Attack Some while Protecting Others: Selective Attack Strategies for Attacking and Protecting Multiple Concepts*

Vibha Belavadi
University of Texas at Dallas
Richardson, Texas, USA
VibhaChandramouli.Belavadi@utdallas.edu

Yan Zhou
University of Texas at Dallas
Richardson, Texas, USA
yan.zhou2@utdallas.edu

Murat Kantarcioglu
University of Texas at Dallas
Richardson, Texas, USA
muratk@utdallas.edu

Bhavani Thuraisingham
University of Texas at Dallas
Richardson, Texas, USA
bxt043000@utdallas.edu

ABSTRACT

Machine learning models are vulnerable to adversarial attacks. Existing research focuses on attack-only scenarios. In practice, one dataset may be used for learning different concepts, and the attacker may be incentivized to attack some concepts but protect the others. For example, the attacker might tamper a profile image for the "age" model to predict "young", while the "attractiveness" model still predicts "pretty". In this work, we empirically demonstrate that attacking the classifier for one learning task may negatively impact classifiers learning other tasks on the same data. This raises an interesting research question: is it possible to *attack one set of classifiers while protecting the others trained on the same data?*

Answers to the above question have interesting implications for the complexity of test-time attacks against learning models, such as avoiding the violation of logical constraints. For example, attacks on images of high school students should not cause these images to be classified as a group of 30-year-old. Such misclassification of age may raise alarms and may easily expose the attacks. In this paper, we address the research question by developing novel attack techniques that can simultaneously attack one set of learning models while protecting the other. In the case of linear classifiers, we provide a theoretical framework for finding an optimal solution to generating such adversarial examples. Using this theoretical framework, we develop a "multi-concept" attack strategy in the context of deep learning tasks. Our results demonstrate that our techniques can successfully attack the target classes while protecting the "protected" classes in many different settings, which is not possible with the existing test-time attack-only strategies.

*An extended version with the appendices is available at https://arxiv.org/abs/2110.10287

CCS '23, November 26–30, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0050-7/23/11…$15.00
https://doi.org/10.1145/3576915.3623177

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Artificial intelligence**; **Neural networks**; **Machine learning approaches**;

KEYWORDS

Neural networks; Adversarial examples

ACM Reference Format:

Vibha Belavadi, Yan Zhou, Murat Kantarcioglu, and Bhavani Thuraisingham. 2023. Attack Some while Protecting Others: Selective Attack Strategies for Attacking and Protecting Multiple Concepts. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3576915.3623177

1 INTRODUCTION

Machine learning (ML) is greatly involved in our daily life, including shopping, finance, healthcare decisions, and more. Recent work has shown that ML models could be easily attacked at application time, fooled by data corrupted with noise. For example, an attacker can modify the facial images using existing attack techniques (e.g., attacking DNN [20]) to distort its embedded sexual orientation to fool the *sexual orientation* classifier. Despite the fact that existing test-time attacks are effective, they typically target only one learning model at a time, which may have undesired side effects in practice, especially when the same data is used for training different concept models. In this context, a machine learning concept is a discrete-valued function defined over training data in a domain. A concept model is a specific formulation of the function that relates data input to decision output in ML tasks. While attacking one classifier (e.g., predicting sexual orientation), the attack may reduce the accuracy of another classifier (e.g., predicting gender). This raises an interesting research question: *Is it possible to craft test-time attacks that only foil one set of classifiers without impacting the other that uses the same test instance?*

Attacking one classifier while preserving the others has important applications in real life. For example, when an attacker tries to tamper the images of high schoolers to pass a job eligibility test, the attack should not cause these images to be classified as a much older generation. Such misclassification with respect to age may

raise alarms and makes the discovery of such attacks a much easier task. Similarly, an attack that modifies online political messages for bypassing spam filters may not want to change the message's political agenda so much that it fails to populate on other platforms, such as Google alerts.

In addition, attacking one classifier while preserving the others has important implications for hiding sensitive information from discriminatory classifiers. For example, an ML classifier may be trained to predict privacy-sensitive information such as sexual orientation (e.g.,[27, 33]), political affiliation (e.g., [13, 33]), attractiveness [10, 15], and life satisfaction [2, 28] using social network profiles. Although some of these usages seem innocuous, the deployment of such ML models may be seen as a threat to individual privacy. [1] Therefore, launching a test time attack against one privacy-invasive model without impacting the performance of other ML models could be beneficial for such scenarios.

The following is an illustrating use case of selective attacks. Suppose a user is interested in product recommendations based on her shopping data. The data is supplied to various companies that use their in-house models to extract implicit information such as gender and age not shared directly in the user data. The extracted information is then used to help recommend products to the user. While the user is conscious about sharing her private information online, she may prefer recommendations based on "age" over "gender". Thus, in this scenario, the user could create "adversarial" examples that attack one concept (gender) and protect another (age).

To address the challenge of simultaneously attacking one set of machine learning models while protecting the other, we propose a novel test-time selective attack algorithm. Unlike the existing test-time attacks, our algorithm considers attacking-and-protecting multiple classifiers, and carefully adjusts the utility function to find an attack that considers multiple constraints (e.g., attack the sexual orientation classifier while minimizing the impact on the attractiveness classifier, or attack both sexual orientation and attractiveness classifier, etc.). For linear classifiers, we provide an optimal formalization of the problem and solve it using linear programming. Later on, using this theoretical foundation, we provide an attack algorithm for Deep Neural Network models (DNNs) and show that it can reduce the side effect of the traditional attacks that do not consider these multiple attack constraints. The contributions of our work can be summarized as follows:

- We empirically show that the existing single-model attacks may reduce the accuracy of other models on the same data.
- We provide a theoretical formalization of our multi-concept attack problem for linear classifiers and provide optimal solutions using linear programming.
- We extend our theoretical model developed for linear classifiers to address similar challenges encountered in learning with DNNs by integrating multiple objectives.
- Using extensive empirical evaluation, we show that our multi-concept attack algorithm significantly improves the targeted attack performance compared to the existing test-time attack techniques.
- Using the Shapely value, an indicator of each feature's contribution to prediction, we illustrate the features most influential to the success of attacks, and how our attack model successfully preserves the important features of the protected concepts.

The rest of the paper is organized as follows. In Section 2, we discuss related work in the domain of adversarial examples, specifically the multi-concept classification scenario, and explain the motivation behind our work. In section 3, we propose the generic problem formulation and show that it could be solved for linear and non-linear (e.g. deep neural networks) classifiers. In Section 4, we present the experimental results for the selective attack against both Linear and non-Linear Classifiers. Section 5 concludes our work.

2 RELATED WORK

The term *adversarial example*, together with its impact on the accuracy/reliability of Deep Neural Networks (DNNs), was introduced formally in [30]. Since then, there have been a wide variety of adversarial attacks and defenses [35, 36]. Adversarial training [5, 14, 34] was proposed as a way to produce more robust models. There were several works [3, 31, 32] that discuss the interpretability of DNNs for adversarial examples. [36] and [35] provide a recent summary of existing adversarial attack and defense methods.

For datasets with images having multiple labels, the labels themselves may be related to each other (e.g., subclass/superclass) making it possible to exploit this for improved adversarial robustness and adversarial example detection. The work in [22] takes user domain knowledge of these relationships formulated in the form of First-Order Logic (FOL) constraints. This approach requires a superclass to create rejection criteria constraints. These constraints get enforced on the unlabeled data points during the learning process. This ensures that the decision boundaries created are in line with the actual marginal distribution. At test time, these rejection criteria constraints also help in detecting and rejecting adversarial samples. The authors evaluated their approach on the ANIMAL, CIFAR, and PASCAL datasets in which super-class and sub-class concepts are defined. In contrast to this, our approach creates adversarial examples on concept labels in the CelebA, UTKFace, MNIST, and Fashion-MNIST datasets without relying on the underlying inheritance relationship or any other domain knowledge. We create adversarial examples using multiple attacked and protected classifiers by combining their loss function during the test time. Another approach involving an ensemble of classifiers is shown in [21]. The authors tackle the problem of adversarial robustness in an ensemble of classifiers trained on the same multi-class classification problem. They consider the following scenarios: 1) misclassification of a given concept (e.g., digit 0) by all the models in the ensemble setting, and 2) misclassification of a given concept by one model but correct prediction by other models in the ensemble set. In both scenarios, the authors focus only on one concept. In addition, the authors only focus on the misclassification of a given concept without guaranteeing to preserve the accuracy of the remaining concepts. In our work, unlike an ensemble of models learning the same concept, each of our models is trained to learn a different concept. Additionally, the goal of our multi-concept attack is to be able to attack concepts in

[1] We would like to stress that in this case, the privacy challenge occurs because the ML model predicts sensitive information. Hence, privacy-preserving ML techniques that output accurate ML models will not address this privacy challenge.

the "attack" set while protecting the accuracy of concepts in the "protect" set.

Perdomo and Singer [24] attacked multiple classifiers trained to learn a single concept; additionally, it solves a diametrically opposing zero-sum game between the attacker and the classifiers. Unlike their work, we are dealing with multiple classifiers trained to learn different concepts, attacking a subset meanwhile protecting the rest. Our attacker acts as both a competitor and a cooperator in game-theoretic terms.

Simultaneously attacking and protecting partially dependent concepts are contradictory by nature. It remains unknown whether existing state-of-the-art attacks can fulfill the attack-protect dual task, by simply changing the objectives and with added constraints, without canceling each other out during action and eventually fail the dual task. As a result, the potential social impact of the contagion effect on models trained for other learning tasks after a single model attack is overlooked. To the best of our knowledge, our work is the first to consider the adversarial learning problem where attacking and protecting multiple classifiers trained for different learning tasks is considered. Our perturbations in the L_p norm formulation are also bounded by the ϵ value, unlike the previous work where a successful multi-concept attack required large distortions. In the case where two concepts are correlated, when one is attacked, the other is inevitably impacted. Hence, attacking some concepts while protecting others is of great importance in real applications where instances may have multiple occasions of usage by various ML models.

3 PROBLEM DEFINITION

In a multi-concept classification setting, there's a one-to-many mapping between every instance and its concept set. Thus, the attacks proposed in this setting target datasets on which multiple learning tasks can be performed. Each successfully learned concept is a potential victim. The goal is to generate adversarial examples attacking a set of concept models without causing a significant drop in the accuracy of the protected concept set. Throughout this paper, we use the words *models* and *classifiers* interchangeably.

We assume our classifiers are binary unless otherwise specified. In addition, the classifiers are all assumed to be partially dependent in our problem setting. More rigorously speaking, the features contributing to the construction of the classifiers have to be partially dependent. If all classifiers are independent, attacking one will not affect the others; if they are strictly coupled, attacking some while protecting others would be impossible. Partial dependence is the necessary condition for the problem under study to be meaningful, and its practical reality is demonstrated in our empirical study.

3.1 Threat Model

We assume a white-box multi-concept attack scenario. Suppose there are T different learning tasks, with each classifier f_i learning one of the T concepts $F_i|_{i\in[1,T]}$. In a real world attack scenario presented in our experiments, we train four classifiers to learn four different concepts: *pretty*, *glasses*, *gender*, and *age* from a same celebrity dataset. We assume the adversary has complete access to the trained DNN models, including the architectures and the trained weights. The adversary's strength is only bounded by the allowed budget. The adversary does not interact with the learning system other than modifying a test instance by adding adversarial noise. In the above example, the adversary may choose to modify a celebrity's image to attack any subset of the four concepts, while keeping the rest intact, for example, attacking *gender* and *age*, but protecting *glasses* and *pretty*.

3.2 Beyond Simple Aggregation

We preview our empirical study on the aforementioned celebrity data to demonstrate that simply aggregating attacks on each classifier is not an effective attacking strategy against the whole. More importantly, the simple strategy cannot, by nature, protect the group of specified classifiers in the context of our threat model. In this demonstration, the naive attacker would simply record the aggregated perturbations against each classifier, and apply the average of the aggregated perturbation to corrupt the image in each iteration. Table 1 compares the native C&W [1] L_2 aggregated attack and PGD [20] L_∞ aggregated attack to our custom attack that attacks and protects desired classifiers simultaneously. As can been seen, the naive attacker cannot attack effectively, and often significantly damages the non-attacked classifiers. Our attack strategy works as designed: attacking and protecting specified classifiers successfully.

3.3 Problem Definition

Let $X \sim \mathcal{D}$ be the data set from a distribution $\mathcal{D}$ on which a set of learning models is trained. This results in $F = \{f_1(X), f_2(X), \ldots, f_n(X)\}$ where $f_i(X)$ is a decision function learned from X. Let x be an instance drawn randomly from $\mathcal{D}$, $y_{i\in[1,n]}$ be the label of x for the i^{th} learning task. The attacker's objective is to corrupt x by Δx so that $F_a = \bigcup_{k_a\in J} F_{k_a} \subset F$ is attacked, while $F_p = \bigcup_{k_p\in K} F_{k_p} \subset F|F_p \cap F_a = \emptyset$ is protected, with a minimum cost. Formally,

$$\begin{aligned} \min \quad & H(\Delta x) \\ s.t. \quad & \forall k_a \in J, \quad f_{k_a}(x+\Delta x) \neq y_{k_a} \\ & \forall k_p \in K, \quad f_{k_p}(x+\Delta x) = y_{k_p} \end{aligned} \tag{1}$$

where H is an objective function. J and K are the index sets of the attacked and the protected model sets, respectively.

Specific formulations of the problem defined in Eq. (1) fall into two general types, depending on the nature of the learning problems. When the learning problem is linear, attacking/protecting multiple linear decision functions can be formulated as a linear programming problem. We assume binary classification for simplicity (although multi-class concepts can be learned with one-against-all binary linear classifiers). For multi-classification problems, potential strategies include choosing the label with the largest loss derivative, loss grouping by correlation of labels, etc. We leave in-depth discussions on the multi-class formulation for future work.

For a linear problem, we show that there exists a linear programming solution. Suppose we are given m binary linear classifiers with coefficients denoted as $\mathbf{w}_{i\in[1,n]}$, the problem defined in Eq. (1) can be specified as finding the minimum-cost perturbation Δx to a given input $(x, y_{i\in[1,n]})$, where $y_i \in \{-1, +1\}$ is the label of x in the i^{th} problem space, such that only a subset J of the linear classifiers

Table 1: Comparing the naive attack to our custom attack (protecting the non-attacked) in the context of attack-protect dual model. Each row lists the accuracy of the classifier trained to learn the indexed concept. Attacked classifier sets are {*Glasses*}, {*Pretty, Glasses*}, and {*Age, Glasses, Gender*} and the corresponding accuracy after attack is underlined. Accuracy of the protected classifiers in our strategy is bolded.

		Attack: **Glasses**				Attack: **Pretty+Glasses**				Attack: **Age+Glasses+Gender**			
	Original	C&W L_2	OUR L_2	PGD L_∞	OUR L_∞	C&W L_2	OUR L_2	PGD L_∞	OUR L_∞	C&W L_2	OUR L_2	PGD L_∞	OUR L_∞
Pretty	85.4	85.4	**79.66**	68.18	**90.63**	46.73	62.54	1.11	47.83	85.7	**86.2**	71.6	**95.27**
Glasses	98.39	93.76	6.34	21.75	4.23	93.35	17.82	72.1	46.93	93.25	30.92	77.44	13.39
Gender	93.76	95.67	**96.07**	86.0	**93.76**	95.77	**98.39**	95.07	**99.3**	77.14	45.72	18.43	9.67
Age	88.42	88.42	**89.83**	64.75	**93.86**	88.12	**92.04**	85.2	**99.5**	52.27	42.7	5.44	15.71

outputs a false prediction for x' where $x' = x + \Delta x$:

$$\begin{aligned} min \quad & ||c \odot \Delta x||_p \\ s.t. \quad & \forall k_a \in J, \qquad y_{k_a} \cdot [\mathbf{w}_{k_a}(x+\Delta x) + b_{k_a}] \leq 0 \\ & \forall k_p \in [1,n]/J, \quad y_{k_p} \cdot [\mathbf{w}_{k_p}(x+\Delta x) + b_{k_p}] > 0 \end{aligned}$$

where c is the cost of modifying each feature, $\odot$ denotes element-wise product, and $||\cdot||_p$ denotes L_p norm, defined as:

$$||v||_p = (\sum_{i=1}^{n} |v_i|^p)^{\frac{1}{p}}.$$

This type of problem can be solved with linear programming.

When the learning problems on a data set are highly non-linear, each concept is learned independently with a complex non-linear classifier, for example, a deep neural network (DNN). The problem given in Eq. (1) is best formulated as a multi-concept attack problem that can be modeled as a multi-objective optimization problem. Given a set of classification functions $f_k(x)$ where $k \in \{1, \ldots, n\}$ for n concepts, the multi-objective problem is formulated to find a perturbation for instance x so that individual loss functions ℓ_a and ℓ_p are simultaneously maximized in a feasible region $\mathcal{X} \subset \mathbb{R}^n$:

$$\begin{aligned} \max \quad & L(\Delta x) = (\ell_a(\Delta x), \ell_p(\Delta x))^T \\ s.t. \quad & x + \Delta x \in \mathcal{X} \end{aligned}$$

where,

$$\ell_a(\Delta x) = \sum_{\forall k_a \in J} L(f_{k_a}(x+\Delta x), y_{k_a})$$

$$\ell_p(\Delta x) = \sum_{k_p \in [1,n]/J} L(f_{k_p}(x+\Delta x), -y_{k_p})$$

in which we attack a subset J of the concepts by maximizing the classification loss in ℓ_a, and protect the rest of the concepts by maximizing the "reverse classification" loss in ℓ_p. Note that the extended non-linear solution is also applicable to linear problems, simply replacing the loss of the non-linear classifiers with the loss of the linear classifiers. However, the final solution to a linear problem may not converge to the exact optimal of its closed-form solution, given that the multi-objective formulation is solved heuristically.

3.4 Optimization for Linear Classifiers

In this section, we present the solution to the multi-concept attack problem, in which learning problems are linear. We assume the learning models are linear classification models $f_{k_i}(x) = \mathbf{w}_{k_i}(x) + b_{k_i}$. An optimal modification to a given instance x is computed, assuming L_1, L_∞, and L_2 norms.

3.4.1 L_1 Norm Minimization. L_1 norm minimization is also known as the least absolute values method. Our multi-concept attack problem is defined as the following single objective constrained optimization problem:

$$\begin{aligned} \min_{i \in A} \quad & \sum_i |c_i \Delta x_i| \\ s.t. \quad & \forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a}(x+\Delta x) + b_{k_a}] \leq 0 \\ & \forall k_p \in K, \quad y_{k_p} \cdot [\mathbf{w}_{k_p}(x+\Delta x) + b_{k_p}] > 0 \end{aligned}$$

where A is the index set of the features that can be modified, J and K denote the attacked model set and the protected model set respectively. Modification to the i^{th} feature is denoted as Δx_i, and the corresponding cost is c_i. Note that y_{k_a} (y_{k_p}) is the label for a given x in the k_a^{th} (k_p^{th}) problem space. The problem is reduced to a linear program as follows:

$$\begin{aligned} \min \quad & \sum_{i \in A} t_i \\ s.t. \quad & \forall i \in A, \quad c_i \Delta x_i \leq t_i \\ & \forall i \in A, \quad -c_i \Delta x_i \leq t_i \\ & \forall i \in A, \quad t_i \geq 0 \\ & \forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a}(x+\Delta x) + b_{k_a}] \leq 0 \\ & \forall k_p \in K, \quad y_{k_p} \cdot [\mathbf{w}_{k_p}(x+\Delta x) + b_{k_p}] > 0 \end{aligned}$$

Note that if the goal is to attack the positive instances only, we simply relax the attack constraint from:

$$\forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a}(x+\Delta x) + b_{k_a}] \leq 0$$

to

$$\forall k_a \in J, \quad \mathbf{w}_{k_a}(x+\Delta x) + b_{k_a} \leq 0.$$

This allows to only attack positive samples without forcing false positives.

3.4.2 L_∞ Norm. For L_∞ norm minimization, the multi-concept attack problem is formulated as follows:

$$\begin{aligned} \min \quad & \max_{i \in A} |c_i \Delta x_i| \\ s.t. \quad & \forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a}(x+\Delta x) + b_{k_a}] \leq 0 \\ & \forall k_p \in K, \quad y_{k_p} \cdot [\mathbf{w}_{k_p}(x+\Delta x) + b_{k_p}] > 0 \end{aligned}$$

Similarly, we can reduce the above problem to a linear program:

$$\begin{aligned}
\min \quad & t \\
s.t. \quad & \forall i \in A, \quad c_i \Delta x_i \leq t \\
& \forall i \in A, \quad -c_i \Delta x_i \leq t \\
& t \geq 0 \\
& \forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a} \cdot (x + \Delta x) + b_{k_a}] \leq 0 \\
& \forall k_p \in K, \quad y_{k_p} \cdot [\mathbf{w}_{k_p} \cdot (x + \Delta x) + b_{k_p}] > 0
\end{aligned}$$

3.4.3 L_2 Norm. For L_2 norm minimization, the original multi-concept attack problem is defined as follows:

$$\begin{aligned}
\min_{i \in A} \quad & \sum_i |c_i \Delta x_i|^2 \\
s.t. \quad & \forall k_a \in J, \quad y_{k_a} \cdot [\mathbf{w}_{k_a}(x + \Delta x) + b_{k_a}] \leq 0 \\
& \forall k_p \in K, \quad y_{k_p} \cdot [\mathbf{w}_{k_p}(x + \Delta x) + b_{k_p}] > 0
\end{aligned}$$

The problem can be cast as a dual problem with a Lagrangian as:

$$\begin{aligned}
\max \quad & -\frac{1}{4} G c^{-1} (c^{-1})^T G^T + B \\
s.t. \quad & \lambda_{k_a \in J, k_p \in K} \geq 0
\end{aligned}$$

where,

$$G = \sum_{k_a \in J} \lambda_{k_a} y_{k_a} \mathbf{w}_{k_a} - \sum_{k_p \in K} \lambda_{k_p} y_{k_p} \mathbf{w}_{k_p},$$

$$B = \sum_{k_a \in J} \lambda_{k_a} k_{k_a} (\mathbf{w}_{k_a} x + b_{k_a}) - \sum_{k_p \in K} \lambda_{k_p} y_{k_p} (\mathbf{w}_{k_p} x + b_{k_p}),$$

where $\lambda_{k_a \in J}$ and $\lambda_{k_p \in K}$ are the Lagrangian multipliers.

3.5 Optimization for Non-linear Classifiers

In practice, we often encounter more complex problems in which the learning functions are highly non-linear. When the decision functions are non-linear, we can formulate the multi-concept attack problem as a multi-objective optimization problem, where the objectives are potentially in conflict.

In the multi-concept attack problem, we are given a set of n objectives, including maximization of classification losses for the attacked classifiers and minimization of classification losses for the protected classifiers, where n is the number of classifiers in consideration. Let the decision functions of the classifiers be

$$f_{i \in [1,n]}(x + \Delta x)$$

and $t \in \mathbb{R}$ be the optimal value, and $d \in \mathbb{R}^m$ be the common descent direction of the gradients. If it exists, we can find such a common descent direction by minimizing the first-order Pareto stationarity [4]:

$$\begin{aligned}
\arg\min_{d \in \mathbb{R}^m, t \in \mathbb{R}} \quad & t + \frac{1}{2}||d||^2 \\
s.t. \quad & \\
& \nabla f_i(x + \Delta x)^T d - t \leq 0, \forall i \in [1, n]
\end{aligned}$$

The dual of the above problem is [16]:

$$\begin{aligned}
\arg\min_{\lambda^n} \quad & || \sum_{i=1}^{n} \lambda_i \nabla f_i(x + \Delta x)||^2 \\
s.t. \quad & \lambda \in \Delta^n
\end{aligned}$$

where $\Delta^n = \{\lambda : \sum_{i=1}^n \lambda_i = 1, \forall i \in [1, n]\}$. For simplicity and efficiency, in this paper, we do not search for the optimal λ, but simply use a negative multi-gradient $g = -\sum_{i=1}^n \lambda_i \nabla f_i(x + \Delta x)$ with preset λ_i values. We have discussed the impact of λ in the discussions section at the end of the paper.

Our multi-objective optimization moves in the direction of common gradient descent g, guided by maximizing the summation of model losses of the classifiers in both the attacked and the protected concept sets. We calculate the loss of the attacked concept set for the ground truth label. For concepts in the protected set, the model losses are calculated for flipped labels. We model multi-objective optimization as follows:

$$\begin{aligned}
\max(& \frac{1}{M} \sum_{\forall i \in M} L(f_i(x + \Delta x), y_i) \\
& + \frac{1}{N} \sum_{\forall j \in N} L(f_j(x + \Delta x), label_flip(y_j)))
\end{aligned}$$

In this formulation, we have $M + N$ independent binary classifiers, each classifier learning an independent concept. We attack M concepts and preserve the remaining N concepts. The classification function $f_i(x)$ belongs to the attack concept set M, and $f_j(x)$ belongs to the protected concept set N. Here functions $f_i(x)$ and $f_j(x)$ learn concepts i and j respectively using the ground truth labels y_i and y_j. $label\_flip()$ is a function that generates the label different from the ground truth label for the classifiers in the protect concept set. Given all the possible labels for a given concept, we find a label at test-time with the maximum absolute difference of the model loss for the ground-truth label. The advantage of this formulation is that it can also be extended to multi-class scenarios. For binary classifiers, this translates to the negation of the ground truth label. In essence, we "mask" the labels for the protected classes, while generating adversarial examples for the attacked classifiers. Both actions are performed during test time. Due to label masking, we observe that the accuracy of the protected classifiers may increase compared to their original accuracy.

4 EXPERIMENTAL RESULTS

We demonstrate the effectiveness of our multi-concept attack techniques on several image datasets. In the linear case, we experimented with the MNIST dataset, selectively attacking a set of linear SVM classifiers with hinge loss. In the non-linear case, we used two more complex datasets: Celeb and UTKFace, selectively attacking a set of deep neural networks. We choose to use image data in the experiments because they can supply visual artifacts that demonstrate partial dependence of multiple learning problems defined on a single data set. In the two set of experiments, we provide the Shapley values to illustrate that important features (pixels) used to

learn different concepts overlap partially, confirming our hypothesis that attacking one concept would have significant impact on others learned from the same dataset.

4.1 Selective Attack against Linear Classifiers

Given a set of linear SVM classifiers, we tested the following scenarios: 1.) attack one classifier while protecting the rest; 2.) attack more than one classifier while protecting the rest. The concepts being attacked and protected were randomly chosen. The experiments were performed on the MNIST dataset [2]. We define three learning tasks on the MNIST data:

a.) learning even/odd digits (i.e., concept EVEN);
b.) learning digits ≥ 5 (i.e., concept ≥ 5);
c.) learning zero/non-zero digits (i.e., concept ZERO).

We transform the original learning problem into a multi-concept problem with three sub-tasks. For example, digit "0" can be labeled as 'zero', 'even', and 'not ≥ 5' simultaneously. We randomly select 200 samples from the independent test set to attack/protect the randomly selected concepts.

4.1.1 Results of L_1 Norm Minimization. Figure 1 illustrates the results of the L_1 attack with two attack targets: concepts "≥ 5" and "*ZERO*". The concept "*EVEN*" is protected. The accuracy of the ≥ 5 classifier dropped to less than 20% from 88.5%. The accuracy of the *ZERO* classifier dropped from 98% to less than 90%, a significant drop considering there is only 10% of digit '0' in the dataset. The recall values of both classifiers dropped to zero, while the accuracy/recall of the protected classifier slightly improved. These results confirmed our hypothesis that multi-concepts can be attacked simultaneously while selectively protecting the others.

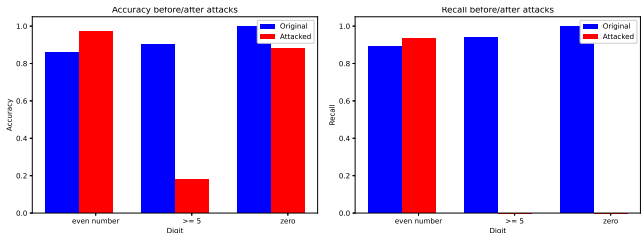


Figure 1: Multi-concept attack on MNIST with L_1 norm minimization. The concepts attacked are ≥ 5 and *ZERO* and the protected concept is *EVEN*.

We also investigate how the attacks influence the predictions of the classifiers when modifying the input images. We compute the Shapley values of each pixel in the original images and the images after the attack using SHAP [18]. SHAP (SHapley Additive exPlanations) is a game-theoretic approach to explaining the decision of a machine learning model by providing the Shapley values from game theory. SHAP values represent each pixel's contribution to the decision output of the model. Small SHAP values indicate a low contribution to the prediction of the concept. Figure 2 shows the images after the attack, the SHAP values of the pixels in the original images, and the SHAP values in the images after the attack.

[2]http://yann.lecun.com/exdb/mnist/

We show the SHAP values of the positive class. Hence, pixels with larger SHAP values have a significant contribution to the positive prediction by the model, that is, a prediction of ZERO by the classifier trained to learn the *ZERO* concept. The smaller the SHAP values, the darker the pixels in the SHAP-value images, and hence the smaller contribution of the pixels in the images of the digits to the prediction of the concept "ZERO". Likewise, the brighter the SHAP images, the more contribution of the pixels to the prediction of "ZERO".

From Figure 2, we can clearly observe the following:

- The SHAP values of the pixels in the original "0" images (red boxes) are much higher (brighter) than the pixels in images of other digits; while the SHAP values of the "0" pixels are much smaller (darker) in the attacked "0" images (red boxes). Also, notice the dark SHAP images (representing non-zero digits) on the original images (middle panel) turned out much lighter in the SHAP images (right panel) after the attack.
- The attacked images of "0" (the left panel in Figure 2) are made more perceptible to human eyes while compromising the classifier ZERO. This calls for more precautions in situations where human-AI cooperation is desired. Misunderstanding of AI behavior may result in upsetting consequences, especially in mission-critical applications such as self-driving vehicles.

4.1.2 Results of L_∞ Norm Minimization. Figure 3 illustrates the results of the L_∞ attack. The "≥ 5" classifier and the "ZERO" classifier were attacked, and the "*EVEN*" classifier was protected. As in the case of L_1 attack, the L_∞ attacks improved the overall accuracy and recall of the protected classifier while slashing the accuracy and recall of the attacked classifier to nearly zero.

Figure 4 shows the images after the attack. To human eyes, the L_∞ attacks are more aggressive than the L_1 attacks when modifying the images. The SHAP images were similar as in the case of the L_1 attacks.

4.1.3 Results of L_2 Norm Minimization. Figure 5 illustrates the results of the L_2 attack. The attacked concepts were *EVEN* and ≥ 5, and the protected concept was *ZERO*. As in the studies of the L_1 and L_∞ norm minimization, the L_2 attacks improved the overall accuracy of the protected classifier while tanking the performance of the classifiers under attack.

As shown in Figure 6, the L_2 attacks attempt to add noise more evenly to the images than the L_1 and the L_∞ attacks.

4.2 Selective Attack against Non-Linear Classifiers

In this section, we selectively attack non-linear classifiers trained on more complicated image data. Each classifier models a unique concept such as *gender*, *age*, *eyeglasses*, and *pretty*. We refer to an attack scenario in short as $Attack : \{X\}$, $Protect : \{Y\}$, where X is a set of classifiers that are attacked, Y is a set of classifiers that are **protected**. Classifiers are either binary or *multi-class* [3]. We show that our custom attack reduces the drop of accuracy, in some

[3]While we show promising preliminary results in this paper, we leave complex versions of *multi*-attacked and *multi*-protected classifiers for future work

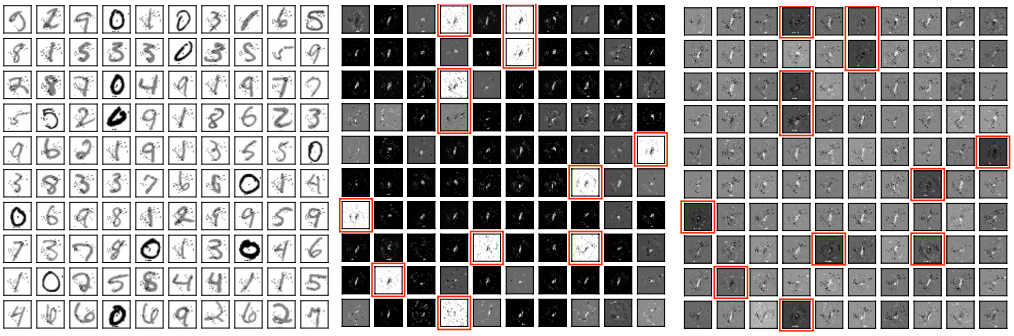


Figure 2: Multi-concept attack on the MNIST images with L_1 norm minimization with two concepts attacked. The left plot shows the attacked images, the middle plot shows the SHAP values of the pixels in the original images, and the right plot shows the SHAP values of pixels in the same images after the attack. The predictor is the "ZERO" classifier.

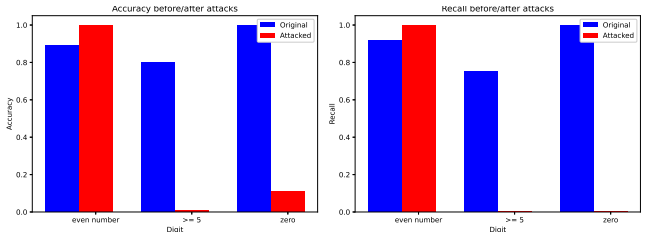


Figure 3: Multi-concept attack on MNIST with L_∞ norm minimization. The concepts attacked are ≥ 5 and *ZERO* and the protected concept is *EVEN*.

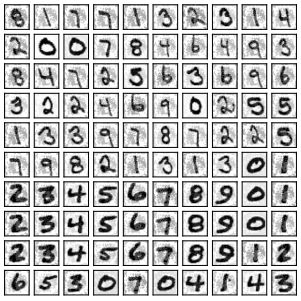


Figure 4: MNIST images attacked with L_∞ norm minimization. The concepts attacked are ≥ 5 and *ZERO* and the protected concept is *EVEN*.

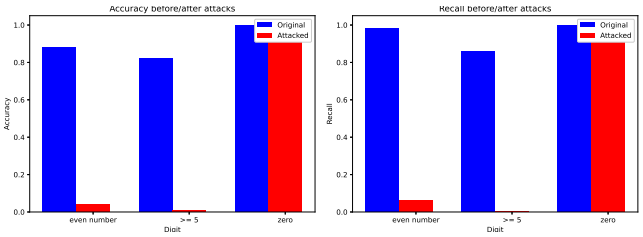


Figure 5: Multi-concept attack on MNIST with L_2 norm minimization. The concepts attacked are ≥ 5 and *EVEN* and the protected concept is *ZERO*.

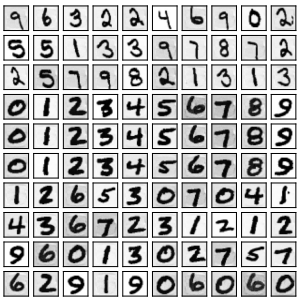


Figure 6: MNIST images attacked with L_2 norm minimization. The concepts attacked are *EVEN* and ≥ 5 and the protected concept is *ZERO*.

cases, even improves the accuracy of the protected classifiers, while effectively lowering the accuracy of the attacked classifiers.

4.2.1 Datasets and Learning Models. We chose the CelebA [17] and the UTKFace [37] datasets for this experiment. For the CelebA dataset, we trained four binary classifiers to predict *Gender*, *Glasses*, *Age* and *Pretty*. To train the *Age* binary classifier, an example x is either labeled as '1' or '0' as follows:

$$\begin{cases} 0 & \text{younger than 30} \\ 1 & \text{else} \end{cases}$$

We also experiment when *Age* is multi-class '0', '1', and '2':

$$\begin{cases} 0 & \text{younger than 25} \\ 1 & \text{between 25 and 36} \\ 2 & \text{else} \end{cases}$$

For UTKFace, we trained binary classifiers for *Age*, *Gender*, and *Ethnicity*. We selected 993 images from *CelebA* and 1000 images from *UTKFace* to test our multi-concept attack strategy. Test images are equally distributed in each class.

We used a pre-trained ResNet50 [6] model architecture with a custom layer concatenating Average Pooling and Max Pooling layers, a Flatten layer, and two blocks of Batch normalization [9], Dropout [7], Linear and ReLU layers. We employed fastai [8] to train our models. Throughout this section, we interchangeably use 'GE', 'GL', 'AG', 'PR' and 'ET' to refer to the concepts of *Gender*, *Glasses*, *Age*, *Pretty* and *Ethnicity*.

4.2.2 Baselines. We compare the results of our multi-concept attack with the following baselines:

- No attack;
- Projected Gradient Descent (PGD) [20] L_2 and L_∞ attacks;
- Carlini & Wagner (C&W) [1] L_2 attacks;
- Deepfool (DF) [23] L_2 and L_∞ attacks;
- Fast Gradient Sign Method (FGSM) [5] L_2 and L_∞ attacks.

We use the Foolbox [25] and TorchAttacks [11] libraries to implement the baseline PGD, FGSM, DF, and C&W attacks. PGD is one of the strongest attacks, providing a good benchmark to evaluate our multi-concept attack. While "weaker" attacks such as FGSM and DF are expected to have less adverse impact on the non-attacked classifiers, we show that this is not the case in the subsequent sections. We also compared our attack strategy with the simple idea of averaging the losses of all classifiers. We show that our custom attack is more sophisticated than a simple average of losses.

Hyperparameter tuning for different attack methods is investigated. Hyperparameters offering strongest attacks are used in the experiment.

4.2.3 Transferability of Attacks. Existing attacks, when successfully defeating the classifier trained for one learning task, inevitably weaken the accuracy of classifiers trained (on the same dataset) for other learning tasks. In other words, adversarial attack against one classifier is transferable to other classifiers even if the latter are trained for different tasks. Transferability is clearly demonstrated in our empirical study, presented in Tables 2, 3, 4 and 5. For example, in Table 2, we show that when the *Gender* (GE) classifier trained on the *CelebA* data was successfully attacked by PGD L_2, dropping its accuracy to 3.73% from 93.76%, the other three classifiers trained for learning the *Age*, *Glasses*, and *Pretty* concepts were also devastated. The accuracy of the *Glasses* classifier dropped to 88.42%, from 98.69%; the accuracy of the *Age* classifier dropped from 85.7% to 64.05%. In Table 3, we observed similar results for the UTKFace data. PGD attacks on the *Gender* and *Age* classifiers also damaged the *Ethnicity* classifier. This observation is true for weaker attacks such as FGSM and DF. Hence, without explicit protection, attack is transferable.

4.2.4 Results of L_2 Norm Minimization. In our L_2 *attack-protect dual* strategy, choosing a step size of 0.8 can quickly lead to the maximum attack success rate within 200 iterations. The λ value is set to one as in [4]. Tables 2 and 3 illustrate the results of our L_2 attack-protect dual for the CelebA and the UTKFace datasets. For comparison, the *Original Accuracy* and the accuracy when attacked by the baselines are listed at the top. The naive idea of attacking multiple by averaging the losses of all classifiers is also illustrated.

Table 2: Accuracy of the CelebA classifiers after L_2 norm attacks. *Attack* and *Protect* are the attacked and protected classifier groups. GE, GL, AG, PR are used to denote *Gender*, *Glasses*, *Age* and *Pretty* classifiers, respectively.

	Attack	Protect	GE	GL	AG	PR
Original Accuracy	-	-	93.76%	98.69%	85.7%	81.77%
PGD L_2	GE	-	3.73%	88.42%	64.05%	59.62%
	GL	-	82.18%	3.63%	64.25%	65.46%
CW L_2	GE	-	5.74%	50.15%	44.51%	41.59%
	GL	-	54.98%	3.63%	31.42%	30.92%
FGSM L_2	GE	-	36.66%	91.24%	70.29%	74.12%
Average All Losses	{GE,GL}	-	16.52%	13.8%	63.65%	62.54%
	{GE,GL,AG}	-	30.51%	7.85%	22.56%	54.98%
Our Aattack-Protect Dual Strategy:						
1Attack-1Protect	GE	GL	15.31%	**99.60%**	69.59%	67.07%
	GL	GE	**99.09%**	17.22%	67.67%	69.79%
1Attack-2Protect	GE	{GL,AG}	17.32%	**99.3%**	**91.74%**	70.69%
	GE	{GL,PR}	11.78%	**99.19%**	74.22%	**76.33%**
	GL	{GE,AG}	**96.98%**	10.17%	**92.25%**	73.21%
	GL	{GE,PR}	**97.78%**	8.36%	72.21%	**80.46%**
2Attack-1Protect	{GE,AG}	GL	29.41%	**99.9%**	29.31%	60.73%
	{GE,PR}	GL	28.1%	**100.0%**	64.65%	48.54%
	{GL,AG}	GE	**99.09%**	26.59%	39.68%	67.17%
	{GL,PR}	GE	**99.6%**	28.2%	66.47%	58.71%
	{GE,GL}	AG	47.33%	28.0%	**97.28%**	72.0%
	{GE,GL}	PR	40.48%	21.85%	71.4	**88.52%**
2Attack-2Protect	{GE,PR}	{GL,AG}	29.31%	**99.8%**	**93.15%**	57.0%
	{GL,PR}	{GE,AG}	**98.39%**	17.82%	**92.04%**	62.54%
3Attack-1Protect	{GE,AG,PR}	GL	36.35%	**99.9%**	36.46%	47.83%
	{GL,AG,PR}	GE	**99.5%**	35.75%	44.11%	63.75%
	{GE,GL,AG}	PR	45.72%	30.92%	42.7%	**86.2%**
1Attack-3Protect	GE	{GL,AG,PR}	13.6%	**98.49%**	**91.24%**	**75.93%**
	GL	{GE,AG,PR}	**96.07%**	6.34%	**89.83%**	**79.66%**
	PR	{GE,AG,GL}	**96.27%**	**99.09%**	**80.56%**	46.73%

In Table 2, attacking *m* classifiers while protecting *n* others is denoted as *mAttack-nProtect* in our dual strategy. For example, under *1Attack-1Protect*, when the Glasses classifier is attacked and the **Gender** classifier is protected, our L_2 strategy protects the accuracy of the **Gender** classifier (99.09%). This is a significant improvement compared to the accuracy of 82.18% after the PGD attack on Glasses. It is even slightly better than the original accuracy of 93.76%. Similarly, under *1Attack-1Protect* (Attack:Gender, Protect:**Age**) in Table 3, the accuracy of **Age** fared better when protected using our strategy, compared to the PGD L_2 attack on the Gender classifier. The accuracy of 95% after our custom attack is much better than the accuracy of 82.7% after the PGD Attack on Gender. With explicit protection in our strategy, there is no significant loss in the attack accuracy of the protected classifiers compared to regular PGD attacks.

We also observed that we retained the accuracy of the protected classifiers in both the *1Attack-2Protect* and the *2Attack-1Protect* scenarios. In the case of the *1Attack-2Protect* scenario, for example,

Table 3: Accuracy of the UTKFace classifiers after L_2 norm attacks. *Attack* and *Protect* are the attacked and protected classifier groups. GE, AG, ET are used to denote *Gender*, *Age* and *Ethnicity* classifiers respectively.

Original Accuracy		PGD		CW		DF		FGSM		Avg_loss		**Our L_2 Attack** 1Attack-1Protect		1Attack-2Protect		2Attack-1Protect	
Attack	-	GE	AG	GE	AG	GE	AG	GE	AG	{GE,AG}	{GE,AG,ET}	GE	AG	GE	AG	{GE,ET},	{AG,ET}
Protect	-	-	-	-	-	-	-	-	-	-	-	AG	GE	{AE,ET}	{GE,ET}	AG	GE
GE	99%	33.8%	80.4%	0.0%	50.00%	82.2%	84.7%	47.8%	82.1%	44.8%	56.4%	47.6%	**96%**	44.2%	**93.9%**	56.5%	**94.8%**
AG	97.2%	82.7%	40.3%	49%	0.10%	87.6%	85.1%	82.3%	46%	54.6%	49.5%	**95%**	53.9%	**94.4%**	48.2%	**95.3%**	64.1%
ET	100%	80%	79.2%	51.50%	36.60%	87.7%	87.7%	82.7%	81.9%	80%	44.6%	79.7%	79.6%	**96%**	**95.3%**	44.1%	46.4%

attacking Gender and protecting {**Age**, **Glasses**}, our dual strategy successfully protected the **Age** and the **Glasses** classifiers compared to the baseline PGD L_2. The accuracies of the **Age** and the **Glasses** classifiers were 91.74% and 99.3% after our custom attack, compared to the accuracy of 64.05% and 88.42% after the PGD attack against the Gender classifier. We observed similar results in the *2Attack-1Protect* scenario, where the **Age** classifier is sufficiently protected (97.28% after our custom attack versus 64.25% after the PGD attack on the Glasses classifier). In both cases, the accuracy of the protected classifier is better than the original accuracy after our custom attack. Table 2 also lists additional results of other *attacking m and protecting n* scenarios for the *CelebA* data. Our results are consistent for any combination of the classifiers in the attacked group and the protected group.

Our results are also consistent for the *UTKFace* data as shown in Table 3. In the *1Attack-2Protect* scenario, for example, attacking Gender and protecting {**Age**, **Ethnicity**}, the accuracy of both **Age** and **Ethnicity** was preserved. The accuracy of **Age** was 94.4% after our custom attack. This is in contrast to the accuracy of 40.3% after the PGD L_2 attack against Gender, which dropped from 97.2% before the attack. Similarly, the accuracy of *Ethnicity* was 96% after our custom attack, versus 80% after the PGD attack on Gender, with the original accuracy being 100%.

We observed that even though "weaker attacks" such as FGSM and DF were better than PGD in preserving the accuracy of the non-attacked classifiers, only our custom attacks were able to truly protect those classifiers. For example, consider the FGSM attack against Gender in Table 2. The accuracy of the *Glasses* classifier dropped from 98.69% to 91.24% after the FGSM attack. Meanwhile, the accuracy of the **Glasses** classifier was kept at 99.6% in one of our *1Attack-1protect* scenarios—attacking Gender and protecting **Glasses**, compared to 91.24% in the case of FGSM. This applies to other non-attacked classifiers which are actively protected as a part of the protected group in our custom multi-concept attack scenarios.

4.2.5 Results of L_∞ Norm Minimization. In our L_∞ *attack-protect dual* strategy, a smaller step size of 0.06 can quickly lead to the maximum attack success rate within 200 iterations. λ is still set to one as suggested in [4]. Tables 4 and 5 show that in the case of L_∞ attack, our custom attack demonstrates great attack strength against the attacked classifier group while protecting the accuracy of the protected group.

As shown in Table 4, in one of the *1Attack-1Protect* attack scenarios where we attack Gender and protect **Age**, our custom attack strength was comparable to the PGD L_∞ attack against Gender (7.5% versus 7%). It also successfully preserved the accuracy of the **Age** classifier compared to its corresponding accuracy for the PGD L_∞ attack against **Age** (80.9% versus 50%). We also observed similar results in the scenario where we attack Gender and protect **Glasses**. The accuracy of **Glasses** was 82.28% after our custom attack, compared to 50.45% on adversarial examples generated with the PGD L_∞ attack on Gender. We protected the **Glasses** classifier without causing a significant drop in the attack strength against the Gender classifier. The accuracy of Gender dropped to 41.39% using our custom attack, versus 40.79% for the PGD L_∞ attack against Gender.

We obtained similar results for other attacked/protected classifier combinations. In one of the *1Attack-2Protect* scenarios in which

Table 4: Accuracy of the CelebA classifier after L_∞ norm attacks. *Attack* and *Protect* are the attacked and protected concepts. GE, GL, AG, PR are *Gender*, *Glasses*, *Age* and *Pretty* classifiers respectively.

	Attack	Protect	GE	GL	AG	PR
Original_Accuracy	-	-	93.76%	98.69%	85.7%	81.77%
PGD L_∞	GE	-	0%	50.45%	40.79%	35.65%
	GL	-	51.86%	0.2%	43.2%	68.08%
FGSM L_∞	GE	-	46.42%	50.55%	49.55%	68.18%
	GL	-	50.35%	50.35%	43.71%	68.18%
Average All Losses	{GE,GL}	-	0.0%	13.7%	41.09%	51.76%
	{GE,GL,AG}	-	4.03%	10.27%	0.7%	46.83%
Our Custom Attack:						
1Attack-1Protect	GE	GL	0%	**82.28%**	41.39%	52.06%
	GL	GE	**100%**	12.49%	58.61%	68.69%
1Attack-2Protect	GE	{GL,AG}	0.0%	**72.31%**	**98.79%**	59.01%
	GE	{GL,PR}	0.0%	**71.9%**	51.66%	**79.05%**
	GL	{GE,AG}	**97.28%**	7.15%	**99.7%**	74.82%
	GL	{GE,PR}	**99.7%**	6.14%	65.26%	**89.53%**
2Attack-1Protect	{GE,AG}	GL	3.42%	**88.52%**	0.91%	46.93%
	{GE,PR}	GL	0.0%	**93.25%**	40.38%	30.21%
	{GL,AG}	GE	**99.9%**	34.74%	2.32%	54.98%
	{GL,PR}	GE	**100.0%**	39.98%	55.59%	37.06%
	{GE,GL}	AG	10.88%	38.47%	**100.0%**	64.75%
	{GE,GL}	PR	1.61%	42.3%	64.35%	**98.59%**
2Attack-2Protect	{GE,PR}	{GL,AG}	1.01%	**54.68%**	**99.3%**	38.17%
	{GL,PR}	{GE,AG}	**99.3%**	46.93%	**99.5%**	47.83%
3Attack-1Protect	{GE,AG,PR}	GL	6.34%	**93.76%**	1.61%	36.15%
	{GL,AG,PR}	GE	**100%**	8.56%	11.48%	36.25%
	{GE,GL,AG}	PR	9.67%	13.39%	15.71%	**95.27%**
1Attack-3Protect	GE	{GL,AG,PR}	0.0%	**96.68%**	**89.53%**	**71.2%**
	GL	{GE,AG,PR}	**93.76%**	4.23%	**93.86%**	**90.63%**
	PR	{GE,GL,AG}	**92.45%**	**92.95%**	**87.71%**	7.96%

Table 5: Accuracy of UTKFace concepts after L_∞ norm attacks. Attack and Protect are the attacked and protected concepts. GE, AG, ET are *Gender*, *Age* and *Ethnicity* classifiers respectively.

	Original Accuracy	PGD		Deepfool		FGSM		Avg_loss		**Our L_∞ Attack** 1Attack-1Protect		1Attack-2Protect		2Attack-1Protect	
Attack	-	GE	AG	GE	AG	GE	AG	{GE,AG}	{GE,AG,ET}	GE	AG	GE	AG	{GE,ET},	{AG,ET}
Protect	-	-	-	-	-	-	-	-	-	AG	GE	{AE,ET}	{GE,ET}	AG	GE
GE	99%	7%	50.2%	82.1%	84.6%	47.1%	50%	7.9%	3.6%	7.5%	**95.9%**	6.5%	**97.4%**	3.6%	**95.4%**
AG	97.2%	50%	30%	86.7%	85%	50%	50%	25.6%	21.3%	**80.9%**	23.4%	**61.4%**	21.1%	**83.7%**	45.8%
ET	100%	58.5%	58.7%	87.6%	87.6%	57%	50.9%	57.1%	1.8%	57%	57.2%	**99.6%**	**99.9%**	1.7%	1.2%

we attack Glasses and protect {**Gender**, **Age**}, as shown in Table 4, we successfully protected the **Gender** and the **Age** classifiers in the protected group. The accuracy after the custom attack is better than the original accuracy (97.28% versus 93.76% for the **Gender** classifier, and 99.7% versus 85.7% for the **Age** classifier). At the same time, this increase in defense did not come at the cost of the attack strength against the Glasses classifier. The attack outcome was comparable to the regular PGD L_∞ attack against Glasses (7% versus 0%). This observation is consistent with the **Pretty** classifier shown in Table 4. In the *1Attack-2Protect* scenario where we attack Gender and protect {**Glasses**, **Pretty**}, not only the **Pretty** classifier was successfully protected without a significant drop in accuracy (79.05% versus 81.77% originally), but the attack against Gender was as strong as the PGD L_∞ attack against Gender.

4.2.6 Attacking adversarially trained classifiers. We also investigated attacking adversarially trained classifiers [19], that is, classifiers trained with augmented adversarial examples in each batch. We could not adversarially train the *Pretty* and *Gender* classifiers to obtain reasonable accuracy on the original *CelebA* images using the ResNet50 structure. Therefore, for this experiment, we trained and adversarially trained DNN classifiers with the pre-trained MobileNet [26], replacing the top fully connected layer to match the binary output. Table 6 shows the validation accuracy of the (*Gender*, *Glasses*, *Pretty*, *Age*) classifiers and their adversarially trained counterparts, and the single-model PGD attack results. As expected, the adversarially trained models are less accurate on the original images, but are relatively more resilient to adversarial attacks.

Table 6: Accuracy of DNNs & adversarially trained DNNs before/after single-model PGD attacks.

	DNN	GD Attack DNN	Adv. DNN	PGD Attack Adv. DNN
Gender	98.85	0.00	97.29	42.57
Glasses	99.60	0.75	93.04	6.96
Pretty	81.58	0.00	76.35	47.99
Age	88.13	0.00	79.66	25.34

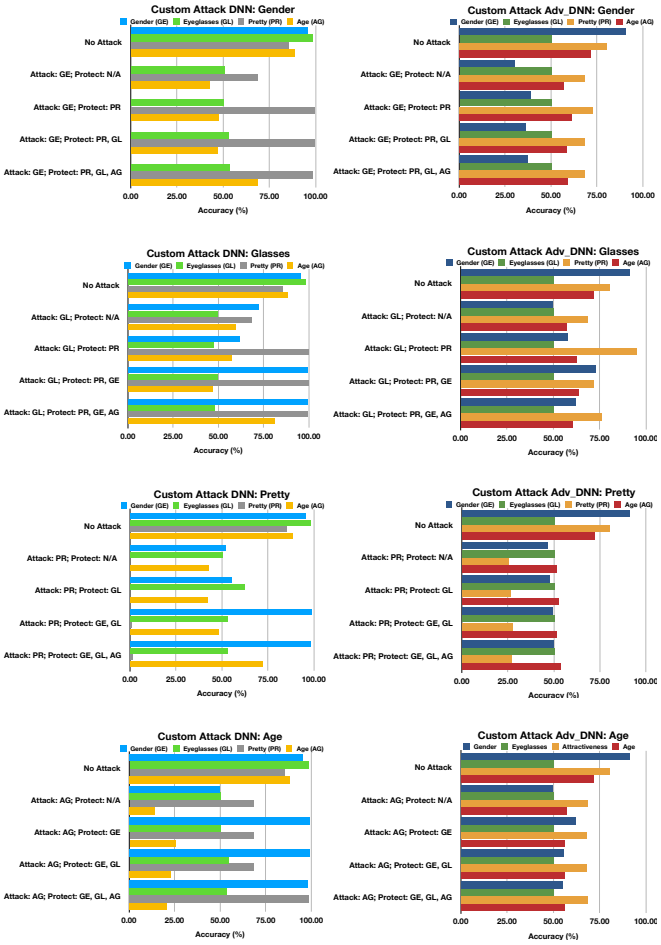


Figure 7: PGD (L_∞) Attack against *Gender*, *Glasses*, *Pretty*, and *Age* (left plots) and their adversarially trained counterparts (right plots).

Without protection, all classifiers are vulnerable to attacks against any single classifier, as shown in Figure 7 marked as **Protect: N/A**. The plots on the left show the custom attack results of the regularly trained DNNs, and on the right are the adversarially trained. In general, adversarially trained models are relatively more resilient to the attacks and also resilient to the protection. For the regularly trained models, our protection can not only prevent catastrophic accuracy drop to the protected classifiers, but sometimes improves their accuracy. This "over-protection" rarely occurs with the adversarially trained models. The protection was present most of the time, but it seldom went overboard. The attack was also less severe across all four concepts. We also identified one defect of the adversarially trained models: their unstable predictive accuracy. For

example, we thoroughly trained the adversarial "Glasses" model to achieve 93.04% validation accuracy on the original images, but can only obtain 50.35% accuracy on the original 993 images selected for attacks.

4.2.7 Extending custom attacks to the multi-classification scenario. In addition to binary problems, we show that our custom attack can be extended to the multi-classification settings for both L_2 and L_∞ norms. We train a multi-class *Age* classifier with three possible labels '0', '1' and '2' in our experiments, and consider the following scenarios:

1.) *1Attack-1Protect* with *Age* in the protected group;
2.) *1Attack-1Protect* with *Age* in the attacked group;
3.) *1Attack-2Protect* with *Age* in the protected group
4.) *1Attack-2Protect* with *Age* in the attacked group;
5.) *2Attack-1Protect* with *Age* in the protected group.

We compare our results with the corresponding PGD L_2 and L_∞ attacks in Table 7. We can see that in scenarios 2 and 4, our custom attacks were as strong as the PGD L_2 and L_∞ attacks against the Age multi-class classifier. In scenario 1, *1Attack-1Protect* L_2 attack with the Age classifier in the attacked group and the **Gender** classifier in the protected group, the accuracy of the **Gender** classifier was 95.4%. In contrast, the PGD L_2 Attack against Age reduced the accuracy of **Gender** from 99.0% to 81.1%. In scenarios 1, 3, and 5 where we have **Age** in the protected group, we observed that, compared to the regular PGD L_∞ attack, our custom attack preserved the accuracy of all other classifiers in the protected group. In scenario 3, *1Attack-2Protect* L_∞ attack with **Age** and **Ethnicity** in the protected group and Gender in the attacked group, the accuracy of the multi-classifier *Age* and the **Ethnicity** classifier is 47.1% and 99.4% respectively, compared to the accuracy of 43.8% and 58.5% in the case of PGD L_∞ attack on Gender.

4.2.8 The impact of λ values. We also examine the impact of different λ values on our custom attacks and compare them with the preset λ_i value of 1 in Table 8. We have λ value pairs (λ_1, λ_2), where λ_1 is the value of λ for the attacked classifier group, and λ_2 is the λ value for the protected classifier group. We see that tuning the λ values based on the attacker's needs can strengthen the attack against the attacked classifiers. It can also make the adversarial examples less influential on the classifiers in the protected group. For example, in the *1Attack-2Protect* L_2 attack scenario with Gender in the attacked group and **Age** and **Ethnicity** in the protected group, the λ value pair of (0.6, 0.4) is best suited if the main objective of the attacker is to maximize the attack strength. However, for the same example, the λ value of (0.2, 0.8) is better suited if, in addition to the attack capabilities, the attacker also wants minimum degradation in the accuracy of the models in the protected group. For the λ pair of (0.2, 0.8), the accuracy of **Age** and **Ethnicity** classifiers is 96.1% and 98.3%, respectively.

4.2.9 Impact on Protection As Attacks Unroll. We explore the accuracy of the protected **Gender** classifier during the process of L_2 attack, as shown in Figures 8. The accuracy of **Gender** for the *1Attack-1Protect*, *1Attack-2Protect*, and *2Attack-1Protect* scenarios is compared with the original accuracy and the accuracy after PGD L_2 attacks. For the PGD attack and our attack scenarios, we chose to attack the Age classifier. We show that, compared to the PGD attack, our multi-concept attack consistently achieved better accuracy with the protected classifier **Gender**. We also observed that the custom attacks have successfully protected the **Gender** classifier, with an accuracy close to the original when there is no attack.

Table 7: Accuracy of UTKFace after L_∞ and L_2 attacks in the multi-class setting. *Attack* and *Protect* are the attacked and protected classifiers. GE and ET are *Gender* and *Ethnicity* binary classifiers respectively. AG is the multi-class *Age* classifier.

	Attack	Protect	GE	AG	ET
Original_Accuracy			99.0%	93.4%	100%
L_2 norm attack					
PGD	GE	-	33.8%	83.5%	80%
	AG	-	81.1%	8.2%	79.3%
1Attack-1Protect	GE	AG	46.3%	**87.1%**	79.8%
	AG	GE	**95.4%**	16.0%	77.4%
1Attack-2Protect	GE	{AG,ET}	43.2%	**88.2%**	**96.1%**
	AG	{GE,ET}	**95.8%**	23.8%	**95.3%**
2Attack-1Protect	{GE,ET}	AG	55.0%	**86.7%**	43.4%
L_∞ norm attack					
PGD	GE	-	7%	43.8%	58.5%
	AG	-	49.9%	12.8%	55.5%
1Attack-1Protect	GE	AG	6.6%	**55.3%**	56.6%
	AG	GE	**95.4%**	14.2%	57.2%
1Attack-2Protect	GE	{AG,ET}	6.4%	**47.1%**	**99.4%**
	AG	{GE,ET}	**96.1%**	23.8%	**99.5%**
2Attack-1Protect	{GE,ET}	AG	4.5%	**55.0%**	1.6%

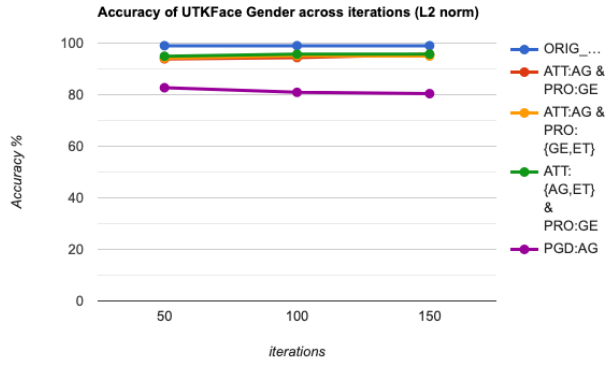


Figure 8: The accuracy of the protected Gender classifier during L_2 attacks. The accuracy of Gender is retained in our custom attacks, while the PGD attack on Age lowers the accuracy of the Age classifier.

We show the comparison of accuracy of the non-attacked classifiers given our custom L_∞ attack in Figure 9. In all these scenarios, only Gender belongs to the attacked group. The remaining classifiers are either in the protected group or independent. We observe

Table 8: Impact of λ values on the accuracy of UTKFace after L_2 and L_∞ attacks. *Attack* and *Protect* are the attacked and protected classifiers. GE, AG and ET are *Gender*, *Age* and *Ethnicity* classifiers, respectively.

			λ value pairs														
			(0.2, 0.8)			(0.4, 0.6)			(0.6, 0.4)			(0.8, 0.2)			(1, 1)		
	Attack	Protect	GE	AG	ET	GE	AG	ET	GE	AG	ET	GE	AG	ET	GE	AG	ET
L_2 norm custom attacks																	
1Attack-1Protect	GE	AG	60.9%	**96.3%**	79.1%	47.3%	**95.9%**	78.8%	36.5%	**95.8%**	78.6%	34.2%	**93.4%**	78.2%	47.6%	**95%**	79.7%
	AG	GE	**97.2%**	66.3%	78.7%	**97.8%**	52.0%	79.5%	**96.8%**	42.2%	79.3%	**94.0%**	39.6%	79.2%	**96%**	53.9%	79.6%
1Attack-2Protect	GE	{AG, ET}	57.6%	**96.1%**	**98.3%**	43.4%	**94.8%**	**98.0%**	36.1%	**94.1%**	**95.9%**	35.6%	**92.7%**	**91.2%**	44.2%	**94.4%**	**96%**
	AG	{GE,ET}	**95.9%**	61.5%	**98.2%**	**95.8%**	47.7%	**97.5%**	**94.4%**	39.5%	**95.7%**	**91.2%**	40.5%	**92.1%**	**93.9%**	48.2%	**95.3%**
L_∞ norm custom attacks																	
1Attack-1Protect	GE	AG	4.6%	**93.4%**	56.8%	4.2%	**79.3%**	57.0%	5.8%	**62.9%**	57.3%	6.1%	**52.8%**	57.1%	7.5%	**80.9%**	57%
	AG	GE	**96.1%**	49.1%	57.2%	**96.4%**	43.1%	57.1%	**97.7%**	27.3%	57.2%	**95.9%**	11.3%	57.4%	**95.9%**	23.4%	57.2%
1Attack-2Protect	GE	{AG, ET}	3.9%	**61.6%**	**100.0%**	2.1%	**55.4%**	**99.7%**	5.5%	**52.4%**	**97.9%**	6.2%	**50.4%**	**82.7%**	6.5%	**61.4%**	**99.6%**
	AG	{GE,ET}	**98.8%**	49.1%	**99.9%**	**99.1%**	46.1%	**100.0%**	**98.0%**	31.2%	**99.3%**	**89.7%**	9.6%	**91.6%**	**97.4%**	21.1%	**99.1%**

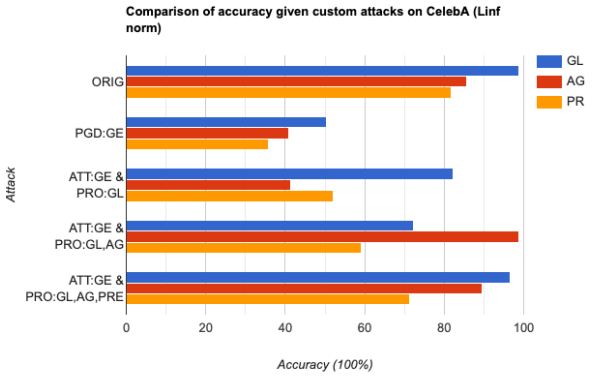
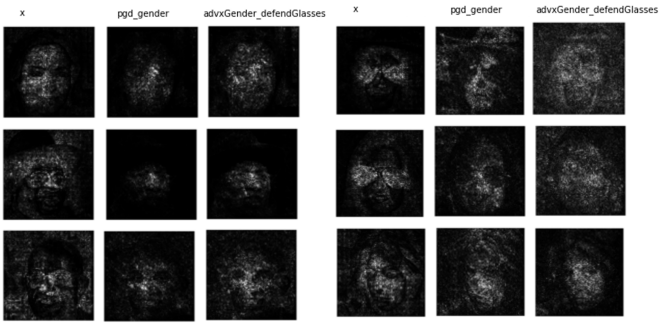


Figure 9: Comparison of accuracy given custom L_∞ attack scenarios on CelebA. GE, GL, AG and PR refer to *Gender*, *Glasses*, *Age* and *Pretty*, respectively.

that all custom attack scenarios deliver better accuracy for the non-attacked group, compared to PGD L_∞ attack on Gender. This observation holds as we increase the number of classifiers in our protected group. There seems to be no fixed upper limit on the number of classifiers allowed in the protected group to make the custom attack 'successful'.

4.2.10 Visualizing Attack and Protection. To show the impact of our attack on the features critical to the protected classifiers, we calculate the SHAP [18] values for the protected classifier using Captum's [12] GradientSHAP method. We plot the SHAP values as a heat map using Captum's [12] visualization tool against a color map of the reverse gray background. Brighter pixels in the plot indicate a higher SHAP score and more contribution to the decision by the protected classifier. The GradientSHAP method approximates the SHAP [18] values by computing the expected values of gradients and multiplying them by the difference between the input and the baseline. The GradientSHAP method can also be considered an approximation of Integrated Gradients [29], as it computes the expected values of the gradients against different baselines.



L_2 Attack: Gender Protect: Glasses

L_∞ Attack: Gender Protect: Glasses

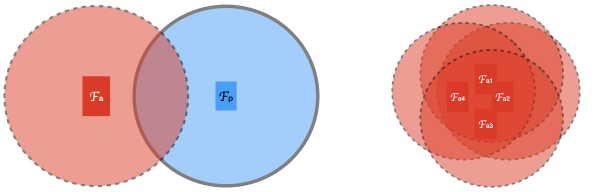
Figure 10: Visualizing SHAP values for CelebA images on *Glasses* in three scenarios: "x" (original image), "pgd_gender" (PGD attack on *Gender*) and "attackGender_defendGlasses" (our L_2 and L_∞ attacks that attack *Gender* and protect *Glasses*).

The left plot in Figure 10 compares the calculated SHAP values as heat maps for the **Glasses** classifier and contrasts them between:

1. the original normalized image;
2. PGD L_2 attack against Gender;
3. our attack against Gender and protecting **Glasses**.

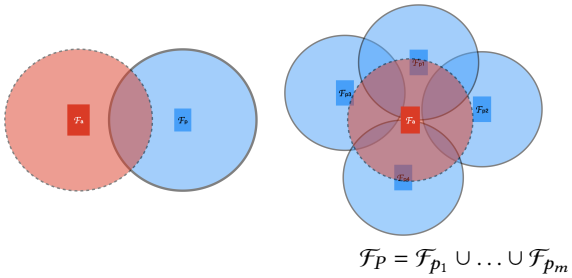
Case 1 illustrates the important features picked up by the **Glasses** classifier for prediction on the ground truth label. These features are not highlighted in a corresponding heat map image in case 2, indicating that PGD L_2 adversarial attacks against Gender added noise that also masked the important features for **Glasses**. In case 3, our custom attack successfully protected the said features, and they were highlighted as important features with brighter pixel values.

The right plot in Figure 10 shows the SHAP values on CelebA images for the L_∞ attack. We can see that adversarial examples generated using our custom attack successfully preserved features that are important for identifying the protected **Glasses** concept.



(a) $\mathcal{F}_a \cap \mathcal{F}_p$ (b) $\mathcal{F}_A = \mathcal{F}_{a_1} \cap \ldots \cap \mathcal{F}_{a_n}$

Figure 11: Protection is impossible when $\mathcal{F}_A \subset \mathcal{F}_p$.



$\mathcal{F}_P = \mathcal{F}_{p_1} \cup \ldots \cup \mathcal{F}_{p_m}$

Figure 12: Attack becomes impossible if $\mathcal{F}_a \subset \mathcal{F}_P$.

This is in contrast to the PGD L_∞ attack against Gender, where the bright pixels on the **Glasses** concept were darkened. We also see that the SHAP values for L_∞ attack were sharper and more distinct compared to the L_2 attack. This is because the L_∞ attack is stronger than the L_2 variant.

Attacks with L_∞ norm are typically stronger than attacks with L_2 norm. The accuracy of the Age classifier after the PGD L_∞ attack, shown in Table 5, was lower than the accuracy after the PGD L_2 attack (40.3% vs 30%) in Table 3, thus signifying greater attack success using the L_∞ norm. We observe that this increase in attack strength is at the expense of the accuracy of the non-attacked classifiers. In a previously discussed scenario, the accuracy of the non-attacked classifier *Gender* is 80.4% for the PGD L_2 attack against Age, and 50.2% for the PGD L_∞ attack. However, in our custom *1Attack-1Protect* attack, where we attacked the Age classifier and protected the **Gender** classifier, there was little drop in the accuracy of the protected classifiers for both L_2 and L_∞ attacks (96% versus 95.9%). Thus, the advantage of the L_∞ norm for the defense of the protected classifiers is more pronounced compared to the L_2 norm in our custom attack. This can be seen by contrasting the improvement in the accuracy of the protected **Gender** classifier for L_∞ attack (50.2% versus 95.9%) with the accuracy increase for the corresponding L_2 attack (80.4% versus 96%).

4.2.11 Discussion on the Impact of Number of Attacked and Protected Classifiers. In previous sections, we demonstrate that an attack against one classifier is transferable to other classifiers even if the latter are trained for different learning tasks. We used SHAP values to illustrate how critical features for making decisions by different classifiers often overlap, as shown in Figure 10. Besides overlapping features, the SHAP values clearly indicate that there are non-overlapping features that can be explored strategically by different classifiers. The objective of our attack-protect dual strategy is to selectively attack features ($\mathcal{F}_a$) important to the attacked classifier and avoid the overlapping features ($\mathcal{F}_a \cap \mathcal{F}_p$) affecting the protected classifier, as shown in Figure 11a. As the number of the attacked classifiers increases, in order to effectively attack them all, our choice of attack is limited to the overlapping critical features of the attacked classifiers $\mathcal{F}_A = \mathcal{F}_{a_1} \cap \ldots \cap \mathcal{F}_{a_n}$ as shown in Figure 11b, where $\mathcal{F}_{a_i}$ is the critical feature set to attack the i^{th} classifier. Protection becomes impossible when $\mathcal{F}_A \subset \mathcal{F}_p$ since any modification to features in $\mathcal{F}_A$ would inevitably modify features in $\mathcal{F}_p$, making the protected classifiers vulnerable to the attacks. Vice versa, when the number of protected classifiers increases, let $\mathcal{F}_P = \mathcal{F}_{p_1} \cup \ldots \cup \mathcal{F}_{p_m}$ where $\mathcal{F}_{p_i}$ is the critical feature set to protect the i^{th} classifier, our choice of attack is limited to $\mathcal{F}_a - \mathcal{F}_P$ in order to protect the entire protected group. Attack becomes impossible if $\mathcal{F}_a \subset \mathcal{F}_P$ as shown in Figure 12, unless we give up on protecting some of the classifiers in the protected group.

5 CONCLUSION

In this paper, we present multi-concept attack models targeting a set of classifiers. The goal of the multi-concept attack problem is to attack one set of classifiers and protect the rest without resulting in a significant drop in their accuracy. We motivate our work by showing that when an instance is assessed by multiple classifiers, attacking the classifier trained to learn one concept (e.g. *Gender*) may reduce the accuracy of the classifier trained on another concept (e.g. *Glasses*), either because of a potential correlation between multiple concepts or simply because of the added noise misleading other classifiers. We show that there is a linear programming solution to learning problems that are linear. We use linear programming to generate adversarial examples at test time. We then extend our study to the non-linear learning problems, and include the loss functions of both the attacked classifiers and the protected classifiers in our formulation. We present our experimental results on datasets where each instance belongs to different categories of concepts. We show that our approaches are successful in attacking targeted concepts while protecting others in both settings, compared to the baseline attacks. In some cases, we observe that the protected classifier's accuracy is higher than its original accuracy. We also show that our attack strategy can successfully attack and protect classifiers regardless of the size of each of the concept sets. We discuss when the attack and protect equilibrium will break, making it impossible to achieve the desired goals of attacking some and protecting others. In the case where the models are adversarially trained, we show that our custom attack can still provide protection to the protected adversarial model set while lowering the predictive accuracy of the attacked adversarial model set. However, adversarially trained models are relatively more resilient to both attack and protection.

6 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The research reported herein was supported in part by NSF awards OAC-1828467, DMS-1925346, CNS-2029661, OAC-2115094 and ARO award W911NF-17-1-0356 and a gift from Cisco Inc.

REFERENCES

[1] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 39–57. https://doi.org/10.1109/SP.2017.49

[2] Susan Collins, Yizhou Sun, Michal Kosinski, David Stillwell, and Natasha Markuzon. 2015. Are You Satisfied with Life?: Predicting Satisfaction with Life from Facebook. In *Social Computing, Behavioral-Cultural Modeling, and Prediction*, Nitin Agarwal, Kevin Xu, and Nathaniel Osgood (Eds.). Springer International Publishing, Cham, 24–33.

[3] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. 2017. Towards Interpretable Deep Neural Networks by Leveraging Adversarial Examples. arXiv:1708.05493 [cs.CV]

[4] Jörg Fliege and Benar Svaiter. 2000. Steepest descent methods for multicriteria optimization. *Math Methods Oper Res* 51 (08 2000), 479–494. https://doi.org/10.1007/s001860000043

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). ICLR, San Diego, CA, USA, https://arxiv.org/abs/1412.6572. http://arxiv.org/abs/1412.6572

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Las Vegas, NV, USA, 770–778. https://doi.org/10.1109/CVPR.2016.90

[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580 (2012), 2012. arXiv:1207.0580 http://arxiv.org/abs/1207.0580

[8] Jeremy Howard and Sylvain Gugger. 2020. fastai: A Layered API for Deep Learning. *CoRR* abs/2002.04688 (2020), https://arxiv.org/abs/2002.04688. arXiv:2002.04688 https://arxiv.org/abs/2002.04688

[9] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, Lille, France, 448–456.

[10] Agastya Kalra and Ben Peterson. 2020. Photofeeler-D3: A Neural Network with Voter Modeling for Dating Photo Impression Prediction. In *Proceedings of the Future Technologies Conference (FTC) 2019*, Kohei Arai, Rahul Bhatia, and Supriya Kapoor (Eds.). Springer International Publishing, Cham, 188–203.

[11] Hoki Kim. 2020. Torchattacks : A Pytorch Repository for Adversarial Attacks. *ArXiv* abs/2010.01950 (2020).

[12] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896 [cs.LG]

[13] Michal Kosinski. 2021. Facial recognition technology can expose political orientation from naturalistic facial images. *Scientific Reports* 11 (01 2021). https://doi.org/10.1038/s41598-020-79310-1

[14] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial Machine Learning at Scale. *CoRR* abs/1611.01236 (2016), 2017. arXiv:1611.01236 http://arxiv.org/abs/1611.01236

[15] Leqi Liu, Daniel Preotiuc-Pietro, Zahra Riahi Samani, Mohsen E. Moghaddam, and Lyle Ungar. 2021. Analyzing Personality through Social Media Profile Picture Choice. *Proceedings of the International AAAI Conference on Web and Social Media* 10, 1 (Aug. 2021), 211–220. https://ojs.aaai.org/index.php/ICWSM/article/view/14738

[16] Suyun Liu and Luís Nunes Vicente. 2019. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *CoRR* abs/1907.04472 (2019), 2019. arXiv:1907.04472 http://arxiv.org/abs/1907.04472

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, Santiago, Chile, 3730–3738. https://doi.org/10.1109/ICCV.2015.425

[18] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJzIBfZAb

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [stat.ML]

[21] István Megyeri, István Hegedüs, and Márk Jelasity. 2020. Adversarial Robustness of Model Sets. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*. IEEE, Glasgow, United Kingdom, 1–8. https://doi.org/10.1109/IJCNN48605.2020.9206656

[22] Stefano Melacci, Gabriele Ciravegna, Angelo Sotgiu, Ambra Demontis, Battista Biggio, Marco Gori, and Fabio Roli. 2020. Can Domain Knowledge Alleviate Adversarial Attacks in Multi-Label Classifiers? *CoRR* abs/2006.03833 (2020), https://arxiv.org/abs/2006.03833. arXiv:2006.03833 https://arxiv.org/abs/2006.03833

[23] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. 2016. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 2574–2582. https://doi.org/10.1109/CVPR.2016.282

[24] Juan C. Perdomo and Yaron Singer. 2019. Robust Attacks against Multiple Classifiers. *CoRR* abs/1906.02816 (2019). arXiv:1906.02816 http://arxiv.org/abs/1906.02816

[25] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox v0.8.0: A Python toolbox to benchmark the robustness of machine learning models. *CoRR* abs/1707.04131 (2017), http://arxiv.org/abs/1707.04131. arXiv:1707.04131 http://arxiv.org/abs/1707.04131

[26] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

[27] Emre Sarigol, David Garcia, and Frank Schweitzer. 2014. Online Privacy as a Collective Phenomenon. In *Proceedings of the Second ACM Conference on Online Social Networks* (Dublin, Ireland) *(COSN '14)*. Association for Computing Machinery, New York, NY, USA, 95–106. https://doi.org/10.1145/2660460.2660470

[28] J. Patrick Seder and Shigehiro Oishi. 2012. Intensity of Smiling in Facebook Photos Predicts Future Life Satisfaction. *Social Psychological and Personality Science* 3, 4 (2012), 407–413. https://doi.org/10.1177/1948550611424968 arXiv:https://doi.org/10.1177/1948550611424968

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, Sydney, NSW, Australia, 3319–3328.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). ICLR, Banff, AB, Canada, http://arxiv.org/abs/1312.6199. http://arxiv.org/abs/1312.6199

[31] Pedro Tabacof and Eduardo Valle. 2016. Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Vancouver, Canada, 426–433. https://doi.org/10.1109/IJCNN.2016.7727230

[32] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. The Space of Transferable Adversarial Examples. arXiv:1704.03453 [stat.ML]

[33] Yilun Wang and M. Kosinski. 2018. Deep Neural Networks Are More Accurate Than Humans at Detecting Sexual Orientation From Facial Images. *Journal of Personality and Social Psychology* 114 (2018), 246–257.

[34] Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial Training for Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1778–1783. https://doi.org/10.18653/v1/D17-1187

[35] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. 2020. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *Int. J. Autom. Comput.* 17, 2 (2020), 151–178. https://doi.org/10.1007/s11633-019-1211-x

[36] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2805–2824. https://doi.org/10.1109/TNNLS.2018.2886017

[37] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, Honolulu, HI, USA, 4352–4360. https://doi.org/10.1109/CVPR.2017.463