T_kML-AP: Adversarial Attacks to Top-k Multi-Label Learning

Shu Hu[1], Lipeng Ke[1], Xin Wang[2], Siwei Lyu[1]
[1]University at Buffalo, State University of New York [2]Keya Medical
{shuhu, lipengke, siweilyu}@buffalo.edu, xinw@keyamedna.com

Abstract

Top-k multi-label learning, which returns the top-k predicted labels from an input, has many practical applications such as image annotation, document analysis, and web search engine. However, the vulnerabilities of such algorithms with regards to dedicated adversarial perturbation attacks have not been extensively studied previously. In this work, we develop methods to create adversarial perturbations that can be used to attack top-k multi-label learning-based image annotation systems (T_kML-AP). Our methods explicitly consider the top-k ranking relation and are based on novel loss functions. Experimental evaluations on large-scale benchmark datasets including PASCAL VOC and MS COCO demonstrate the effectiveness of our methods in reducing the performance of state-of-the-art top-k multi-label learning methods, under both untargeted and targeted attacks.

1. Introduction

The past decade has witnessed the *tour de force* of modern deep neural networks (DNNs), which have significantly improved, or in some cases, revolutionized, the state-of-the-art performance of many computer vision problems. Notwithstanding this tremendous success, the omnipotent DNN models are surprisingly vulnerable to adversarial attacks [24, 6, 12]. In particular, inputs with specially designed perturbations, commonly known as *adversarial examples*, can easily mislead a DNN model to make erroneous predictions. The vulnerabilities of DNN models to adversarial examples impede the safe adoptions of machine learning systems in practical applications. It also motivates the explorations of algorithms generating adversarial examples [3, 17, 14] as a means to analyze the vulnerabilities of DNN models and improve their security.

Most existing works on generating adversarial examples have been focused on the case of multi-class classification [1, 24, 6, 19, 3, 17], where one instance can only be assigned to exactly one out of a set of mutually exclusive classes (labels). Because of the singleness of the labels, existing

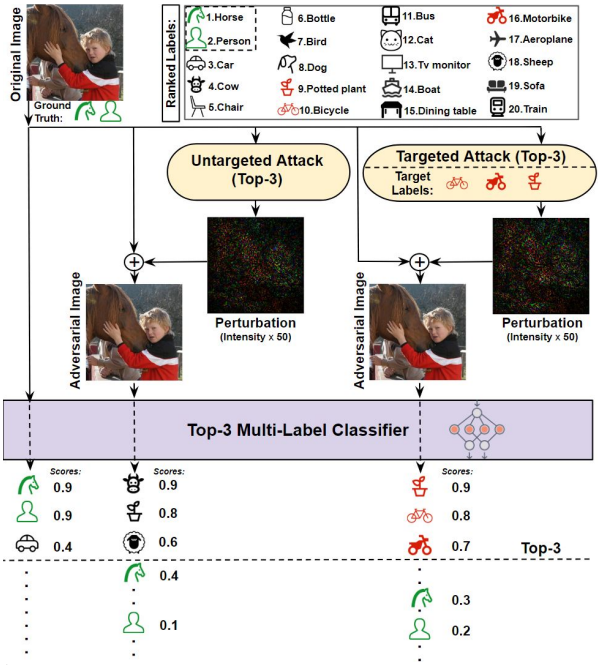

Figure 1: *Illustrative examples of the untargeted and targeted attacks to the top-3 multi-label image annotation for an image from the PASCAL VOC 2012 dataset. The green icons correspond to the ground truth labels. The red icons represent the targeted labels for attacking. The figure is better viewed in color.*

adversarial perturbation generation schemes for multi-class classification are based on the top-1 attack (*i.e.*, C&W [3], Deepfool [17]), only aiming to alter the top predicted label using the adversarial perturbation.

However, in many real-world applications such as image annotation, document categorization, and web search engines, it is more natural to solve the multi-label learning problem, where an instance is associated with a non-empty subset of labels. Furthermore, in these applications, the output of the system is usually a set of labels of a fixed size, corresponding to the top-k predicted labels. We term this as the *top-k multi-label learning* (T_kML). The practical cases of T_kML open more opportunities for attackers and leading to larger uncertainties for defenders. There are two common settings that we will consider subsequently for T_kML adversarial attacks. The *untargeted attack* aims to only replace the top-k labels with a set of arbitrary k labels that are

not true classes of the untampered input. The *targeted attack*, on the other hand, aims to coerce the T$_k$ML classifier to use a specific set of k labels that are not true classes of the input as the top-k predictions.

In this work, we describe the first untargeted and targeted adversarial attack algorithms for T$_k$ML based on a continuous formulation of the ranking operation, which we term as T$_k$ML-AP. Specifically, we note that to perturb the predictions of a T$_k$ML algorithm, it is sufficient to clear any ground-truth labels from the top-k set. There are many different ways to achieve this, but we will focus on ones that enlist the "least actions", *i.e.*, perturbing the predicted labels with minimum changes to the original label rankings. For the untargeted attack, this means move the ground-truth labels out of the top-k predictions, and for the targeted attack, this means move the target labels to the top-k set. Fig.1 gives an illustrative explanation of the proposed idea.

Thus, the key challenge in generating adversarial examples for T$_k$ML is to optimize perturbations that can lead to the change of top-k rankings of the predicted label. To this end, we introduce a reformulation of the top-k sum that lends itself to efficient numerical algorithms based on gradient descent methods. In particular, we provide loss functions for adversarial perturbations to T$_k$ML that are convex in terms of the individual prediction scores. This has a further advantage that even though the model may be nonlinear, a convex loss function can encourage many equally effective local optima. Hence any adversarial perturbation that can lead the model to have the same loss value will have equal effects. We demonstrate the effectiveness of our method on attacking state-of-the-art T$_k$ML algorithms using large scale benchmark datasets (PASCAL VOC 2012 [4] and MS COCO 2014 [11]). The main contributions of our work can be summarized as follows:

1. We present the first algorithms for untargeted and targeted adversarial attacks to the T$_k$ML problem.
2. Our method is based on a continuous reformulation of the non-differentiable ranking operation. The objective function is convex in terms of the individual prediction scores, which is easier to optimize.
3. Numerical experiments on large-scale benchmark datasets confirm the effectiveness of our method in attacking state-of-the-art T$_k$ML algorithms.

2. Backgrounds

2.1. Top-k Multi-label Learning (T$_k$ML)

Let us assume a general multi-label classification problem with a total of $m > 1$ possible labels. For an input $\mathbf{x} \in \mathbb{R}^d$, its true labels are represented by a binary label vector $\mathbf{y} = [y_1, y_2, \cdots, y_m]^\top \in \{0,1\}^m$, with $y_j = 1$ indicating that $\mathbf{x}$ is associated with the j-th label. We also use $Y = \{j | y_j = 1\}$ to represent the set of true labels of $\mathbf{x}$. Note that Y and $\mathbf{y}$ are equivalent notations of the truth labels of $\mathbf{x}$.

We introduce a continuous multi-label prediction function $F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})]^\top$, with each $f_j(\mathbf{x}) \in [0,1]$ corresponding to the prediction score of $\mathbf{x}$ with regards to the j-th class[1]. We denote $[f_{[1]}(\mathbf{x}), f_{[2]}(\mathbf{x}), \cdots, f_{[m]}(\mathbf{x})]^\top$ as the sorted values of $F(\mathbf{x})$ in descending order, *i.e.*, $f_{[1]}(\mathbf{x})$ is the largest (top-1) score, $f_{[2]}(\mathbf{x})$ is the second largest (top-2) score, and so on. Furthermore, $[j]$ corresponds to the label index of the top-j prediction score, *i.e.*, $j' = [j]$ if $f_{j'}(\mathbf{x}) = f_{[j]}(\mathbf{x})$. In ranking the values, ties can be broken in any consistent way. For input $\mathbf{x}$, the top-k multi-label classifier returns the set $\hat{Y}_k(\mathbf{x}) = \{[1], \cdots, [k]\}$ for $1 \le k < m$. In other words, we can convert a general multi-label predictor F to a top-k multi-label classifier by returning the set of labels corresponding to the set of top-k prediction scores from $F(\mathbf{x})$. This problem is related to many types of learning problems. If $|Y| = 1$, $k = 1$, it becomes the conventional multi-class problem. If $|Y| = 1$, $k \ge 1$, it becomes top-k multi-class problem [10]. If $k = |Y|$, $|Y| \ge 1$, it becomes the conventional multi-label problem. The top-k setting is often implicitly used in applications of multi-label learning. For instance, in image annotation, when the number of possible labels is large, the system often returns a fixed number of top annotations that are most relevant to the image[2].

A successful top-k multi-label classification should lead to consistency between the true labels (Y) and the predicted labels $\hat{Y}_k(\mathbf{x})$ of the input. The situation is complicated by the difference in size of Y and $\hat{Y}_k(\mathbf{x})$, so we use the following criterion: when $k \ge |Y|$, it corresponds to $Y \subseteq \hat{Y}_k(\mathbf{x})$; when $k \le |Y|$, it is the case $\hat{Y}_k(\mathbf{x}) \subseteq Y$. In other words, one is the subset of the other depending on the relation of k and the number of the truth labels. We define the top-k label consistency score as:

$$E(Y, \hat{Y}_k(\mathbf{x})) = \mathbb{I}_{Y \subset \hat{Y}_k(\mathbf{x})} + \mathbb{I}_{\hat{Y}_k(\mathbf{x}) \subset Y} + \mathbb{I}_{Y = \hat{Y}_k(\mathbf{x})}, \quad (1)$$

where $\mathbb{I}_c$ is the indicator function that takes value 1 when condition c is true and 0 otherwise. As such, $E(Y, \hat{Y}_k(\mathbf{x}))$ is 1 for a successful multi-label classification of input $\mathbf{x}$ and 0 otherwise.

2.2. Top-k and Average Top-k

Top-k ranking emerges as a natural element in the learning objectives in various problems such as multi-class learning and robust binary classification [10, 13, 8]. However, as

[1] Here we assume the prediction scores are *calibrated*, *i.e.*, taking values in the range of $[0,1]$. For $f \in \mathbb{R}$, we can use simple transforms such as $\frac{1}{1+e^{-f}}$ to map it to the range of $[0,1]$ without changing their ranking.

[2] Another strategy in multi-label learning is to return all labels with prediction score above a preset threshold. The result will be a list of labels of varying length. The top-k multi-label classification can be regarded as using a varying threshold to fix the number of the returned labels.

a function of all elements in a set, the top-k ranking function is non-continuous, non-differentiable, and non-convex. This makes the optimization involving top-k ranking challenging.

To mitigate these problems of the top-k operator, we can use the *average of top-k* function [5], which is defined for a set $F = \{f_1, \cdots, f_m\}$ as

$$\phi_k(F) = \frac{1}{k} \sum_{j=1}^{k} f_{[j]}. \tag{2}$$

It is not difficult to show that (i) $\phi_k(F) \geq f_{[k]}$, and (ii) $\phi_k(F) = f_{[k]}$ when $f_{[1]} = \cdots = f_{[k]}$. As such, the average of top-k is a tight upper-bound of the top-k. It can be proved that $\phi_k(F)$ is a convex function of the elements of F [2]. More importantly, it affords an equivalent form as an optimization problem [18].

Lemma 1. *For $f_i(\mathbf{x}) \in [0, 1]$, we have*

$$\phi_k(F) = \frac{1}{k} \min_{\lambda \in [0,1]} \{k\lambda + \sum_{j=1}^{m} [f_j - \lambda]_+\} \tag{3}$$

$$f_{[k]} \in \operatorname{argmin}_{\lambda \in [0,1]} \{k\lambda + \sum_{j=1}^{m} [f_j - \lambda]_+\}, \tag{4}$$

where $[a]_+ = \max\{0, a\}$ is the hinge function.

For completeness, we include the proof of Lemma 1 in Appendix A.1. Lemma 1 enables us to incorporate the average top-k function in conventional sub-gradient based optimization.

2.3. Related Works

Due to the limit of space, we only provide a brief overview of relevant works. A full survey of adversarial attacks on deep learning models can be found in [1]. The major differences between the work in this paper and the related works are summarized in Table 1.

Most existing adversarial attacking methods target multi-class classification problems (corresponding to the special case of T_kML with $k = 1$ and $|Y| = 1$ for all inputs). As such, these methods often target the top prediction and aim to change it with perturbations. For untargeted attacks, DeepFool [17] is a generalization of the minimum attack under general decision boundaries by swapping labels from the top-2 prediction. The work of [16] (UAP) aims to find universal adversarial perturbations that are independent of individual input images. Both DeepFool and UAPs are top-1 multi-class adversarial attack methods. For targeted attacks, FGSM [6] and I-FGSM [9] are two popular attack schemes that use the gradient of the DNN models with regards to the input to generate adversarial samples. The CW method [3] improves on the previous method by using regularization and modified constraints.

Realizing that only attacking the top predictions may not be effective, several works introduce attacks to the top-k (for $k > 1$) predictions in a multi-class classification system. kFool [25] and CW^k [26] extend the original DeepFool [17] and CW [3] methods to exclude the truth label out of the top-k predictions. kFool is based on a geometry view of the decision boundary between k labels and the truth label in the multi-class problem. The UAP method is extended in [25] to top-k Universal Adversarial Perturbations (kUAPs). In addition, the CW method is extended to a top-k version known as CW^k in [26]. However, all these methods are still designed for multi-class classification (*i.e.*$|Y| = 1$), and cannot be directly adapted to the attacks to the more general top-k multi-label learning.

Methods \ Features	Multi Label	Untargeted Attack	Universal Attack	Targeted Attack	Top-k
kFool [25]	×	✓	×	×	✓
kUAPs [25]	×	✓	✓	×	✓
CW^k [26]	×	×	×	✓	✓
ML-AP [22]	✓	×	×	✓	×
T_kML-AP-U (this paper)	✓	✓	×	×	✓
T_kML-AP-Uv (this paper)	✓	✓	✓	×	✓
T_kML-AP-T (this paper)	✓	×	×	✓	✓

Table 1: *Summary of the difference between previous works with our methods (T_kML-AP).*

The authors of [22] describes an adversarial attack to multi-label classification extending existing attacks to multi-class classification. This method is further studied in [27], which transfers the problem of generating attack to a linear programming problem. To make the predictions of adversarial examples lying inside of the training data distribution, [15] proposed a multi-label attack procedure with an additional domain knowledge-constrained classifier. These are all for multi-label learning without the top-k constraint. Our experiments in Section 4 show that they are not effective for the top-k setting.

3. Method

In this work, we introduce new methods to generate adversarial perturbations to attack top-k multi-label classification. We term our method as T_kML-AP. Unlike the multi-label adversarial perturbation method in [22], we consider the top-k ranking in the T_kML problem an essential requirement to design loss functions in our methods. Hence, in our methods, the ranking relation is explicitly handled, using the results in Section 2.2. Specifically, we describe our methods in detail for the instance-specific and instance-independent (universal) untargeted attacks (T_kML-AP-U and T_kML-AP-Uv) and targeted attacks (T_kML-AP-T). A comparison with previous works is given in Table 1.

3.1. Untargeted Attack

Formulation. The untargeted attack to top-k multi-label learning (T_kML-AP-U) aims to find a minimum perturbation to the input that can push the prediction scores of the truth labels outside of the top-k set. It can be formulated as finding a perturbation signal $\mathbf{z}$ for an input $\mathbf{x}$ such that

$$\min_{\mathbf{z}} \|\mathbf{z}\|_2, \text{ s.t. } E(Y, \hat{Y}_k(\mathbf{x} + \mathbf{z})) = 0, \tag{5}$$

where $E(Y, \hat{Y}_k)$ is defined in Eq.(1)[3]. Because $k \leq m-1$, we can rewrite Eq.(5) with a more revealing equivalent form,

$$\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{z}\|_2^2, \text{ s.t. } \max_{j \in Y} f_j(\mathbf{x}+\mathbf{z}) \leq f_{[k+1]}(\mathbf{x}+\mathbf{z}). \quad (6)$$

Note that the constraint is equivalent to have $Y \subseteq \{j | f_j(\mathbf{x}+\mathbf{z}) < f_{[k]}(\mathbf{x}+\mathbf{z})\}$, the converse of which means at least one truth label is inside the top-k range.

Relaxation. The optimization problem in Eq.(6) is difficult to optimize directly, so we introduce the top-k multi-label loss, $\left[\max_{j \in Y} f_j(\mathbf{x}+\mathbf{z}) - f_{[k+1]}(\mathbf{x}+\mathbf{z})\right]_+$, as a surrogate to the constraint. The top-k multi-label loss precisely reflects the requirement to exclude the true labels out of the top-k range. It is zero when the maximal prediction score from the true labels is no greater than the $k+1$-th prediction score, and positive otherwise. Rewriting the objective function using the Lagrangian form, we have

$$\min_{\mathbf{z}} \frac{\beta}{2}\|\mathbf{z}\|_2^2 + \left[\max_{j \in Y} f_j(\mathbf{x}+\mathbf{z}) - f_{[k+1]}(\mathbf{x}+\mathbf{z})\right]_+, \quad (7)$$

where $\beta > 0$ is a prechosen trade-off parameter.

Optimization. The ranking operation in Eq.(7) can be further removed. Specifically, denote $t_{m+1-j} = [\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_j(\mathbf{x}+\mathbf{z})]_+$ for $j = 1, \cdots, m$. With a bit of abuse of the notation, we denote $t_{[j]}$ as the top-j element in the set $\{t_1, \cdots, t_m\}$[4]. Note that there is a simple correspondence, as $t_{[m-k]} = [\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_{[k+1]}(\mathbf{x}+\mathbf{z})]_+$.

As shown in Section 2.2, we have the following bound of the top-k value using the average of $\{t_1, \cdots, t_m\}$, as $\frac{1}{m-k}\sum_{j=1}^{m-k} t_{[j]} \geq t_{[m-k]} = [\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_{[k+1]}(\mathbf{x}+\mathbf{z})]_+$. Furthermore, using Lemma 1, we can rewrite the average of top-$(m-k)$ elements of $\{t_1, \cdots, t_m\}$ as $\frac{1}{m-k}\sum_{j=1}^{m-k} t_{[j]} = \min_{\lambda \in [0,1]} \lambda + \frac{1}{m-k}\sum_{j=1}^{m}[t_j - \lambda]_+$. Replacing the definition of t_j, the inner term $[[\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_{[m+1-j]}(\mathbf{x}+\mathbf{z})]_+ - \lambda]_+$ can be further simplified by removing the double hinge function to $[\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_{[j]}(\mathbf{x}+\mathbf{z}) - \lambda]_+$, according to the following result.

Lemma 2. *For $\lambda \geq 0$, $[[a-x]_+ - \lambda]_+ = [a - x - \lambda]_+$.*

The proof of Lemma 2 is deferred to Appendix A.2. Putting all results together, we get the objective function for finding adversarial perturbation in an untargeted attack

[3]Note that the definition is minimal: it only changes labels that are correctly predicted by $F(\mathbf{x})$, *i.e.*$Y \cap \hat{Y}_k(\mathbf{x})$. True labels that are incorrectly predicted by F and not in $\hat{Y}_k(\mathbf{x})$ are expected to be intact.

[4]Note that $[j]$ in $t_{[j]}$ may not correspond to the same index as in the case of $f_{[j]}$ as it depends on the ranking of different sets.

Algorithm 1: Untargeted Attack (T_kML-AP-U)

Input: $\mathbf{x}$, predictor F, k, η_l, β
Output: adversarial example $\mathbf{x}^*$, perturbation $\mathbf{z}^*$
Initialization: $l = 0$, $\mathbf{x}^* = \mathbf{x}$, $\mathbf{z}_0$, and λ_0
while $E(Y, \hat{Y}_k(\mathbf{x}+\mathbf{z})) \neq 0$ **do**
  Compute $\mathbf{z}_{l+1}$ and λ_{l+1} with Eq.(9);
  $\mathbf{x}^* = \mathbf{x} + \mathbf{z}_{l+1}$, $\mathbf{z}^* = \mathbf{z}_{l+1}$;
  $l = l + 1$;
end
return $\mathbf{x}^*$, $\mathbf{z}^*$

to the top-k multi-label learning as:

$$\min_{\lambda \in [0,1], \mathbf{z}} \frac{\beta}{2}\|\mathbf{z}\|_2^2 + \lambda + \frac{1}{m-k}\sum_{j=1}^{m}[\max_{y \in Y} f_y(\mathbf{x}+\mathbf{z}) - f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+. \quad (8)$$

This optimization problem can be solved with an iterative gradient descent approach [8, 5]. We initialize $\mathbf{z}$ and λ, then update them with the following steps:

$$\begin{aligned}\mathbf{z}_{l+1} =& (1-\beta\eta_l)\mathbf{z}_l - \frac{\eta_l}{m-k}\sum_{j=1}^{m}\left(\frac{\partial f_{y'}(\mathbf{x}')}{\partial \mathbf{x}'} - \frac{\partial f_j(\mathbf{x}')}{\partial \mathbf{x}'}\right)\Big|_{\mathbf{x}'=\mathbf{x}+\mathbf{z}_l} \\ & \cdot \mathbb{I}_{[f_{y'}(\mathbf{x}+\mathbf{z}_l) - f_j(\mathbf{x}+\mathbf{z}_l) > \lambda_l]}, \\ \lambda_{l+1} =& \lambda_l - \eta_l \cdot \left(1 - \frac{1}{m-k}\sum_{j=1}^{m}\mathbb{I}_{[f_{y'}(\mathbf{x}+\mathbf{z}_l) - f_j(\mathbf{x}+\mathbf{z}_l) > \lambda_l]}\right),\end{aligned} \quad (9)$$

where η_l is the step size and $y' \in \max_{y \in Y}$. This iterative process continues until the termination conditions are met. The overall procedure is described in detail in Algorithm 1.

Universal untargeted attack. We can extend the instance-specific untargeted attack to a universal adversarial attack that is independent of the input [16] so can be shared by all instances. Specifically, given a dataset $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and its ground truth label set $Y = \{Y_1, \cdots, Y_n\}$, where $Y_i := Y(\mathbf{x}_i)$, finding the instance-independent (universal) adversarial perturbation $\mathbf{z}$ is formulated as $\min_{\lambda \in [0,1], \mathbf{z}} \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{z}, \lambda; \mathbf{x}_i, Y_i)$, where $L(\mathbf{z}, \lambda; \mathbf{x}_i, Y_i)$ is the objective function in Eq.(8). The solution to the universal untargted attack can be obtained using a similar procedure based on Algorithm 1. Please refer to Appendix B.3 for the details about the T_kML-AP-Uv algorithm.

3.2. Targeted Attack

Formulation. We next consider the targeted attack, the aim of which is to plant a set of k labels, $\widetilde{Y} \subset \{1, \cdots, m\}$ and $\widetilde{Y} \cap Y = \varnothing$, as the top-$k$ predictions. We formulate the learning objective of the targeted attack on top-k multi-label learning (T_kML-AP-T) as

$$\min_{\mathbf{z}} \|\mathbf{z}\|_2, \text{ s.t. } \widetilde{Y} = \hat{Y}_k(\mathbf{x}+\mathbf{z}). \quad (10)$$

The constraint in Eq.(10) exactly reflects the requirement that the top-k predicted labels of the perturbed are all from the targeted label set.

Relaxation. Analogous to the untargeted attack, we rewrite the objective function into a form that lends itself to optimization. Specifically, if we have $\widetilde{Y} = \hat{Y}_k(\mathbf{x}+\mathbf{z})$, it means that the prediction scores of labels in $\widetilde{Y}$ occupy the top-k ranks. So the sum of prediction scores from labels in the sets $\widetilde{Y}$ and $\hat{Y}_k(\mathbf{x}+\mathbf{z})$ are also the same, *i.e.*, we have $\sum_{j=1}^{k} f_{[j]}(\mathbf{x}+\mathbf{z}) - \sum_{j\in\widetilde{Y}} f_j(\mathbf{x}+\mathbf{z}) = 0$. Furthermore, if $\widetilde{Y} \neq \hat{Y}_k(\mathbf{x}+\mathbf{z})$, $\sum_{j=1}^{k} f_{[j]}(\mathbf{x}+\mathbf{z}) - \sum_{j\in\widetilde{Y}} f_j(\mathbf{x}+\mathbf{z}) \geq 0$ as by definition, the second term cannot be greater than the first term. This suggest that $\sum_{j=1}^{k} f_{[j]}(\mathbf{x}+\mathbf{z}) - \sum_{j\in\widetilde{Y}} f_j(\mathbf{x}+\mathbf{z})$ is a surrogate to the constraint in Eq.(10). It is zero when all target attacked labels are in the top-k positions, and positive otherwise. Introducing the Lagrangian form, we can reformulate Eq.(10) as

$$\min_{\mathbf{z}} \tfrac{\beta}{2}\|\mathbf{z}\|_2^2 + \textstyle\sum_{j=1}^{k} f_{[j]}(\mathbf{x}+\mathbf{z}) - \sum_{j\in\widetilde{Y}} f_j(\mathbf{x}+\mathbf{z}), \quad (11)$$

where $\beta > 0$ is a prechosen trade-off parameter. Note that the second term in Eq.(11) is precisely the sum of top-k elements. Using the results of Lemma 1, we can remove the explicit ranking operation in Eq.(11). Specifically, we have $\sum_{j=1}^{k} f_{[j]}(\mathbf{x}+\mathbf{z}) = \min_{\lambda\in[0,1]}\{k\lambda + \sum_{j=1}^{m}[f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+\}$. Further simplifying the last two terms in Eq.(11) yields

$$\begin{aligned}
&\left\{k\lambda + \textstyle\sum_{j=1}^{m}[f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+\right\} - \sum_{j\in\widetilde{Y}} f_j(\mathbf{x}+\mathbf{z}) \\
=&\textstyle\sum_{j\in\widetilde{Y}} \Big([f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+ - (f_j(\mathbf{x}+\mathbf{z}) - \lambda)\Big) \\
+&\textstyle\sum_{j\notin\widetilde{Y}} [f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+ \\
=&\textstyle\sum_{j\in\widetilde{Y}} [\lambda - f_j(\mathbf{x}+\mathbf{z})]_+ + \sum_{j\notin\widetilde{Y}} [f_j(\mathbf{x}+\mathbf{z}) - \lambda]_+,
\end{aligned}$$

where we use a fact that $[a]_+ - a = [-a]_+$. Introducing $s_j = 2\mathbb{I}_{j\in\widetilde{Y}} - 1 \in \{-1, 1\}$, we can rewrite Eq.(11) more concisely as

$$\min_{\lambda\in[0,1],\mathbf{z}} \tfrac{\beta}{2}\|\mathbf{z}\|_2^2 + \textstyle\sum_{i=1}^{m}[s_j(\lambda - f_j(\mathbf{x}+\mathbf{z}))]_+ \quad (12)$$

Optimization. This optimization problem can also be solved with an iterative gradient descent approach as in the untargeted attack. We initialize $\mathbf{z}$ and λ, then update them with the following steps:

$$\begin{aligned}
\mathbf{z}_{l+1} =& (1-\beta\eta_l)\mathbf{z}_l \\
&- \eta_l \sum_{j=1}^{m} (-s_j) \left.\frac{\partial f_j(\mathbf{x}')}{\partial \mathbf{x}'}\right|_{\mathbf{x}'=\mathbf{x}+\mathbf{z}_l} \cdot \mathbb{I}_{[s_j(\lambda_l - f_j(\mathbf{x}+\mathbf{z}_l))>0]} \\
\lambda_{l+1} =& \lambda_l - \eta_l \sum_{j=1}^{m} s_j \cdot \mathbb{I}_{[s_j(\lambda_l - f_j(\mathbf{x}+\mathbf{z}_l))>0]}
\end{aligned} \quad (13)$$

where η_l is the step size. The overall procedure is described in detail in Algorithm 2. The algorithm stops when the termination conditions are met.

Algorithm 2: Targeted Attack (T_kML-AP-T)

Input: $\mathbf{x}$, predictor F, $\widetilde{Y}$, max_iter, η_l, β
Output: adversarial example $\mathbf{x}^*$, perturbation $\mathbf{z}^*$

Initialization: $l = 0$, $\mathbf{x}^* = \mathbf{x}$, $\mathbf{z}_0$, and λ_0
while $l \leq$ *max_iter* **do**
  Compute $\mathbf{z}_{l+1}$ and λ_{l+1} with Eq.(13);
  $\mathbf{x}^* = \mathbf{x} + \mathbf{z}_{l+1}$, $\mathbf{z}^* = \mathbf{z}_{l+1}$;
  $l = l + 1$;
end
return $\mathbf{x}^*$, $\mathbf{z}^*$

4. Experiments

We evaluate the performance of the proposed adversarial attacks (*i.e.*, T_kML-AP-U, T_kML-AP-Uv, and T_kML-AP-T) in the practical problem of image annotation, the goal of which is to predict the labels of an input image. Due to the limit of the space, we present the most significant information and results of our experiments, with more detailed information and additional results in the complementary materials[5].

4.1. Experimental Settings

Datasets and baseline models. Our experiments are based on two popular large-scale image annotation datasets, namely PASCAL VOC 2012 [4] and MS COCO 2014 [11]. Both datasets have multiple true labels associated with each image: the average number of positive labels per instance in PASCAL VOC 2012 and MS COCO 2014 are 1.43 (out of 20) and 3.67 (out of 80), respectively. All RGB images are with pixel intensities in the range of $\{0, 1, \cdots, 255\}$.

On the two datasets, we train deep neural network-based top-k multi-label classifiers as baseline models. For the PASCAL VOC 2012 dataset, similar to [22], we adopt the inception-v3 [23] model pre-trained on ImageNet [21] and fine-tuned on PASCAL VOC 2012. For MS COCO 2014 datasets, we use a ResNet50 [7] based model. Both models are originally designed for multi-class classification, so we convert them to multi-label models by replacing the softmax layer with sigmoid classification layer as in [20][6]. We further modify the model to output the top-k predicted labels.

We select 1,000 images from the validation set in PASCAL VOC 2012 and MS COCO 2014 datasets respectively as the test set to test the untargeted and targeted attack methods. These images are correctly predicted by the baseline T_kML models, *i.e.*, the predicted top-k labels either contain

[5]Code: https://github.com/discovershu/TKML-AP.

[6]After retraining the models, we get 0.934 mAP performance for PASCAL VOC 2012 and 0.867 mAP performance for MS COCO 2014 on the corresponding validation datasets, which are close to the state-of-the-art performance [22, 20].

or are completely from the true labels. For the universal untargeted attack, however, we need a training dataset to find the universal perturbation. Therefore, we select 3,000 images from the validation set of MS COCO 2014 as the training set and evaluate the attack performance on another different 1,000 images from the same validation set. For the targeted attacks, we choose the target labels as in [3, 26], where we consider three different strategies (see Fig.4 for more details).

- Best Case. In this case, we select k labels that are not true labels and have the highest prediction scores. These labels are the runner-ups and the regarded as the easiest labels to attack.
- Random Case. In this case, we randomly select k labels that are not true labels following a uniform distribution.
- Worst Case. In this case, we select k labels that are not true labels with the lowest prediction scores. These labels are the most difficult to attack.

Evaluation metrics. For the instance-specific untargeted and targeted attacks, we use the attack success rate (ASR) as an evaluation metric of the attack performance, which is defined as

$$\texttt{ASR} = 1 - \frac{1}{n}\sum_{i=1}^{n} E(Y(\mathbf{x}_i), \hat{Y}_k(\mathbf{x}_i + \mathbf{z}_i)), \qquad (14)$$

where n is the number of evaluation data. Higher values of ASR indicate the corresponding method has a high attacking performance. This metric extends the one used in [25] for multi-class classification $|Y| = 1$. In the universal untargted attack, we use a slightly different ASR definition to reflect that the perturbation is shared by all instances, as $\texttt{ASR} = 1 - \frac{1}{n}\sum_{i=1}^{n} E(Y(\mathbf{x}_i), \hat{Y}_k(\mathbf{x}_i + \mathbf{z}))$. To evaluate the perceptual quality, we define the average per-pixel perturbation over all successful attacks as

$$\texttt{Pert} = \frac{1}{n \cdot \texttt{ASR}} \sum_{i=1}^{n} \frac{\|\mathbf{z}_i\|_2 (1 - E(Y(\mathbf{x}_i), \hat{Y}_k(\mathbf{x}_i + \mathbf{z}_i))}{\text{\# of pixels of } \mathbf{x}_i}. \qquad (15)$$

The lower value of Pert means that the perturbation is less perceivable. The hyper-parameter β is chosen to achieve a good trade-off between ASR and Pert.

Compared Methods. We use experiments to test the practical performance of T$_k$ML-AP. However, as there are no dedicated adversarial perturbation generation methods for the top-k multi-label learning, we adapt several existing adversarial attacks designed for the general multi-label or multi-class learning as comparison baselines. Specifically, we use the following methods.

- Untargeted attack (kFool): We replace the prediction score of one ground-truth label in the kFool [25] algorithm with the maximum prediction score among all ground truth labels as an untargeted attack comparative method.
- Universal attack (kUAPs): We use the kUAPs from [25] with only replace the inner kFool method with our modified untargeted attack comparative method.

k	Methods	PASCAL VOC 2012		MS COCO 2014	
		Pert($\times 10^{-2}$)	ASR	Pert($\times 10^{-2}$)	ASR
3	kFool	1.64	93.7	5.49	61.4
	T$_k$ML-AP-U	**0.51**	**99.6**	**0.49**	**100**
5	kFool	2.39	93.5	9.91	65.2
	T$_k$ML-AP-U	**0.56**	**99.3**	**0.53**	**100**
10	kFool	4.88	88.7	16.44	68.1
	T$_k$ML-AP-U	**0.63**	**98.3**	**0.59**	**100**

Table 2: *Comparison of Pert and ASR (%) of the untargeted attack methods with k=3, 5, 10 on two datasets. The best results are shown in bold.*

- Targeted attack (ML-AP): We adapt the Rank I algorithm from [22] with the loss function $[\max_{j \notin P} f_j(\mathbf{x}+\mathbf{z}) - \min_{j \in P} f_j(\mathbf{x}+\mathbf{z})]_+$ to a targeted attack comparative method, where P contains the targeted labels (exclude the ground truth labels) and $|P| = k$. It should be mentioned that this loss is similar to the loss function in [26] when we do not consider the order of targeted labels.

These methods, together with the proposed methods, namely T$_k$ML-AP-U, T$_k$ML-AP-Uv, T$_k$ML-AP-T, are applied to attack the baseline models trained on the datasets.

4.2. Results

Untargeted Attacks. The performance of untargeted attacks is shown in Table 2. Note that for different k values, the T$_k$ML-AP-U method achieves a nearly complete obviation (with very high ASR values) on both the PASCAL VOC 2012 dataset and the MS COCO 2014 dataset with small perturbation scales (indicated by the smaller Pert values). On the other hand, the simple adoption of the DeepFool method (kFool) is much less effective. This could be attributed to the explicit consideration of the top-k prediction in T$_k$ML-AP-U. The quantitative results are corroborated with an example from the PASCAL VOC 2012 shown in Fig.2. Although both kFool and T$_k$ML-AP-U show effectiveness in attacking the top-k predictions from the baseline method, kFool introduces larger perturbations in general. In many cases, the perturbations are visible as shown in Fig.2.

When deployed in practice, it is possible that the attack is designed for top-k predictions but the actual system is used to find the top-k' outputs. In other words, there can be a mismatch between the cutoff rank that is used in the attack k from that used in the system k'. Note that by the definition of T$_k$ML-AP-U, a successful attack to a top-k multi-label learning system is necessarily a successful attack to the same system for the case of top-k' for $k' \leq k$. This is because the top k' set is a subset of the top k set. On the other hand, we perform a set of experiments to validate the case when $k' > k$. Specifically, in Table 3, we show the results of running T$_k$ML-AP-U for $k = 3$ and $k' = 5, 10$, respectively. Note that in these cases, the effectiveness of the effect significantly reduces from the case of $k = k'$. This is expected since a successful top-k attack will move the original top-k labels to ranks greater than k. However, our objective Eq.(8) cannot avoid the case when some of the

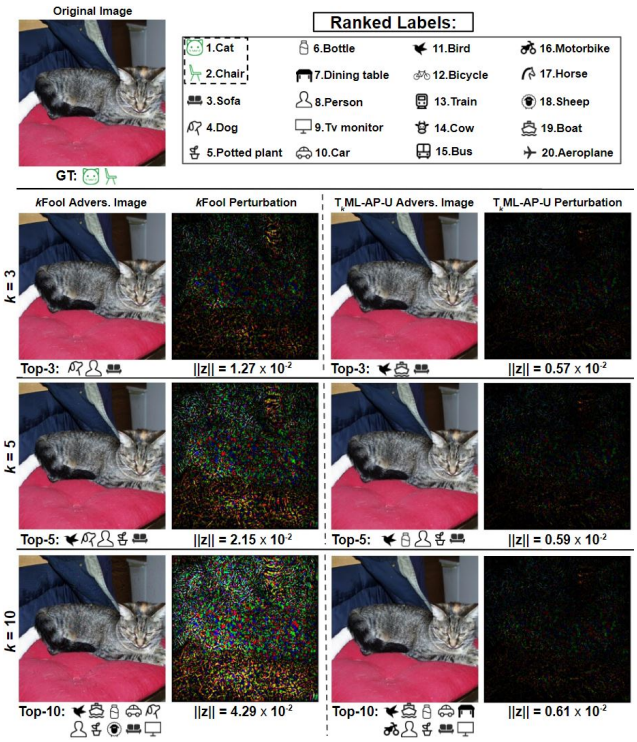

Figure 2: *Visual examples of untargeted attack methods on PASCAL VOC 2012. The perturbations are scaled by a factor of 20 to increase visibility. Green icons represent the truth labels (GT) that are attacked. The figure is better viewed in color.*

k'	Method (k=3)	PASCAL VOC 2012		MS COCO 2014	
		Pert($\times 10^{-2}$)	ASR	Pert($\times 10^{-2}$)	ASR
3	T_kML-AP-U	0.51	99.6	0.49	100
5	T_kML-AP-U	0.24	3.6	0.43	26.5
10	T_kML-AP-U	0.18	0.3	0.35	3.9

Table 3: *Comparison of* Pert *and* ASR *(%) of the untargeted attack methods in $k' = 3, 5, 10$ on two datasets when setting k=3.*

k	1		2		3	
Metrics	Pert	ASR	Pert	ASR	Pert	ASR
kUAPs	0.51	63.9	0.51	74.6	0.51	73.2
T_kML-AP-Uv	**0.13**	**86.5**	**0.15**	**82**	**0.16**	**80.5**

Table 4: *Comparison of* Pert *and* ASR *(%) of the universal untargeted attack methods on MS COCO 2014.*

original labels are placed between k and k', so a successful attack to the top-k case may not generalize to a successful attack to the case of top-k', $(k < k')$.

Universal Untargeted Attacks. The results of universal untargeted attacks are shown in Table 4. On the MS COCO 2014 dataset, T_kML-AP-Uv outperforms kUAPs in all cases. Fig.3 further exhibits visual examples of universal perturbation with T_kML-AP-Uv and kUAPs for $k = 3$. With similar perturbations, T_kML-AP-Uv is successful in attacking all top-3 labels but there are images that kUAPs fails to attack. On the other hand, because of the requirement of being instance-independent, to achieve the same level of attacks, universal untargeted attacks need to introduce larger visual perturbations than those in the instance-specific attacks.

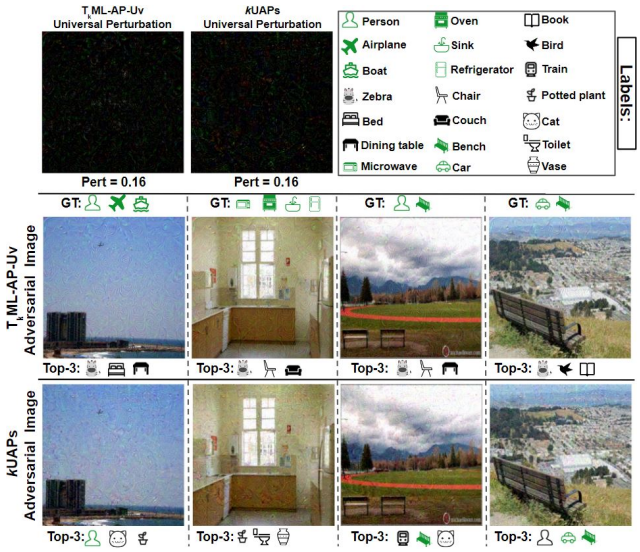

Figure 3: *Examples of universal untargeted attack methods with $k = 3$ on MS COCO 2014. The figure is better viewed in color.*

Cases	k	Methods	PASCAL VOC 2012		MS COCO 2014	
			Pert($\times 10^{-2}$)	ASR	Pert($\times 10^{-2}$)	ASR
Best	3	ML-AP	0.44	96.2	0.55	100
		T_kML-AP-T	0.44	**96.6**	0.57	**100**
	5	ML-AP	0.50	92	0.66	99.9
		T_kML-AP-T	0.50	**92.8**	0.69	**99.9**
	10	ML-AP	0.55	84.2	0.81	99.8
		T_kML-AP-T	0.56	**86.4**	0.85	**99.8**
Random	3	ML-AP	0.59	86	0.95	99.8
		T_kML-AP-T	0.59	**89.8**	0.99	**99.9**
	5	ML-AP	0.62	77.9	1.11	96.5
		T_kML-AP-T	0.63	**83.7**	1.18	**97.8**
	10	ML-AP	0.63	67.7	1.22	84.2
		T_kML-AP-T	0.64	**76.4**	1.28	**94.5**
Worst	3	ML-AP	0.66	68	1.08	90
		T_kML-AP-T	0.66	**75.8**	1.14	**91.4**
	5	ML-AP	0.67	53.3	1.18	81.8
		T_kML-AP-T	0.69	**66.6**	1.25	**87.2**
	10	ML-AP	0.67	39.1	1.25	59
		T_kML-AP-T	0.69	**57**	1.30	**73.1**

Table 5: *Comparison of* Pert *and* ASR *(%) of the targeted attack methods with k=3, 5, 10 in the Best, Random, and Worst cases on two datasets. The best* ASR *results are shown in bold.*

Targeted Attacks. In Table 5, we evaluate the performance of T_kML-AP-T (our method) and compare it with the ML-AP method in the three attack settings (Best, Random, and Worst) as described in Section 4.1. On both datasets and over all cases, T_kML-AP-T outperforms ML-AP for different k values and comparing settings in terms of the ASR scores with comparable perturbation strengths. In particular, with increasing k, the gap of ASR scores between T_kML-AP-T and the ML-AP method also increases. On the other hand, we note that the ASR scores decrease when the k value increases. This is because the attack methods need to take more effort to put labels in the set $\widetilde{Y}$ to top-k positions. The attacks become more challenging for both methods when the target labels are chosen according to the Worst setting, reflecting that larger perturbations are required to modify the predictions to the more difficult labels. Visual results of targeted attack using T_kML-AP-T

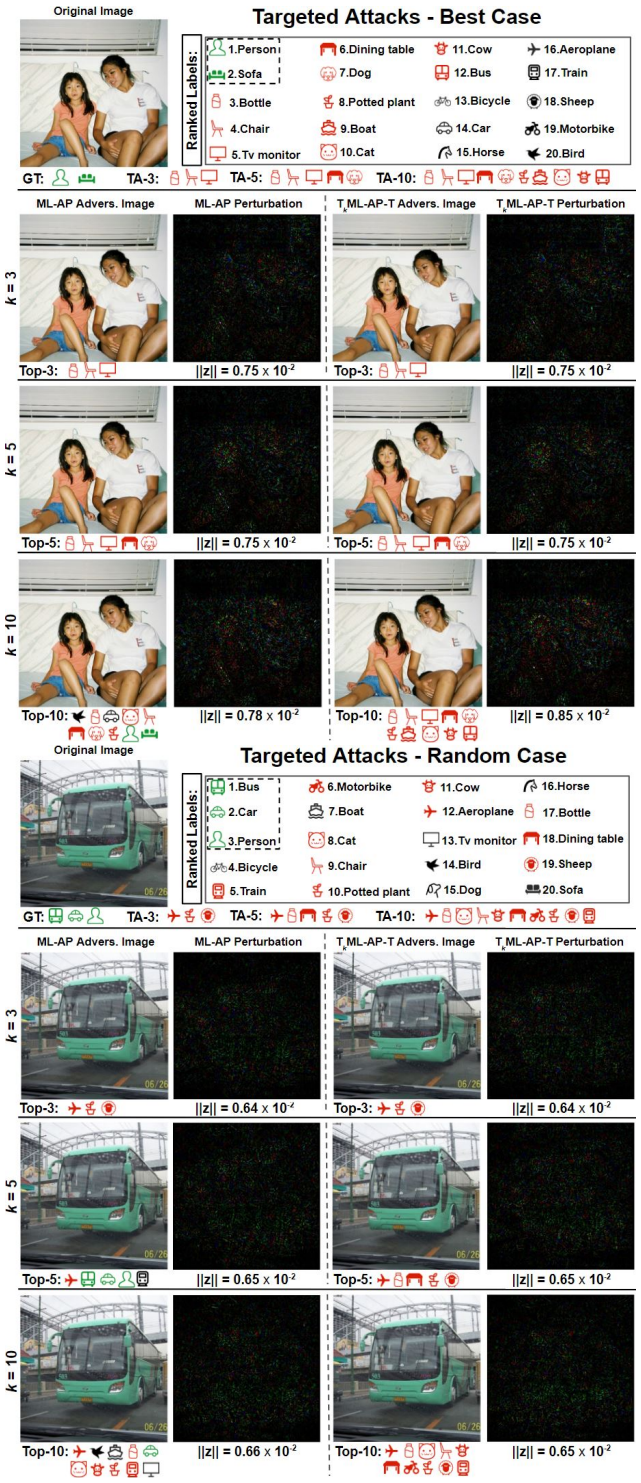

Figure 4: *Targeted attack in Best (**left-top**, targeted labels are near GT) , Random (**left-bottom**, targeted labels are randomly selected), and Worst (**right**, targeted labels are far from GT) cases. TA means the targeted attack labels. The perturbations are scaled by a factor of 20 to increase visibility. The red icons represent the targeted labels for attacking. The figure is better viewed in color.*

and ML-AP are shown in Fig.4.

5. Conclusion

Top-k multi-label learning (T_kML) has many practical applications. However, the vulnerability of such algorithms with regards to dedicated adversarial perturbation attacks has not been extensively studied previously. In this work, we develop white-box generation methods of adversarial perturbations to T_kML based image annotation system for both untargeted and targeted attacks. Our methods explicitly consider the top-k ranking relation and are based on novel loss functions. Experimental evaluations on large-scale benchmark datasets including PASCAL VOC and MS COCO demonstrate the effectiveness of our methods in reducing the performance of state-of-the-art T_kML methods.

There are several directions in which we would like to further improve our current methods. First, we only consider white-box attacks in the current work, it is natural to extend similar attacks to the black-box setting in which we do not have detailed knowledge of the model to be attacked. Furthermore, labels are not independent, and targeted attacks tend to be easier for semantically distant labels, *e.g.*, it is probably easier to change label `Dog` to `Cat` than `Airplane`. Therefore, in our subsequent work, we would like to consider the semantic dependencies in designing more effective attacks to T_kML algorithms. We will also study defenses against such attacks as an important future work.

Acknowledgments. This research was developed with funding from the National Science Foundation under Grant No. IIS-2103450.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 1, 3

[2] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 3

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1, 3, 6, 11

[4] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 2, 5

[5] Yanbo Fan, Siwei Lyu, Yiming Ying, and Baogang Hu. Learning with average top-k loss. In *Advances in neural information processing systems*, pages 497–505, 2017. 3, 4

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 1, 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[8] Shu Hu, Yiming Ying, Siwei Lyu, et al. Learning by minimizing the sum of ranked range. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 4

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017. 3, 11

[10] Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k multiclass svm. *Advances in neural information processing systems*, 2015. 2

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[12] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations*, 2017. 1

[13] Siwei Lyu and Yiming Ying. A univariate bound of area under roc. In *Proceedings of the Conference on Uncertainty on Artificial Intelligence (UAI)*, 2018. 2

[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. 1, 11

[15] Stefano Melacci, Gabriele Ciravegna, Angelo Sotgiu, Ambra Demontis, Battista Biggio, Marco Gori, and Fabio Roli. Can domain knowledge alleviate adversarial attacks in multi-label classifiers? *arXiv preprint arXiv:2006.03833*, 2020. 3

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 3, 4

[17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 3

[18] Wlodzimierz Ogryczak and Arie Tamir. Minimizing the sum of the k largest functions in linear time. *Information Processing Letters*, 85(3):117–122, 2003. 3

[19] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1

[20] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1400–1409, 2021. 5

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[22] Qingquan Song, Haifeng Jin, Xiao Huang, and Xia Hu. Multi-label adversarial perturbations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1242–1247. IEEE, 2018. 3, 5, 6, 10

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 1

[25] Nurislam Tursynbek, Aleksandr Petiushko, and Ivan Oseledets. Geometry-inspired top-k adversarial perturbations. *arXiv preprint arXiv:2006.15669*, 2020. 3, 6

[26] Zekun Zhang and Tianfu Wu. Learning ordered top-k adversarial attacks via adversarial distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 776–777, 2020. 3, 6

[27] Nan Zhou, Wenjian Luo, Xin Lin, Peilan Xu, and Zhenya Zhang. Generating multi-label adversarial examples by linear programming. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 3, 10