize the distance from the centroids of the embeddings of different demographic groups, it can be thought of as performing indirect demographic inference on individuals in images.

Additionally, we also evaluated our attacks against a second fairness-aware re-ranking algorithm, DetConSort [38], developed by and deployed at LinkedIn in their talent search system. Unlike FMMR, DetConSort explicitly requires demographic labels for the items it is trying to fairly re-rank. However, prior work [39] shows that DetConSort has significant limitations when demographic inference is used rather than ground-truth demographic labels, making it unfair even without perturbed images. As a result, evaluating an attack against DetConSort is not meaningful, and we defer our discussion of DetConSort to § A.1.

3.3 Attack Construction

Having described our search engine, we are ready to turn our attention to our attack. First, we introduce the demographic inference models (Deepface [90] and FairFace [52]) that we use to train our attack. Next, we describe how we generate adversarial perturbations from a demographic inference model, modifying images in a way that is imperceptible to human eyes, yet significant enough to fool the fair re-ranking algorithm of our search engine.

3.3.1 Demographic Inference Algorithms. For large-scale datasets such as images scraped from the web, demographic meta-data for people in the images is (1) not readily available and (2) prohibitively expensive to collect through manual annotation [6, 17]. Pipelines using demographic inference are commonly used in practice when demographic labels are not available. For example, the Bayesian Improved Surname Geocoding (BISG) tool is used to measure fairness violations in lending decisions [2, 22], and it relies on inferred demographic information. This makes attacks on demographic inference models a natural candidate for adversely affecting ranking fairness.

We consider two image demographic inference models to train our attacks:

1. Deepface [90] is a face recognition model for gender and race inference developed by Facebook. We use its public wrapper [82], which includes models fine tuned on roughly 22,000 samples for race and gender classification.
2. FairFace [52] is a model designed for race and gender inference, trained on a diverse set of 108,000 images.

Since both of these models infer race/ethnicity, we used a mapping to infer skin tone, since we could not find commercially available algorithms to infer skin tones from human images.\(^3\) We also use these models to infer demographics as input to the DetConSort algorithm, matching the pipeline of [39], which we discuss in § A.1.

3.3.2 Subpopulation Generative Adversarial Perturbations. Recall our adversarial image curator’s goal: to produce a database of malicious images that, when indexed by our image search engine, undermine its purported fairness guarantees. Concretely, this means fooling the fair re-ranker such that it believes a given set of search results is fair across two or more subgroups, when in fact the results are unfair because some subgroups are under- or over-represented. Additionally, these malicious images must (1) retain their relevance

\(^3\)The mapping we used is: White, East Asian, Middle Eastern → Light, and Black, South Asian, Hispanic → Dark. We acknowledge that this is a crude mapping, but it enabled us to train a successful attack.
to a given query and (2) not be perceived as “manipulated” to human users of the search engine.

Prior work (see §2.2) has demonstrated that neural image classification models can be fooled by adding adversarial perturbations to images. At a high-level, the adversary’s goal is to train a model that can add noise to images such that specific latent characteristics of the images are altered. In our case, these altered characteristics should impact the image embeddings calculated by the image embedding model (e.g., InceptionV3) that FMMR relies upon to do fair re-ranking.

Running an adversarial perturbation algorithm on each of the images in the adversary’s database would be prohibitive, as these algorithms involve computationally expensive optimization algorithms that are not practical at the scale of an entire database. We avoid this limitation by training a Generative Adversarial Perturbation (GAP) model [75]. A GAP model $f_{\text{GAP}}$ takes a clean image as input and returns a perturbed image that is misclassified by some target model $f_{\text{targ}}$. This replaces the per-image optimization problem with a much less expensive forward pass of $f_{\text{GAP}}$. Training the GAP is a one time expense for the adversary, amortized over the large number of image perturbations done later. Universal Adversarial Perturbations (UAPs) [66] are another approach to amortizing runtime, but require all images to be the same dimensions—an unrealistic assumption for real-world image databases.\(^4\)

Having motivated the choice of a GAP model for our attack, we now consider the problem of impacting fairness by attacking the fair re-ranking algorithm used by a victim search engine. We choose to design a GAP to target a demographic inference model $f_{\text{DI}}$.\(^5\) This will produce perturbations that, to a deep image model, make an image of a person from one demographic group appear to be from a different demographic group. This attack would heavily impact a demographic-aware re-ranking algorithm such as DetConSort [38] (see §A.1) if it used an accurate demographic inference algorithm to produce annotations.

Although FMMR does not use annotations, we show in §5 that our attack is still successful at compromising FMMR’s fairness guarantees. Our attack can be seen as an application of the transferability property of adversarial examples. Additionally, training our GAP against a demographic inference model causes our attack to be independent of the ranking algorithm and image corpus used by the victim search engine, both of which are strong adversarial assumptions.

In designing our GAP to compromise fairness, we first note that an attack that simply forces a $f_{\text{DI}}$ to make arbitrarily many errors may not impact fairness. For example, suppose the image database contained two subpopulations, the advantaged class $A$ and the disadvantaged class $B$. Suppose the attack causes $f_{\text{DI}}$ to misclassify all members of $B$ as $A$ and all members of $A$ as $B$. This is the best possible result of an attack on the demographic inference algorithm, but results in no changes to a fair ranking algorithm—it will simply consider $A$ to be the disadvantaged class, and thus produce the same ranking! For this reason, our adversary must incorporate subpopulations into the attack. To do so, we propose the Class-Targeted Generative Adversarial Perturbation (CGAP):

**Definition 1 (CGAP).** We consider a loss function $\ell$, target model $f_{\text{targ}}$, distribution $D$ over inputs $x$ and outputs $y$. The adversary provides a source class $y_s$ and target class $y_t$. Then the CGAP model $f_{\text{CGAP}}$ is a model that takes as input an image $x$ and returns an image $x'$, minimizing the following loss functions:

\[
\ell^s_{\text{CGAP}}(D) = \mathbb{E}_{(x, y) \sim D}[\ell(f_{\text{CGAP}}(x), y) | y = y_s],
\]

\[
\ell^r_{\text{CGAP}}(D) = \mathbb{E}_{(x, y) \sim D}[\ell(f_{\text{CGAP}}(x), y) | y \neq y_s].
\]

That is, the CGAP should force the demographic inference model to misclassify samples of class $y_s$ to class $y_t$, while maintaining its performance for samples not from class $y_s$.

We also consider two extensions of this definition. First, we permit the adversary to target multiple classes at once. In the extreme, an adversary may want all samples to be classified to the same class (this approach is proposed by [75]). For a demographic inference algorithm, all samples having the same demographic label will cause the fair re-ranking system to have similar performance to an unfair ranking system, as all points will appear to fall into the same subpopulation. The second extension is the untargeted attack, where the CGAP simply increases loss for points from class $y_s$, inducing arbitrary misclassifications. Simultaneously making both relaxations recovers the original untargeted GAP approach. We experiment with both relaxations independently, as well as multiple instantiations of CGAP as defined above.

## 4 EXPERIMENTS

In this section we introduce the dataset we used for our evaluation, describe the setup for our experiments, and define the metrics we use to evaluate our attacks.

### 4.1 Dataset, Annotation, and Preprocessing

We use Microsoft’s Common Objects in Context (MS-COCO) [59] as our retrieval dataset, since it contains a variety of images with variable dimensions and depths. This closely mimics what a real-world image search dataset might contain.

To specifically measure for demographic bias, we filter the dataset, keeping only images that contain people. We also need the images to have demographic annotations for fair ranking, so we use an annotated subset of the COCO 2014 dataset, constructed by Zhao et al. [100]. Similar to prior work [26, 39], Zhao et al. crowdsource skin color (on the Fitzpatrick Skin Type Scale, which the authors simplified to Light and Dark) and binary perceived gender expression for 15,762 images. For the purposes of our experiments we only considered the 8,692 images that contain one person. After filtering, our final dataset consisted of 5,216 Light Men, 2,536 Light Women, 714 Dark Men, and 226 Dark Women.

### 4.2 Experimental Setup

As a starting point for our experiments, we need to collect ranked lists from our baseline, unfair retrieval system, as described in §3.2.1. To do so, we run three different search queries on the retrieval system: “Tennis Player”, “Person eating Pizza”, and “Person at table”. We chose these queries because they all reference a human

---

\(^4\) A UAP can also be seen as a GAP, where $f_{\text{GAP}}(x) = x + \delta$ for a fixed $\delta$. Therefore, we expect that a GAP will perform strictly better than a UAP.

\(^5\) Recall that, per our threat model in §3.1, the attacker does not know what fair re-ranking algorithm is used by the victim and thus cannot train against it directly.
being, are ethnicity and gender neutral, and are well-supported in the COCO dataset (we picked popular object tags, see § A.2). We set the upper bound in the baseline retrieval system to be 200 images. The three queries return 131, 75, and 124 images, respectively, along with their relevance scores.

We show the distribution of the relevance scores and the skin color/gender distributions of the images within the top 40 search results for each query in Figure 2. As also shown by Zhao et al. [100], Light Men comprise the overwhelming majority in all three lists, and they also have high relevance scores across the board, meaning that the retrieval system places Light Men near the top of the search results. We call these lists the baseline lists.

We also need to produce fair versions of the baseline lists. To do so, we pass the baseline lists for each of our three queries through FMMR with the three embedding algorithms, without any adversarial perturbations. We refer to the nine lists obtained via the fair re-ranker (three queries times three image embedding models) as the oracle lists.

To train our adversarial attacks, we first remove the 330 images in our oracle lists from the original dataset, leaving 8,362 images. These 8,362 images were then split randomly into training and testing sets in an 8:2 ratio to train CGAP models for all possible combinations of training objectives and demographic inference algorithms $f_{DI}$ (described in detail below). We ran our experiments on PyTorch with a CUDA backend on two NVIDIA RTX-A6000 GPUs, and trained CGAP models for 10 epochs each, with the $L_{\infty}$ norm bound set to 10.

We describe our different training and inference combinations below. Table 1 shows a summary of the different settings involved during the training and testing of our CGAP attacks.

<table>
<thead>
<tr>
<th>Search Queries</th>
<th>Attack Training</th>
<th>Embedding</th>
<th>Training Objective</th>
<th>Attack Probability</th>
<th>Top $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Tennis Player&quot;</td>
<td>DeepFace</td>
<td>F-RCNN</td>
<td>Any $\rightarrow$ Light Men</td>
<td>0.2, 0.5, 0.7, 1.0</td>
<td>10, 15, 20..., 45, 50</td>
</tr>
<tr>
<td>&quot;Person eating pizza&quot;</td>
<td>FairFace</td>
<td>InceptionV3</td>
<td>Light Men $\rightarrow$ Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;Person at Table&quot;</td>
<td>ResNet</td>
<td>Light Men $\rightarrow$ Light Men</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Light Men $\rightarrow$ Dark Men</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Light Men $\rightarrow$ Dark Men</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Variables and hyperparameters we used for evaluating our attack.

To train our adversarial attacks, we first remove the 330 images from the original dataset, leaving 8,362 images. These 8,362 images were then split randomly into training and testing sets in an 8:2 ratio to train CGAP models for all possible combinations of training objectives and demographic inference algorithms $f_{DI}$ (described in detail below). We ran our experiments on PyTorch with a CUDA backend on two NVIDIA RTX-A6000 GPUs, and trained CGAP models for 10 epochs each, with the $L_{\infty}$ norm bound set to 10.

We describe our different training and inference combinations below. Table 1 shows a summary of the different settings involved during the training and testing of our CGAP attacks.

4.2.1 Embedding Algorithm. As we discuss in § 3.2.2, FMMR requires image embeddings. The authors of the original paper used a pretrained InceptionV3 model, which we also adopt. Additionally, we test the performance of FMMR using embeddings generated by pretrained Faster R-CNN and ResNet models. These models are trained for standard image classification tasks and have no inherent concept of demographic groups.

4.2.2 Attack Training Algorithm. As detailed in § 3.3, we train CGAP models to induce adversary-selected misclassifications in two target demographic inference models, denoted as $f_{DI}$: Deepface [90] and FairFace [82]. These models are trained for demographic inference, and so do not overlap in training objective with the image embedding models for FMMR. The only similarity in architecture between the demographic inference and FMMR embedding models is that FairFace uses a ResNet architecture.

4.2.3 Training Objectives. As discussed in § 3.3.2, we select certain subpopulations to be systematically misclassified by the two $f_{DI}$ described above. The four CGAPs we train induce misclassifications with the following source-target pairs: Any $\rightarrow$ Light Men, where every subgroup was perturbed to be predicted as Light Men; Light Men $\rightarrow$ Any, where only Light Men are arbitrarily misclassified; Dark Men $\rightarrow$ Light Men, where only Dark Men are misclassified as Light Men; and Light Men $\rightarrow$ Dark Men, where only Light Men are misclassified as Dark Men.

4.2.4 Attack Probability $p_r$. It is a strong assumption that an adversary can perturb the entire image database of a victim search engine. This is only possible if the search engine itself is malicious or it is utterly compromised. Instead, we measure the effect of our attack when the attacker may perturb $p_r$ = 20%, 50%, 70% and 100% of the image database relevant to each query. If a small number of queries are targeted, only few images are required to run the attack.

4.2.5 Top $k$. Ranking is very sensitive to position bias [39, 80], so we measure with different lengths $k$ of the top list, ranging from top 10 to top 50, to gauge our attack’s impact on the fair ranking algorithms as final list sizes vary.

4.3 Evaluation Metrics

To evaluate the impact of our attacks, we use three metrics that aim to measure (1) representation bias, (2) attention or exposure bias, and (3) loss in ranking utility due to re-ranking. Additionally, we introduce a summarizing meta-metric that enables us to clearly present the impact of our attacks with respect to each metric.

4.3.1 Skew. The metric we use to measure the bias in representation is called Skew [38, 39]. For a ranked list $r$, the Skew for attribute value $a_i$ at position $k$ is defined as:

$$\text{Skew}_{a_i}@k(r) = \frac{p_{r^k,a_i}}{p_{q,a_i}}$$

$p_{r^k,a_i}$ represents the fraction of members having the attribute $a_i$ among the top $k$ items in $r$, and $p_{q,a_i}$ represents the fraction of members of subgroup $a_i$ in the overall population $q$. In an ideal, fair representation, the skew value for all subgroups is equal to 1, indicating that their representation among the top $k$ items exactly matches their proportion in the overall population.

4.3.2 Attention. Even if all subgroups were fairly represented in the top $k$ ranked items of a list, the relative position of the ranked items adds another dimension of bias—unequal exposure. Previous
In this study, we model attention decay using the geometric distribution as done in prior work by Sapiezynski et al. [80]. We compute attention at the $k^{th}$ rank as:

$$\text{Attention}_{p}(k) = 100 \times (1 - p)^{k-1} \times (p)$$ (2)

where $p$ is the fraction of total attention given to the top search result. The choice of $p$ is application specific—for this paper we fixed $p$ to be 0.36, based on a study [44] that reported that the top result on Google Search receives 36.4% of the total clicks. We then calculate the average attention per subgroup:

$$\text{Average attention}_{a_i, \tau} = \frac{1}{|a_i|} \sum_{k=1}^{|a_i|} \text{Att}(k) \text{ where } a_i^k = a_i.$$ (3)

Ideally, in a perfectly fair ranked list, all subgroups should receive equal average attention.

4.3.3 Normalized Discounted Cumulative Gain. NDCG is a widely used measure in IR to evaluate the quality of search rankings [49]. It is defined as

$$\text{NDCG}(\tau) = \frac{1}{Z} \sum_{i=1}^{|\tau|} \frac{s_f^i}{\log(1 + i)}$$ (4)

where $s_f^i$ is the utility score from the MMT retrieval model of the $i^{th}$ element in the ranked list $\tau$ and $Z = \sum_{i=1}^{|\tau|} \frac{1}{\log(1 + i)}$. NDCG scores range from 0 to 1, with the latter capturing ideal search results.

4.3.4 Summarizing Metric. For the purpose of quantifying how much unfair advantage our attacks confer on members of the majority class relative to all other classes, we define a new meta-metric called Attack Effectiveness $\eta$. For a given metric $m \in \{\text{Skew, Attention}\}$ and a subgroup $g$, it is defined as:

$$\eta(m, g) = \% \text{ change in } m \text{ for subgroup } g - \% \text{ minimum change in } m \text{ over other subgroups.}$$ (5)

We chose this formulation of $\eta$ for two reasons. First, comparing percentage changes makes the metric scale invariant, which is useful since group sizes vary. Second, comparing to the group that gets the minimum boost ensures that the metric presents the widest fairness disparity, regardless of the total number of groups.

For the purposes of this paper, we set $g$ as Light Men, because they are socially and historically the most advantaged group, and a large $\eta$ for Light Men indicates that the attack causes their ranking to be unfairly boosted relative to the least privileged group. To make sure that the fairness impacts we observe are due to the effectiveness of our attack on the re-ranking algorithms only, the $\eta$ values and the % change in NDCG are all measured against the oracle (i.e., fairly re-ranked) lists. Because we compare against the oracle list, all results with attack probability $pr = 0$ will have $\eta = 0$.

5 RESULTS

In this section, we evaluate the impact of our attacks on the fairness guarantees of FMMR. For each set of results we examine how attack effectiveness varies for one particular variable (e.g., top $k$, image embedding model, etc.) as the attack probability $pr$ (i.e., the fraction of images under adversarial control) varies. When focusing on a particular variable, we present results that are averaged across all other variables and all three of our queries.

5.1 Top $k$ and $pr$

We begin by evaluating the impact of our attacks as we vary the length of the top list $k$ and the fraction of images in the query list under adversarial control $pr$, plotted in Figure 3.

Varying $pr$ has the expected effect: as the adversary has more control over the image database, attacks become more effective, i.e., $\eta$ for skew and attention increase. When the adversary is able to control 100% of images in the query list, attacks are especially strong—increasing attention unfairness by over 50% for some values of $k$. Even with only 20% control, the adversary can increase attention unfairness by ~30%. Recall that $pr$ measures the fraction of each query list that is compromised, so as few as 35 images can be compromised at $pr = 0.5$ (for the ‘Person eating Pizza’ query).

Varying $k$ also impacts ranking fairness. As $k$ increases, attention unfairness increases modestly and skew unfairness decreases. That skew unfairness decreases with $k$ indicates that the composition of items in the search results becomes fairer as the length of the list...
top K
η (skew, light men)

(a) Skew

η (attention, light men)

(b) Attention

% Change NDCG

(c) NDCG

Figure 3: Attack effectiveness as a function of attack probability $pr$ and list length $k$. Higher $\eta$ is a more effective attack, i.e., the search results are more favorable to light-skinned men. Unfairness increases as $pr$ increases, yet there is almost no impact on ranking quality (NDCG). As $k$ increases skew is less impacted but attention is impacted somewhat more.

Figure 4: Attack effectiveness is stable when the model used for the FMMR embedding is changed. ResNet embeddings are slightly more robust to attack and F-RCNN are slightly less robust. Interestingly, the ResNet’s robustness is in spite of it having the most similar model architecture to FairFace.

grows. However, our attack is able to cause FMMR to reorder the list such the top-most items remain unfair regardless of $k$, which is why attention unfairness exhibits less dependency on $k$.

Lastly, we observe that our attacks are stealthy. Regardless of $k$ or $pr$, NDCG never changes more than 0.7%, meaning that our attack had effectively zero impact on search result relevance.

5.2 Choice of Training Objective

We evaluate our attack’s impact on fairness with four CGAP models: one that misclassifies Dark Men as Light Men, one for misclassifying Light Men as Dark Men, and relaxed CGAP models that misclassify all people as Light Men and all Light Men as other groups. We show these attacks’ effectiveness in Figure 5.

Each of these attacks performs similarly well at harming fairness in terms of skew and attention, and remaining stealthy in terms of NDCG. One surprising observation is that misclassifying Dark Men as Light Men performs similarly to the exact opposite attack: in both cases, Light Men end up with an significant, unfair advantage.

We explain this seeming contradiction with an example in Figure 6. In essence, using a GAP to misclassify people from a minority group into the majority group reduces the minority group’s overall share of the population. Since group fairness in this case is based on the overall population distribution, this causes FMMR to rerank fewer minority group members into the top of the search results.

Based on the results in Figure 5, it appears that there is no way to advantage a minority group with our attacks.

5.3 Choice of Attack Training Algorithm

We measure our attacks’ effectiveness when the GAP models are trained on Deepface and FairFace demographic inference models. We observe that attack effectiveness is largely independent of the choice of inference model, and all attacks remain stealthy. We defer a plot of the results to the supplementary material, in Figure 8.
5.4 Choice of Query

Lastly, we examine the effectiveness of our attacks against three different queries and plot the results in Figure 7. We observe that all attacks were successful, but that effectiveness varies by query. The differences in attack effectiveness are explained by the underlying distributions of population and utility scores (see Figure 2). The "tennis" results exhibit the most unfairness post-attack because they were most unfair to begin with, i.e., the difference in utility scores between Light and Dark skinned people was greatest in the "tennis" results as compared to the other queries. In contrast, the "pizza" results exhibit the most robustness to attack in terms of attention because these were the only results among the queries where minority people had higher utility scores than majority people in the baseline results (Figure 2).

6 DISCUSSION

In this study, we develop a novel, adversarial ML attack against fair ranking algorithms, and use fairness-aware text-to-image retrieval as a case study to demonstrate our attack’s effectiveness. Unfortunately, we find that our attack is very successful at subverting the fairness algorithm of the search engine—across an extensive set of attack variations—while having almost zero impact on search result relevance.
Although we present a single case study, we argue that our attack is likely to generalize. We adopt a strong threat model and demonstrate that our attacks succeed even when the attacker cannot poison training data, access the victim’s whole image corpus, or know what models are used by the victim. Thus, our attack is highly likely to succeed in cases where the threat model is more relaxed, e.g., when the fairness algorithm used by the victim is known.

Alarming ly, our work shows that an adversary can attack a fairness algorithm like FMMR even when it does not explicitly rely on demographic inference. Thus, it is highly likely that our attack will also succeed against any ranking algorithm that does rely on a demographic inference model, even if that model is highly accurate. We explore this possibility to the best of our ability in § A.1.

We hope that this research will raise awareness and spur further research into vulnerabilities in fair algorithms. Our results highlight how, in the absence of safeguards, fairness interventions can potentially be weaponized by malicious parties as a tool of oppression. In the absence of fair ML development methods and algorithms that are robust to adversarial attacks, it may not be possible for policymakers to safely mandate the use of fair ML algorithms in practice.

We believe that future work is needed to develop more robust fair ML interventions. We adopt a broad view of possible mitigations, spanning from value sensitive design [34] methods that help developers preemptively identify attack surface and plan defenses [32], to models that are hardened against adversarial perturbation techniques [5, 43], to auditing checklists [76] and tools that help developers notice and triage attacks.

Above all, this work highlights that achieving demographic fairness requires high-quality demographic data [6]. Allowing an adversary to influence demographic meta-data is the underlying flaw that enables our attack to succeed. Demographic data may be sourced from data subjects themselves, with full knowledge and consent, or from human labelers [10], with the caveat that these labels themselves will need to be de-biased [100].

6.1 Limitations

Our study has a number of limitations. First, our analysis is limited to two discrete racial and two discrete gender categories. Although our CGAP attack could be tailored to select any group, it is unclear how well our attack would perform in situations with > 4 discrete protected groups, groups with continuous attributes, people with multiple or partial group memberships, or with population distributions that varied significantly from our dataset. Second, while our dataset is sufficiently large to demonstrate our attack, it is smaller than the databases that real-world image search engines retrieve from. Third, our proof-of-concept was tuned to attack FairFace and DeepFace. It is unclear how well our CGAP attack would generalize to other models or real-world deployed systems. Fourth, as we observe in Figure 5, our attack is only successful at generating unfairness in favor of already-advantaged groups. While this is a limitation, it in no way diminishes the potential real-world harm our attack could inflict on marginalized populations. Finally, as shown in Figure 7, our attack’s effectiveness varies by query. In real world scenarios, an attacker could mitigate this to some extent by devoting more of their resources towards perturbing images that are relevant to high-value queries. It is unclear how much the attackers’ effort would need to vary in practice given that we make no attempt to attack deployed search engines.

6.2 Ethics

In this work we present a concrete attack against a fair ranking system. Like all adversarial attack research, our methods can potentially be misused by bad actors. However, this also necessitates our research, since documenting vulnerabilities is the first step towards mitigating them. To the best of our knowledge, with the exception of Shopify and LinkedIn, few services are known to employ fair ranking systems in practice, meaning there currently exists a window of opportunity to preemptively identify attacks, raise awareness, and deploy mitigations.

Prior work on adversarial ML attacks against fairness made their source code publicly available [71]. However, because attack tools are dual-use, we have opted to take a more conservative approach: we will only share source code with researchers from (1) research universities (e.g., as identified by taxonomies like the Carnegie Classification) and (2) companies that develop potentially vulnerable products. Given that our attack can be used for legitimate, black-box algorithm auditing purposes, we opt to restrict who may access
our source code rather than the uses it may be put towards. In our opinion, this process will facilitate follow-up research, mitigation development, and algorithm auditing without supplying bad actors with ready-made attack tools. Like many works in the computer vision field, we rely on images with crowdsourced and inferred demographic labels. Both processes have been criticized for their lack of consent [41], the way they operationalize identity [81], and the harm they may cause through mis-identification [14]. These problems reinforce the need for high-quality, consensual demographic data as a means to improve ethical norms and defend against adversarial ML attacks.

ACKNOWLEDGMENTS

Funding/Support. This work was supported by a 2019 Sloan Fellowship and NSF grants IIS-1553088 and IIS-1910064. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the funders.

The legal department of Google participated in the review and approval of the manuscript. They asked that the authors add additional examples of image search engines besides Google Image Search, and we complied. Aside from the authors, Google had no role in the design and conduct of the study; access and collection of data; analysis and interpretation of data; or preparation of the manuscript.

The authors thank Jane L. Adams, Michael Davinroy, Jeffrey Gleason, and the anonymous reviewers for their comments.

REFERENCES


[34] Batya Friedman and David Hendry. 2019. Value sensitive design: shaping technology with moral imagination. MIT Press.
To facilitate our experiments, we chose to select search query terms that would provide a sizeable list of images. To do so, we looked at the list of terms in the COCO image captions (excluding English stop words and words related to ethnicity or gender). The following terms were among the most popular terms.

“A.1.1 DetConstSort. For DetConstSort, the skew and attention metrics are not impacted by our attack. This can be clearly seen by comparing the $\eta$ values when $pr = 0$ (i.e., there are no perturbed images) to other values of $pr$: for DetConstSort, $\eta$ for skew and attention starts high (unfair) when $pr = 0$, and does not change as $pr$ increases. The correct interpretation of these results is that DetConstSort is resilient to our attack. Rather, the correct interpretation is that DetConstSort starts off unfair due to the use of inaccurate, inferred demographic data [39], and our attack is unable to make the unfairness worse.

Thus, we find that a prerequisite for evaluating the success of our attacks on DetConstSort is an accurate demographic inference model. Developing such models is still an active area of research, and is out-of-scope for our work. Should a more accurate demographic inference model be designed in the future, however, it must be designed with adversarial robustness in mind to prevent our attacks.

A.2 Choice of Queries

To facilitate our experiments, we chose to select search query terms that would provide a sizeable list of images. To do so, we looked at the list of terms in the COCO image captions (excluding English stop words and words related to ethnicity or gender). The following table shows some top terms. From this information, we composed our three queries given that “sitting”, “tennis”, “table”, “person”, “pizza”, etc. were among the most popular terms.
Figure 8: GAP models trained on different demographic inference algorithms offer similar attack effectiveness.

Figure 9: DetConstSort has poor performance even without an attack, making our results uninteresting.

<table>
<thead>
<tr>
<th>Term</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>sitting</td>
<td>55084</td>
</tr>
<tr>
<td>standing</td>
<td>44121</td>
</tr>
<tr>
<td>people</td>
<td>42133</td>
</tr>
<tr>
<td>holding</td>
<td>29055</td>
</tr>
<tr>
<td>large</td>
<td>25305</td>
</tr>
<tr>
<td>person</td>
<td>25123</td>
</tr>
<tr>
<td>street</td>
<td>21609</td>
</tr>
<tr>
<td>table</td>
<td>20775</td>
</tr>
<tr>
<td>small</td>
<td>20661</td>
</tr>
<tr>
<td>tennis</td>
<td>19718</td>
</tr>
<tr>
<td>riding</td>
<td>18809</td>
</tr>
<tr>
<td>train</td>
<td>18287</td>
</tr>
<tr>
<td>young</td>
<td>17767</td>
</tr>
<tr>
<td>red</td>
<td>17522</td>
</tr>
<tr>
<td>baseball</td>
<td>15362</td>
</tr>
<tr>
<td>pizza</td>
<td>11163</td>
</tr>
</tbody>
</table>

Table 2: The most common (gender or race unrelated) caption terms in the evaluation dataset.