Multi-Label Adversarial Perturbations

Qingquan Song, Haifeng Jin, Xiao Huang, Xia Hu
Department of Computer Science and Engineering, Texas A&M University
{song_3134,jin,xhuang,xiahu}@tamu.edu

Abstract—Adversarial examples are delicately perturbed inputs, which aim to mislead machine learning models towards incorrect outputs. While existing work focuses on generating adversarial perturbations in multiclass classification problems, many real-world applications fall into the multi-label setting, in which one instance could be associated with more than one label. To analyze the vulnerability and robustness of multi-label learning models, we investigate the generation of multi-label adversarial perturbations. This is a challenging task due to the uncertain number of positive labels associated with one instance, and the fact that multiple labels are usually not mutually exclusive with each other. To bridge the gap, in this paper, we propose a general attacking framework targeting multi-label classification problem and conduct a premier analysis on the perturbations for deep neural networks. Leveraging the ranking relationships among labels, we further design a ranking-based framework to attack multi-label ranking algorithms. Experiments on two different datasets demonstrate the effectiveness of the proposed frameworks and provide insights of the vulnerability of multi-label deep models under diverse targeted attacks.

I. INTRODUCTION

Adversarial machine learning, which aims to enhance the security of machine learning models in adversarial settings, has attracted a lot of attentions recently [1], [2]. An intriguing property shared among vast models, especially for deep neural networks, is their vulnerabilities to adversarial examples [3]. Existing work has been widely conducted in generating adversarial examples for multiclass classification algorithms [4], [5]. Most of them are gradient-based and have shown effectiveness on both targeted and non-targeted attacks. However, in many real-world applications, an instance may be associated with multiple labels [6]. For example, in text categorization, a document may cover a wide range of topics, such as politics, economics, and diplomacy [7]; in image classification, a neural scene image can contain both fields and mountains [8]. Different from multiclass cases in which classes are mutually exclusive and a class can only be assigned to one class, in multi-label learning, each instance could associate with multiple labels, thus leading to more opportunities for attackers and larger uncertainties for defenders.

Multi-label adversarial examples widely exist in both malicious and benign activities. From attackers’ perspective, they would like to alter certain labels while keeping some others unchanged for more pertinent attacks or camouflage [9]. From a benign perspective, administrators can leverage intentional perturbations to enhance the protection of users’ privacy [10]. Adversarial examples can also be utilized for adversarial learning to enhance the robustness of multi-label models [11]. Motivated by these observations, in this paper, we propose to investigate a novel and important problem, i.e., multi-label adversarial perturbation generation.

Generating targeted multi-label adversarial perturbations is still a rarely touched and challenging task. First, each instance might have an uncertain number of labels. It requires the attacking methods to be sensitive and discriminative in generating perturbations based on different targets. Second, coordinating multiple labels is difficult since labels are not mutually exclusive with each other. It is too arbitrary to target on a certain single label without considering the rests. Third, quantifying the attacking performance can be hard since multiple targeted labels may not be jointly achieved. These challenges prevent exiting attacking methods from being simply applied to generate multi-label adversarial perturbations.

To tackle these challenges, in this paper, we propose a general attacking framework targeting on multi-label classification models, and conduct premier analysis on the perturbations for deep neural networks. Leveraging the ranking relationships among labels, we further introduce a ranking-based framework to attack multi-label ranking models. Through exploring the attacking performance targeting on manipulating different labels, we empirically validate the effectiveness of our frameworks and provide evidence for the vulnerability of multi-label deep learning models. Specifically, we aim to answer the following questions: (1) How to generate multi-label adversarial perturbations? (2) What is the performance of the generated perturbations based on different attacking strategies? The main contributions are summarized as follows:

- Formally define multi-label adversarial examples;
- Propose a general framework and two corresponding methods to generate targeted multi-label adversarial perturbations for multi-label classification;
- Propose a variation framework and two corresponding methods targeting on attacking multi-label ranking;
- Examine the vulnerability of deep models in multi-label learning using various attacking methods and strategies.

II. PRELIMINARIES

Notations: Assume we have a multi-label classification problem with \( l \) labels. Let \( x \in \mathbb{R}^{d \times 1} \) and \( y \in \{-1, 1\}^{l \times 1} \) denote the feature vector and label vector of an instance, respectively. In general, multi-label classification algorithms provide two types of outputs, either binary values that assign concrete relationships between instances and labels or confidence scores indicating the relevance of each instance with labels. We use function \( H : \mathbb{R}^d \rightarrow \{-1, 1\}^l \) to denote a multi-label classifier which generates the first-type outputs, and \( F : \mathbb{R}^d \rightarrow \mathbb{R}^l \)
to denote a multi-label predictor which predicts continuous relevance scores corresponding to the second-type outputs. Both \( H \) and \( F \) can be decomposed as \( n \) sub-functions, i.e., \( H = \{ h_1, \ldots , h_l \} \) and \( F = \{ f_1, \ldots , f_l \} \), where \( h_j (x) \) and \( f_j (x) \) indicate their predictions of \( y_j \). Based on certain classification thresholds, it is easy to induce \( H \) from \( F \), e.g., \( h_j (x) = I_{f_j (x) > t_j (x)} - I_{f_j (x) < t_j (x)} \), where \( I \) is an indicator function. Following the notations, we formally define two categories of multi-label adversarial examples.

Given an instance \( x \), let \( H \) be a classifier satisfying \( H(x) = y \), where \( y \) is the ground truth labels of \( x \). Here, we assume that \( H \) can correctly classify all labels of \( x \). Though it may not be true in practice, one could simply achieve this by limiting \( L \) to cover only the labels that \( H \) correctly classifies \( x \) into.

Then we have the following two cases:

**A Non-targeted Multi-label Adversarial Example** of \( H \) around \( x \) is defined as an instance \( x^* \) that satisfies:

1. \( x^* \) is close to \( x \) under a certain distance measure;
2. \( x^* \) has the same ground truth labels with \( x \);
3. \( H(x^*) \neq y \), i.e., \( \exists i \in L \) such that \( h_i(x^*) \neq y_i \).

**A Targeted Multi-label Adversarial Example** of \( H \) around \( x \) is defined as an instance \( x^* \) that satisfies:

1. \( x^* \) is close to \( x \) under a certain distance measure;
2. \( x^* \) has the same ground truth labels with \( x \);
3. Given label sets \( A \) and \( B \), and \( h_a (x^*) \neq y_a, \forall a \in A \);
   \( h_b (x^*) = y_b, \forall b \in B \). \( (A \neq \emptyset, A \cup B \subset L, A \cap B = \emptyset) \)

For non-targeted adversarial examples, there is no specific label to be manipulated, which is similar to the non-targeted attack in multi-class cases; but for targeted examples, two label sets \( A \) and \( B \) are specified corresponding to the labels we intend to vary and the labels we expect to fix. As each label is binary, there is no need to specify the values we aim to achieve for each \( h_a (x^*) \), \( \forall a \in A \). In the rest of the paper, we focus on investigating the targeted cases, since they are more practical and meaningful in real-world systems [5]. For the ease of presentation, we use “adversarial example generation” and “adversarial perturbation generation” interchangeably.

**III. Targeted Multi-label Adversarial Perturbation Generation**

To cope with general multi-label settings while conducting effective attacks, we propose two joint attacking frameworks towards attacking multi-label learning models with classification and ranking purposes, respectively. We first introduce a general classification-targeted framework, which aims to manipulate a specific set of predicted labels in the multi-label classification problem. The key idea is to construct a new classification problem by reversing the relationship between instances and classifiers, and generate perturbations through optimization. Motivated by the broad applications of multi-label ranking techniques, we then tailor the framework towards a ranking-targeted framework to attack models with ranking purpose. The core idea is to rerank the predicted scores based on the targeted labels to construct a new multi-label ranking problem. For both frameworks, we separately design two specific methods in subsequent sections.

| Ground Truth Division | Attack Division | \( A \) | \( B \) | \( C = L \setminus A \cup B \) |
|-----------------------|----------------|------|------|----------------|
| \( Y_1 \) = \{ \( y_i \mid y_i = 1, i \in L \) \} | \( A_1 \) | \( B_1 \) | \( C_1 \) |
| \( Y_{-1} \) = \{ \( y_i \mid y_i = -1, i \in L \) \} | \( A_{-1} \) | \( B_{-1} \) | \( C_{-1} \) |

A. The General Frameworks of the Classification-targeted and Ranking-targeted Attacks

In this section, we introduce the two types of frameworks we proposed for attacking multi-label classification models and multi-label ranking models, respectively.

1) Type I. Classification-targeted Framework: We first investigate generating adversarial examples for multi-label classification models. Since each label is either \(-1\) or \(1\), for a targeted classifier \( H \) and a benign instance \( x \), our goal can be mathematically formulated as follows,

\[
\min \| r \|
\]  
subject to \( h_a (x + r) = -y_a, a \in A \), \( h_b (x + r) = y_b, b \in B \),

where \( r \) is the expected perturbation. \( x^* = x + r \) is the generated adversarial example. \( \| \cdot \| \) denotes a certain norm which is usually defined as the \( L_p \) norm, where \( p \) is chosen based on specific settings or demands [5].

2) Type II. Ranking-targeted Framework: Existing work on the ranking-based prediction techniques has demonstrated their effectiveness in both multi-label classification and ranking tasks [12]. It motivates us to explore the corresponding adversarial examples to evaluate the robustness of these models. A straightforward way to attack them is to generate examples based on our first type of framework. However, two problems may perplex it. (1) Soft thresholds: since ranking-based models aim at producing ranking relationships among labels, no hard thresholds need to be specified. (2) Label relationships: the classification-targeted framework does not explicitly take the relationships among labels into account which may not be generalizable for attacking ranking-based algorithms. To tackle the issues, we propose a ranking-targeted framework. Mathematically, given a predictor \( F \), we target at,

\[
\min \| r \|
\]  
subject to \( f_\alpha (x + r) \leq f_\beta (x + r) \leq f_\gamma (x + r), \forall \alpha \in A_1 \cup B_{-1}, \beta \in A_{-1} \cup B_1, \gamma \in C \).

These constraints are motivated by two divisions of the whole label set \( L \). As shown in Table I, the header sets of the row and column represents the two types of divisions, i.e., the first column denotes a division using the ground truth labels of instance \( x \) and the first row denotes a division using the attacking sets \( A \) and \( B \). Each grid in the table denotes the intersection of corresponding header sets of its row and column, e.g., \( A_1 = A \setminus Y_1 \). Labels in the set \( A_{-1} \cup B_1 \) should be positive after attacking, and labels in the set \( A_{1} \cup B_{-1} \) should be negative. The predictions of the unconcerned labels...
\[ \gamma \in C \] are put in the middle to reinforce the attacking ability. It could accentuate the ranking gap between labels on the two sides of the inequality. This constraint may become pretty harsh especially when \(|A| + |B| \ll |C|\), where \(| \cdot |\) represents the number of elements in a given set. To obtain relatively mild constraints, we can limit \(\gamma\) within other label sets such as \(C_{-1}, C_{1}\), or 0, or directly fix a hard threshold.

**B. Type I. Attack Multi-label Classification**

In this section, we propose two attacking methods based on the Type I framework. To avoid the intricacy induced by classification thresholds, we fix certain thresholds and express the constraints using predictor \(F\). For example, for a linear predictor \(F(x)\), the corresponding classifier can be induced with threshold 0 as \(H(x) = \text{sgn}(F(x))\), where \(\text{sgn}(\cdot)\) is the signum function. So the constraints in Equation (1) are equivalent to Equation (3a). For a neural network with sigmoid output layer and 0.5 threshold (i.e., \(H(x) = \text{sgn}(F(x) - 0.5)\)), the equivalent constraints are described in Equation (3b).

\[
y_0f_1(x + r) \leq 0, \quad -y_0f_0(x + r) \leq 0; \\
y_0f_1(x + r) \leq 0.5y_0, \quad -y_0f_0(x + r) \leq -0.5y_0. 
\]

(3a)\hspace{1cm} (3b)

Therefore, Equation (1) could be transformed as follows:

\[
\text{minimize } \|r\| \text{ subject to } y' \odot F'(x + r) \leq c, 
\]

(4)

where \(y' = [y_{01}, \ldots, y_{0|A|}, -y_{01}, \ldots, -y_{0|B|}]^T\) and \(F' = [f_{01}, \ldots, f_{0|A|}, f_{11}, \ldots, f_{1|B|}]^T\). \(\odot\) is the Hadamard product. Vector \(c \in \mathbb{R}^{|A|+|B|}\) is defined based on the thresholds, targeted labels, and the predictor model we attack, e.g., \(c = 0\) represents linear predictors with threshold 0.

1) **Multi-label Carlini & Wagner Attack (ML-CW):** A straightforward way to solve Equation (4) is to convert the constraints to regularizers such as using the hinge loss:

\[
\text{minimize } \|r\| + \lambda \sum_{i=1}^{|A|+|B|} \max(0, y'_i F'_i(x + r) - c_i), 
\]

(5)

where \(\lambda\) is a trade-off penalty between the perturbation size and attacking accuracy. The hinge loss is chosen here to select \(\lambda\) for each constraint [5]. Since we target on the white-box attack here, the optimization can be done based on the algorithms we target on attacking, e.g., gradient decent methods for neural networks. To alleviate the influence of the hyperparameter selection, we employ binary search [5] to select \(\lambda\). For each \(\lambda\), the best perturbation is chosen as the one that satisfies the largest number of constraints in Equation (4). If two perturbations satisfy the same number of constraints, the one with a smaller distortion is better.

2) **Multi-label DeepFool Attack (ML-DP):** Considering the high nonlinearity of the constraints, a variation of solving Eq. (4) is to utilize the linear approximation of \(F'\) to linearized the constraints as: \(y' \odot (F'(x) + \left[\frac{\partial F'}{\partial x_k}\right] r) \leq c\), where \(\frac{\partial F'}{\partial x_k} = \left[\frac{\partial f_{01}}{\partial x_k}, \ldots, \frac{\partial f_{|A|1}}{\partial x_k}, \frac{\partial f_{11}}{\partial x_k}, \ldots, \frac{\partial f_{|B|1}}{\partial x_k}\right] \in \mathbb{R}^{d \times (|A|+|B|)}\). Similar linearization of constraints have been proved to be effective in the multiclass attacking [4]. However, generalizing them to the multi-label attack is not straightforward since multiple labels bring more complex restrictions to the perturbation \(r\).

We provides a simple approach here by solving the set of underdetermined linear equations. It is easy to see that the above problem is equivalent to minimizing \(r\) under a set of linear constraints with varying coefficients:

\[
\text{minimize } \|r\|_2^2 \text{ subject to } P(x) \cdot r \leq q(x), 
\]

(6)

where \(P(x) = (1_d \times y')^T \odot \frac{\partial F'(x)}{\partial x}\) and \(q(x) = c - y' \odot F'(x)\). Here we measure the perturbation with \(L_2\) norm for its simplicity and generality. The optimization is quite difficult because \(P(x)\) and \(q(x)\) vary with the changing of the perturbation. As feature dimension \(d\) is usually larger than the number of labels \(l\), i.e., \(d > |A| + |B|\), the system is underdetermined for any fixed coefficients. Since in each iteration, the coefficients are uncorrelated with the current sub-perturbation \(r\), we propose to solve it in a greedy manner using a pseudo-inverse of \(P(x)\). As this greedy algorithm does not guarantee a convergence to the optimal perturbation or may not even converge, we fix the maximum number of iterations and select the optimal perturbations by jointly consider the number of constraints they satisfied and the size of their distortions.

**C. Type II. Attack Multi-label Ranking**

In this section, we propose two more methods leveraging the relationships among labels based on the Type II framework.

1) **Rank I Attack:** Assume \(\Omega^- = A_{-1} \cup B_{-1}, \Omega^+ = A_{+1} \cup B_1, \) and \(\Omega^0 = C\). Motivated by the empirical multi-label ranking loss [13], we proposed to minimize the average fraction of misordered label pairs. By adopting hinge loss similar to ML-CW, an alternative loss can be formulated as:

\[
L_0 = \|r\| + \frac{1}{|\Omega^-| + |\Omega^+|} \sum_{(x,y) \in \Omega^-} \max(0, c^y - c^x) - e^y_r(x + r) \\
+ \frac{1}{|\Omega^-| + |\Omega^+|} \sum_{(x,y) \in \Omega^+} \max(0, c^x - c^y) - e^x_r(x + r) \\
+ \frac{1}{|\Omega^-| + |\Omega^+|} \sum_{(x,y) \in \Omega^0} \max(0, c^y_r(x + r) - e^x_r(x + r)).
\]

(7)

The exponential function is chosen to severely penalize the ranking errors inspired by [14]. To reduce the time complexity without losing much attack power, Equation (7) could be simplified by extracting the maximum and minimum predictions of labels in each label set as follows:

\[
L_1 = \|r\| + \lambda_1 \max(0, e^{y_r(x + r)} - e^{x_r(x + r)}) \\
+ \lambda_2 \max(0, e^{x_r(x + r)} - e^{y_r(x + r)}) \\
+ \lambda_3 \max(0, e^{x_r(x + r)} - e^{y_r(x + r)})
\]

(8)

where \(\lambda_i (i = 1, 2, 3)\) are hyperparameters that make a trade-off among terms. This loss function could be further simplified based on the choice of \(\Omega^0\). If \(\Omega^0 \neq \emptyset\), the first term could be reduced; otherwise the last two terms are nonexistent.
TABLE II: Original accuracy of model to be attacked.

| Dataset | Hamming | macro/micro-F1 | Ranking Loss | mAP |
|---------|---------|----------------|--------------|-----|
| VOC 2007 | 0.0504 | 0.7278/0.7182 | 0.0175 | 0.9239 |
| VOC 2012 | 0.0491 | 0.7340/0.7252 | 0.0166 | 0.9320 |

2) Rank II Attack: Though to some extent, Rank I attack takes the label correlation into account, it may not be sensitive enough in certain cases. For example, if we have a benign instance $x$ with ground truth labels $[-1, 1, 1]^T$ and targeted attacking labels $[-1, -1, 1]^T$, the label probabilities predicted by the predictor $F$ could be $[0.01, 0.98, 0.99]^T$. In this case, Rank I cannot provide successful attack since Equation (8) is equal to 0 for the benign instance $x$ and no loss is defined. This problem comes from the rank-constraint of the general framework defined in Equation (2). To address this issue, no discrimination is made for labels in set $\Omega^+ = A_1 \cup B_1$. Similar situation happens for $\Omega^- = A_1 \cup B_1$. To solve this problem, we add two more constraints as follows:

$$f_{\alpha_1}(x + r) \leq f_{\alpha_2}(x + r), \quad \forall \alpha_1 \in A_1, \alpha_2 \in B_1,$$

$$f_{\beta_1}(x + r) \leq f_{\beta_2}(x + r), \quad \forall \beta_1 \in B_1, \beta_2 \in A_1. \quad (9)$$

The first constraint forces the probabilities of newly added negative labels to be smaller than all negative labels, while the second highlights the new positive labels. Leveraging the two constraints, the loss of Rank II attack is defined as:

$$L_2 = L_1 + \lambda_4 \max(0, \max_{\alpha_1 \in A_1} e^{f_{\alpha}(x + r)} - \min_{\alpha_2 \in B_1} e^{f_{\alpha_2}(x + r)}) + \lambda_5 \max(0, \max_{\beta_1 \in B_1} e^{f_{\beta}(x + r)} - \min_{\beta_2 \in A_1} e^{f_{\beta_2}(x + r)}), \quad (10)$$

where $L_1$ is the loss function of Rank I attack. Similar to ML-CW, we use binary search for both Rank I and Rank II to determine a suitable $\lambda$. To select the best perturbation, after finding the optimal perturbation $r_\lambda$ for each $\lambda$, we utilize Kendall $\tau_b$ ranking correlation coefficient [15] to quantify the similarity between the new label prediction $y_\lambda^*$ and a criterion vector $y^* \in \{-2, -1, 0, 1, 2\}^L$. $y^*$ is defined based on the rank constraints in Eq. (2) and (9), i.e., for any $i \in 1, 2, \ldots, L$, $y^*_i = -2, -1, 0, 1, 2$ correspond to $i \in A_1, B_1, C, B_1, A_1$, respectively. The higher the similarity is, the better the perturbation fits the constraints.

IV. EXPERIMENTS

Two major aspects are analyzed: Q1: What is the general performance of different methods? Q2: What is the performance of different methods on attacking specified labels?

Datasets: Experiments are conducted on two benchmark datasets, i.e., PASCAL VOC 2007 [16] and VOC 2012 [17].

A. Targeted Multi-label Learning Model

We focus on attacking deep neural networks given its superior performance in multi-label learning and vulnerability to adversarial examples [3]. The classifiers (and predictors) are built upon the Inception v3 network [18] pre-trained on ImageNet dataset [19]. To apply them to multi-label cases, we retrain the model by replacing softmax layers with sigmoid classification layers. Both the instance-wise and label-wise losses are considered [20] to ensure prediction performance. After retraining, the final testing performance regarding five commonly used measures [20] is reported in Table II.

B. Attacking Methods

Besides four proposed methods: ML-CW, ML-DP, Rank I and Rank II, we extend two widely adopted multiclass attacking baselines to multi-label settings as follows:

- **Targeted Fast Gradient Sign Method (FGS)** [21] is extended to multi-label case via:
  $$x^* \leftarrow clip(x - \epsilon \cdot \text{sgn}(\nabla_x loss(-y', H'(x)))),$$
  where $\epsilon$ controls distortions, $clip(x)$ clips each pixel of images to the range of $[0, 255]$, $y'$ is defined in Eq. (4), $H'$ is the classifier corresponding to predictor $F'$ defined in Eq. (4), and $Loss$ is the sigmoid cross entropy loss.

- **Targeted Fast Gradient Method (FG)** is a variation of FGS using $L_2$ normalization defined as follows:
  $$x^* \leftarrow clip(x - \epsilon \cdot \frac{\nabla_x loss(-y', H'(x))}{\|\nabla_x loss(-y', H'(x))\|_2}).$$

Parameter Setting: We set the initial $\lambda$ as $10^5$ and apply the binary search ten times for the proposed methods. To avoid the optimization getting stuck, we follow similar image preprocessing and variable transformation methods discussed in [5]. For ML-DP, the maximum iteration is set as 20 to prevent over distortion. For FGS and FG, $\epsilon$ is set to have similar distortion with the best method (either ML-CW or Rank II) based on the attacking performance. $L_2$ distortion norm is used for the proposed approaches.

C. Evaluation Metrics

Three types of metrics are employed. **Distortion**: root mean square deviation (RMSD) [22] is used to measure the perturbation sizes. **Classification**: we set the classification threshold as 0.5 for each label, i.e., $h_i(x^*) = 1$ if $f_i(x^*) \geq 0.5$. The classification performance is measured by Hamming Loss and instance-F1 score (F1) [20]. Hamming loss is applied on two targeted label sets $A$ and $B$. Since we focus on targeted attacks, the targeted labels are considered as the ground truth in classification attacks. **Ranking**: the ranking performance is measured by four metrics, i.e., Ranking Loss, mean average precision (mAP), instance average area under curve (AUC), and Kendall $\tau_b$ rank correlation coefficient. For clarity, an arrow sign $\uparrow$ or $\downarrow$, is annotated behind each metric in tables indicating the higher the better ($\uparrow$), or the lower the better ($\downarrow$).

D. Experimental Settings

We fix the classification threshold as 0.5 and collect target attacking images into a set $X$. Every label of these images should be correctly classified by $H$. The attacking strategies in two experiments are introduced as follows.

**General Attacking:** We first try two types of attacking strategies to test the general performance of six methods:

- **Random Case:** Randomly select 1000 images from $X$. For each image $x$, randomly select one positive and one
negative label from sets $Y_1$ and $Y_{-1}$ defined in Table I as the targeted changing set $A$, and put the rest in $B$.

- Extreme Case: Randomly select 100 images from $X$. Change all labels of them, i.e., $A = L$ and $B = \emptyset$.

Label Specified Attacking: Two more strategies are used to evaluate six methods on attacking specific labels. To avoid lacking of test examples caused by label imbalance, we choose to perturb each positive labels and one of them should be “person”. For a specific set $A$ and $B$, we randomly select 100 images from $X$, and each image should have at least two positive labels and one of them should be “person”. Set $A = \{\text{person}\}$ and $B = L \setminus \{\text{person}\}$.

- Person Reduction (Person): Randomly select 100 images from $X$. Each image should have at least two positive labels and one of them should be “person”. Set $A = \{\text{person}\}$ and $B = L \setminus \{\text{person}\}$.

- Sheep Augmentation (Sheep): Randomly select 100 images from set $X$. Each of them should not have label “sheep”. Set $A = \{\text{sheep}\}$ and $B = L \setminus \{\text{sheep}\}$.

### E. General Attacking Performance

We first compare the performance of all six methods with general attacking strategies. From results shown in Table III, the main observations are described as follows.

#### Distortion Analysis.

We observe that: (1) ML-CW method provides the best RMSD in the “Random Case”, but becomes worse than Rank I and Rank II in “Extreme Case”. This is because the classification constraints used by ML-CW is easier to achieve than ranking-based constraints when the number of labels we need to change, i.e., $|A|$, is small, but will become much harder to achieve when $|A|$ is large. (2) The sizes of the perturbations generated by Rank I & II are relatively stable in both cases. It is because the ranking-based attacks only care about the ranking relationships among labels without considering specific classification threshold. When $|A|$ is small, this might be harder to achieve compared with the classification constraints used by ML-CW and will cause the distortion to be larger. But when $|A|$ is large as in the “Extreme Case”, the soft ranking thresholds would become milder than hard classification thresholds. (3) ML-DP generates the largest perturbations in all cases because multiple linear constraints are hard to be jointly accommodated in multi-label settings.

#### Attack Classification.

From the results measured by three classification metrics, we can see that: (1) The baselines FG and FGS do not perform well. This shows that one-shot methods are not good at generating multi-label adversarial examples. (2) In general, ML-CW performs the best among all six methods. Rank II performs comparably well in the “Random Case”, but becomes worse in the “Extreme Case”. This is because the ranking-based methods do not consider the hard classification threshold 0.5 during the optimization while ML-CW does on the contrary. It should be noted that this does not mean Rank II cannot perform well in this case. As mentioned in Section III-A2, Rank II could also take the classification thresholds into account by setting fixed thresholds in constraints. (3) Rank II outperforms Rank I in the “Random Case”, but becomes slightly worse in the “Extreme Case”. By comparing their loss functions, we know that Rank II has more constraints in discriminating the labels within sets $\Omega^{-1}$ and $\Omega^+$. However, more ranking constraints bring more soft thresholds, which make it worse in the “Extreme Case”.

#### Attack Ranking.

Three main observations can be summarized from Table III. (1) The baselines FG and FGS cannot provide satisfying attacks due to their poor capacities in accommodating multiple labels. (2) Rank II outperforms all other methods. Comparing with Rank I, it enjoys more hierarchies among labels, i.e., smaller range in $\Omega^{-1}$ and $\Omega^+$. However, more ranking constraints bring more soft thresholds, which make it worse in the “Extreme Case”. We can also see that Rank II outperforms Rank I in the “Random Case”, but becomes slightly worse in the “Extreme Case”. By comparing their loss functions, we know that Rank II has more ranking constraints in discriminating the labels within sets $\Omega^{-1}$ and $\Omega^+$. However, more ranking constraints bring more soft thresholds, which make it worse in the “Extreme Case”.

### F. Label Specified Attacking Performance

Two more specific attacking strategies introduced in Section IV-D, i.e., “Person Reduction” and “Sheep Augmentation”, are adopted here to provide a more exquisite evaluation.
We focus on the best three methods ML-CW, Rank I, and Rank II. Since different metrics reflect similar conclusions, we only report several representative ones in Table IV for space limitation. To give a more visualized comparison, we display four probability distributions of the two target labels of all adversarial images generated by different methods in Fig. 1. The X-axis represents the predictions of “person” or “sheep”. The Y-axis denotes the number of images. Based on Table IV and Fig. 1, we analyze the three methods in turn.

ML-CW. Two major observations are found. (1) ML-CW performs well in both cases since the number of labels we want to vary (i.e., \(|A|\)) is small. (2) As shown in Fig. 1, ML-CW successsfully decreases the “person” predictions of most images below the classification threshold 0.5, and increases the predictions of “sheep” beyond 0.5. However, the mean values of the distributions in both cases are close to 0.5 because ML-CW only considers the linear error between predictions and the thresholds, which lacks discriminative power.

Rank I Attack. From Table IV, we find that the perturbations generated by Rank I are pretty small in “person” case. This validates our discussion in Section III-C2, i.e., as long as “person” ranks lower than other originally positive labels and higher than the rest, the optimization will stop, which causes the distortion being small but the probability of “person” remaining high. This explanation can be further verified by Fig. 1. For both cases, the predictions of the adversarial images generated by Rank I (red bars) have high overlap with the predictions of the original images (blue bars).

Rank II Attack. Two main conclusions are drawn. (1) In Fig. 1, Rank II not only successsfully decreases the predictions of “person” below the classification threshold, but also induces a huge discrimination between the mean value and the classification threshold. Similar results happen in the “sheep” case. (2) For the classification attack, Rank II performs better in eliminating “person” than augmenting “sheep”. The reason is that every image in the dataset has much more negative labels than positive ones. Due to the lack of positive labels, during the optimization process in “sheep” attack, it is hard to guarantee that there will always be some labels whose predicted probabilities are larger than 0.5. Thus, even “sheep” ranks the highest among all labels, its prediction may still be smaller than 0.5 with high probability.

V. CONCLUSION AND FUTURE WORK

In this paper, we focus on generating multi-label adversarial examples and propose two general frameworks targeting on attacking multi-label classification and ranking models, respectively. For each type of the framework, we propose two specific methods. Experiments with different attacking strategies on deep neural networks validate the effectiveness of our proposed methods and the vulnerability of multi-label deep models. Further theoretical analysis on different perturbations may enhance the interpretability and security of multi-label learning models, which is intriguing for future exploration.

VI. ACKNOWLEDGEMENTS

This work is, in part, supported by DARPA (#W911NF-16-1-0565) and NSF (#IIS-1657196 and #CNS-1816497).