# Compositional Targeted Multi-Label Universal Perturbations

Hassan Mahmood
Northeastern University
mahmood.h@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

## Abstract

*Generating targeted universal perturbations for multi-label recognition is a combinatorially hard problem that requires exponential time and space complexity. To address the problem, we propose a compositional framework. We show that a simple independence assumption on label-wise universal perturbations naturally leads to an efficient optimization that requires learning **affine convex cones** spanned by label-wise universal perturbations, significantly reducing the problem complexity to **linear** time and space. During inference, the framework allows generating universal perturbations for novel combinations of classes in **constant time**. We demonstrate the scalability of our method on large datasets and target sizes, evaluating its performance on NUS-WIDE, MS-COCO, and OpenImages using state-of-the-art multi-label recognition models. Our results show that our approach outperforms baselines and achieves results comparable to methods with exponential complexity. The code is available at https://github.com/hassan-mahmood/UMLLAttacks.git*

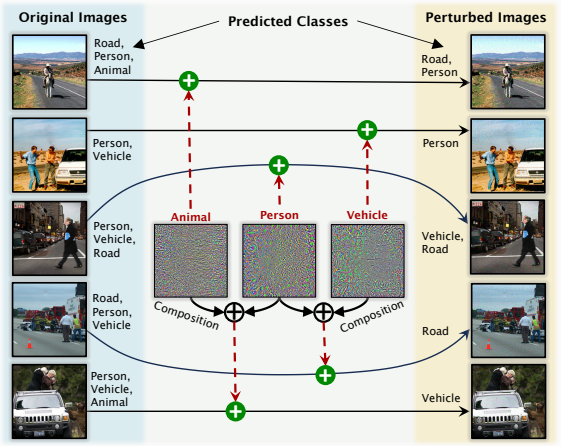


Figure 1. **Targeted Multi-Label Universal Perturbations**. Our framework allows composing (shown as ⊕) different label-specific universal perturbations ('Animal','Person','Vehicle') to attack multiple classes together. These perturbations, once added (shown as ⊕) to an image, cause target labels to be misclassified.

## 1. Introduction

Deep neural networks have been very successful at image-recognition, demonstrating high accuracy in various applications. However, they are also vulnerable to adversarial attacks [9, 25, 30, 31, 43, 74, 84, 86], where small changes to the input image can cause the model to misclassify the image. Notably, Moosavi-Dezfooli et al. [67] discovered that a single perturbation, called universal adversarial perturbation (UAP), can be crafted to misclassify a large number of images. This motivated a large body of work on generating universal adversarial attacks [7, 36, 38, 65, 69, 110].

Existing works on adversarial attacks have mainly focused on multi-class recognition (MCR) whose goal is to identify a single class in an image. On the other hand, many real-world applications require identifying all classes in an image e.g., document classification [75], scene understanding [85], facial attribute recognition [26], image annotation [15], human-object interaction [33], and several safety and security-critical applications e.g., recognizing diseases [24], autonomous driving [47], and surveillance [57, 108]. Therefore, the goal of multi-label learning (MLL) is to identify all labels present in an image [14, 20, 23, 39, 40, 55, 77, 100, 115]. Given the increasing number of applications relying on MLL, it is crucial to understand the internal workings of the MLL models, identify their vulnerabilities, and make them robust. Recently, MLL models have been shown to be vulnerable to adversarial attacks [1, 2, 42, 63, 88], yet the vulnerability of MLL systems against univeral perturbations has not been explored.

Crafting universal perturbations for MCR and MLL have fundamental differences due to the distinct structuring of decision boundaries in these problems. In MCR, a universal perturbation to turn off a label in images will result in the prediction of another label. Conversely, in MLL, multiple labels can co-exist, and the objective of an attack might be to misclassify any subset of these labels either by turning the labels off or on (see Fig. 2). In MLL, any combination of labels can be present in images. For the label set $\mathcal{C}$, there are $2^{|\mathcal{C}|}$ possible combinations of labels. Universal pertur-

bations identify specific directions in the input space where the classifier is consistently vulnerable to attacks. *What makes this particularly concerning in multi-label learning is that there can be exponentially many directions in the input space to attack the model - corresponding to different combinations of labels.*

One approach for generating universal perturbations for MLL is to generate universal vectors that flip the prediction of all or most labels. However, in practice, the adversary usually seeks to modify a specific small set of labels while keeping the prediction of others unchanged (see Fig. 1). This is called the *targeted attack* and requires learning $2^{|\mathcal{C}|}$ distinct universal perturbation vectors. Learning each such universal vector requires iterating through the dataset, resulting in an exponential time and space complexity.

Existing universal perturbation methods focus on MCR setting [7, 8, 60]. Adapting these methods to MLL would require learning a separate universal perturbation for each possible label combination, incurring exponential complexity. Although one could learn label-wise independent perturbations, no effective method exists to combine them for attacking specific label combinations. We empirically show that naively combining them does not lead to successful attacks. Similarly, generative attack approaches [27, 69, 76] would be ineffective for unseen label combinations.

**Contributions.** In this paper, we develop an efficient method for crafting targeted universal adversarial attacks on multi-label learning (MLL) models, addressing the aforementioned combinatorial complexity problem. Our key assumption is to generate a universal attack on a target set as a composition of universal label-wise attacks of the target labels (see Fig. 1). Based on this assumption, we derive a formulation which requires learning affine convex cones spanned by label-wise universal vectors. Our formulation requires linear training time complexity and after trained, the universal perturbation to attack any combination of labels can be composed in constant time using the label-wise attack vectors. Our key contributions are:

- Existing MCR-based approaches are not inherently designed to attack multiple targeted labels for MLL models. Our work is the first to address this gap, offering an efficient and principled solution for MLL setting;
- Our proposed framework to learn targeted universal perturbations for MLL has *linear time and space complexity* and constructs attacks on any target set in constant time, while achieving performance comparable to the existing oracle methods with exponential complexity;
- Our results visualize and provide deeper understanding of the universal adversarial regions learnt by our method;
- We evaluate our method on NUS-WIDE, MS-COCO, and OpenImages benchmark datasets using state-of-the-art multi-label learning models against various baselines, showing the effectiveness of our method.

## 2. Related Works

### 2.1. Multi-Label Recognition

There has been significant recent interest in multi-label learning due to its applications in various tasks. In MLL, the goal is to identify all concepts/labels in an image. This is particularly a harder task than multi-class recognition, since there can be exponentially many possible combinations of classes present in an image and can be related to each other. To address this problem, early works [90, 97] considered independent binary classifiers to classify the presence or absence of each class. However, since such a method does not consider the correlations among labels, several other approaches have been proposed that exploit label correlations or external knowledge [13, 18, 34, 45, 49, 51, 70, 94–96, 98, 101, 105–107, 113, 118], or use graphical models [50, 52]. More recent works use attention mechanism or novel loss functions to consider the positive and negative sample imbalance [6, 17, 53, 54, 58, 93, 112].

### 2.2. Adversarial Attacks

Adversarial attacks [3, 12, 81, 89, 99] have gained a lot of attention as they identify the key vulnerability of deep neural networks (DNNs) to small imperceptible changes in the input. Researchers have explored various ways that attackers can manipulate data to fool models and developed several defenses to counter these attacks [4, 5, 10, 11, 19, 21, 22, 28, 29, 35, 46, 62, 71–73, 78, 82–84, 91, 92, 102, 114]. Based on the attacker's objective and the target setting, the adversarial attacks can be categorized in different ways, such as targeted and untargeted attacks, white-box (which assume full access to the target model parameters) and black-box attacks (no access to the model parameters), or instance-specific and instance-agnostic attacks.

### 2.3. Universal Adversarial Attacks

In instance-specific attacks, the perturbations are generated and targeted for each input image, whereas, instance-agnostic attacks [38, 65, 69] generate a single universal perturbation that can fool all input instances. These universal adversarial perturbations(UAPs) can be untargeted or targeted. The goal of the untargeted universal perturbation is to perturb any image such that it is misclassified by the target model [67, 76]. In the targeted case, the goal is to learn the universal perturbation such that it misclassifies any given image to a specific class [7, 110]. Universal perturbations were first proposed by Moosavi-Dezfooli et al. [67] and were generated by iteratively computing and aggregating perturbations using DeepFool [66] across multiple inputs. Since then, several methods have been proposed to learn better universal perturbations [60, 109–111]. To learn UAPs with better generalization, Liu et al. [59] proposed a gradient aggregation method to address gradient vanish-

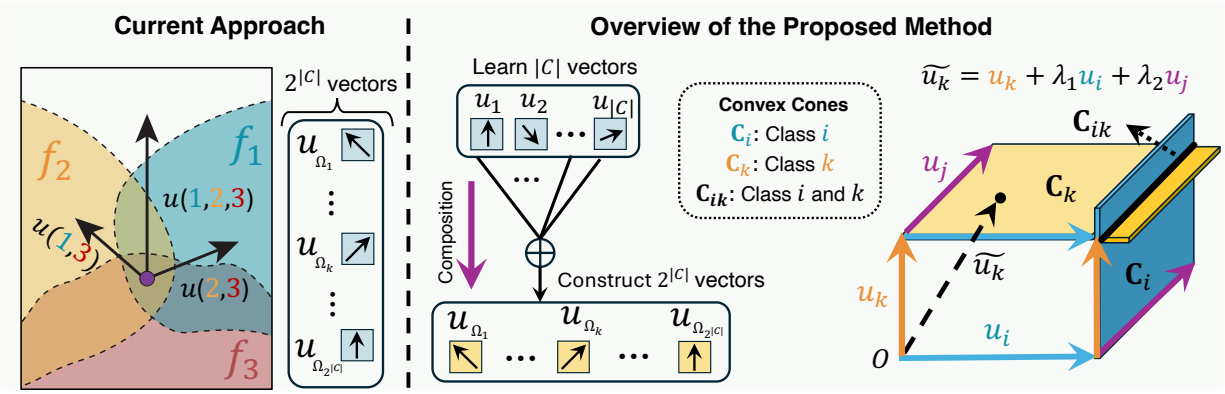


Figure 2. In multi-label recognition, images can have any combination of labels present, leading to a combinatorial explosion in possible universal perturbations. Traditional methods (shown on the left) require exponential time and space complexity to learn perturbations for all class combinations. Our approach (shown on the right) reduces this complexity by learning $|\mathcal{C}|$ affine convex cones, each representing a region of universal perturbations for a specific class. This allows us to efficiently generate perturbations for any combination of classes.

ing problem and Li et al. [48] proposed to utilize instance-specific attack method to generate dominant UAPs.

### 2.4. Multi-Label Adversarial Attacks

Many recent works have studied adversarial attacks for multi-label recognition models [41, 61, 103, 104, 116, 117]. Song et al. [87] proposed white-box multi-label attack framework and Hu et al. [32] proposed to exploit ranking relations to design attacks for top-$k$ multi-label models. Aich et al. [1] proposed a CLIP-based generative model to generate multi-object attacks in the black-box setting. Jia et al. [37] proposed theoretical robustness guarantees to defend against multi-label adversarial attacks and Melacci et al. [64] exploited domain knowledge context to detect multi-label adversarial attacks. Mahmood and Elhamifar [63] proposed an optimization to address the problem of negative gradient correlation in generating multi-label attacks. These works mainly address instance-specific perturbations for multi-label setting. To the best of our knowledge, only Hu et al. [32] proposed universal untargeted attack where a single perturbation was learnt to attack and replace top-$k$ predicted labels across images. In comparison, we address the problem of learning targeted universal perturbations to attack specific combinations of labels.

## 3. Multi-Label Universal Perturbations

### 3.1. Problem Formulation

Consider a multi-label classifier $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{C}|}$ where $\mathcal{C}$ is the set of all labels. For an image $\boldsymbol{x} \in \mathbb{R}^d$, let $\boldsymbol{y} \in \{0,1\}^{|\mathcal{C}|}$ denote the set of its labels, where each entry indicates the absence (0) or presence (1) of the corresponding label in $\mathcal{C}$ in the image. The classifier $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{|\mathcal{C}|}\}$ consists of $|\mathcal{C}|$ binary classifiers (one for each label), where $\mathcal{F}_c(\boldsymbol{x}) \in (-\infty, +\infty)$ represents the *logit* of label $c$. The probability of label $c$ being present in the image is given by $\hat{o}_c = \sigma(\mathcal{F}_c(\boldsymbol{x}))$, where $\sigma(.)$ is the sigmoid function, hence, it is considered present (1), if $\hat{o}_c \geq 0.5$, and absent (0) otherwise. We assume that $\mathcal{F}$ has already been trained on a set of images. Let $\Omega \subseteq \mathcal{C}$ denote the set of target labels that we want to attack. Our goal is to generate a a targeted universal perturbation for $\Omega$.

### 3.2. Multi-Label Universal Perturbations

We want to find a single (universal) perturbation vector, $\boldsymbol{u}_\Omega$, that can attack a set of target labels in $\Omega$ in any image $\boldsymbol{x}$ containing them. One can generate such a vector by maximizing the expected multi-label learning loss for the labels in $\Omega$ across the training dataset, while restricting the norm of $\boldsymbol{u}_\Omega$ (we consider $\ell_\infty$-norm). However, due to correlations among labels in MLL, attacking the target labels may inadvertently affect non-target labels ($\bar{\Omega} = \mathcal{C} \setminus \Omega$). To mitigate this, we seek to minimize the influence of the universal perturbation vector on non-target labels. This requires solving the following **oracle** optimization,

$$P^*_\Omega: \max_{\boldsymbol{u}_\Omega} \sum_{k \in \Omega} \mathcal{H}_k(\boldsymbol{u}_\Omega) - \gamma \sum_{q \in \bar{\Omega}} \mathcal{R}_q(\boldsymbol{u}_\Omega), \tag{1}$$

where $\mathcal{H}_k$ ensures attacking the label $k$ in the target set and $\mathcal{R}_q$ tries to prevent modifying the prediction of non-target labels using the universal perturbation $\boldsymbol{u}_\Omega$. Here, $\gamma$ is a scalar hyperparameter setting a trade-off between the two terms. More specifically, we define $\mathcal{H}_k$ and $\mathcal{R}_q$ as

$$\begin{aligned} \mathcal{H}_k(\boldsymbol{u}_\Omega) &\triangleq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\, \mathcal{L}_{\text{bce}}\big(\mathcal{F}_k(\boldsymbol{x} + \Pi_\epsilon(\boldsymbol{u}_\Omega)),\, y_k\big), \\ \mathcal{R}_q(\boldsymbol{u}_\Omega) &\triangleq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \max\big\{0, -\tanh\big(\mathcal{F}_q(\boldsymbol{x})\mathcal{F}_q(\boldsymbol{x} + \Pi_\epsilon(\boldsymbol{u}_\Omega))\big)\big\}, \end{aligned} \tag{2}$$

where $\Pi_\epsilon$ projects the input on p-norm ball of radius $\epsilon$, $\mathcal{L}_{\text{bce}}$ is the binary cross-entropy loss and the expectation is with respect to samples from the dataset. Therefore, maximizing $\mathcal{H}_k(\boldsymbol{u}_\Omega)$ ensures to find a perturbation that modifies the

prediction of each target label $k$. On the other hand, $\mathcal{R}_k$ aims to minimize the influence of the perturbation ($\boldsymbol{u}_\Omega$) on each non-target label $q$. This is done by incurring a positive penalty when the signs of the logits for the unperturbed and perturbed inputs differ from each other. Thus, minimizing this term leads to the same signs for the logits of each non-target label before and after the attack.

### 3.3. Compositional Multi-Label Universal Attacks (CMLU)

There are $2^{|\mathcal{C}|}$ possible subsets of labels $\mathcal{C}$. Therefore, finding all universal perturbations $\{\boldsymbol{u}_\Omega\}$ by independently solving (1) for each subset $\Omega$ leads to the exponential complexity $O(2^{|\mathcal{C}|})$. We develop a compositional multi-label universal attack (CMLU) that allows us to find any universal perturbation in linear complexity $O(|\mathcal{C}|)$.

Let $\mathcal{P}(\mathcal{C})$ denote the power set of $\mathcal{C}$, i.e., the set of all subsets of $\mathcal{C}$. Our goal is to find all $\boldsymbol{u}_\Omega$'s by solving (1) for every $\Omega \in \mathcal{P}(\mathcal{C})$,

$$\sum_{\Omega\in\mathcal{P}(\mathcal{C})} \max_{\boldsymbol{u}_\Omega} \Big[ \sum_{k\in\Omega} \mathcal{H}_k(\boldsymbol{u}_\Omega) - \gamma \sum_{q\in\bar{\Omega}} \mathcal{R}_q(\boldsymbol{u}_\Omega) \Big]. \tag{3}$$

As mentioned above, solving (3) over the power set of $\mathcal{C}$ has exponential complexity. To tackle this problem, we make the following assumption: each universal perturbation $\boldsymbol{u}_\Omega$ can be written as the sum of atomic universal perturbations

$$\boldsymbol{u}_\Omega \triangleq \sum_{i\in\Omega} \boldsymbol{u}_i, \tag{4}$$

where $\boldsymbol{u}_i$ is the universal perturbation on the label $i$. We also assume that the label-wise universal perturbations are norm-constrained and mutually orthogonal, i.e., $\|\boldsymbol{u}_i\|_p \leq \epsilon$ and $\hat{\boldsymbol{u}}_i^\top \hat{\boldsymbol{u}}_j = 0$ for each $i \neq j$, where $\hat{\boldsymbol{u}} = \boldsymbol{u}/\|\boldsymbol{u}\|_2$.

Let $\boldsymbol{U} = \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_{|\mathcal{C}|}\end{bmatrix}$ represent a matrix whose columns are label-wise universal perturbations and $\boldsymbol{\lambda}_\Omega$ denote a vector whose elements indexed by $\Omega$ are one and other elements are zero. Based on our assumption, we optimize $\boldsymbol{U}$ so that we can construct any $\boldsymbol{u}_\Omega$ using the columns of $\boldsymbol{U}$. Given our assumptions, we can rewrite (3) as

$$\max_{\boldsymbol{U}} \sum_{\Omega\in\mathcal{P}(\mathcal{C})} \sum_{k\in\Omega} \Big( \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_{\Omega\setminus k}) - \frac{\gamma}{|\Omega|} \sum_{q\in\bar{\Omega}} \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_{\Omega\setminus k}) \Big),$$
$$\text{s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I},\ \|\boldsymbol{U}\|_{1,p} \leq \epsilon \tag{5}$$

where $\hat{\boldsymbol{U}}$ is column-normalized $\boldsymbol{U}$ and $\|\boldsymbol{U}\|_{1,p} = \max_{1\leq i\leq|\mathcal{C}|} \|\boldsymbol{u}_i\|_p$, is the maximum norm of the columns of $\boldsymbol{U}$. Notice that we wrote $\boldsymbol{u}_\Omega = \boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_{\Omega\setminus k}$ for a $k$ that belongs to $\Omega$, where $\Omega\setminus k$ denotes removing $k$ from $\Omega$.

One can see that, for any $k \in \mathcal{C}$, there are $2^{|\mathcal{C}|-1}$ subsets that contain $k$. Each of these subsets would contain $k$ and classes from the powerset $\mathcal{P}(\mathcal{C}\setminus k)$. We can rewrite (5) as (see the supplementary materials for derivation details)

$$\max_{\boldsymbol{U}} \sum_{k\in\mathcal{C}} \sum_{\Omega\in\mathcal{P}(\mathcal{C}\setminus k)} \Big( \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_{\Omega\setminus k}) - \eta \sum_{q\in\{\bar{\Omega}\setminus k\}} \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_{\Omega\setminus k}) \Big),$$
$$\text{s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I},\ \|\boldsymbol{U}\|_{1,p} \leq \epsilon, \tag{6}$$

where $\eta = \gamma/|\Omega|$. The power set of $\mathcal{C}$ can be written as the union of all subsets of $\mathcal{C}$: $\mathcal{P}(\mathcal{C}) \triangleq \bigcup_{i=0}^{|\mathcal{C}|} \binom{\mathcal{C}}{i}$ where, each $\binom{\mathcal{C}}{i}$ has all the subsets of $\mathcal{C}$ of size $i$. In our case, $S \in \binom{\mathcal{C}}{i}$ would represent the indices of the columns of $\boldsymbol{U}$. So, we can write the summation across the power set as:

$$\max_{\boldsymbol{U}} \sum_{k\in\mathcal{C}} \sum_{t=0}^{|\mathcal{C}|-1} \sum_{S\in\binom{\mathcal{C}\setminus k}{t}} \Big( \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_S) - \eta \sum_{q\in\{\bar{S}\setminus k\}} \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}_S) \Big),$$
$$\text{s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I},\ \|\boldsymbol{U}\|_{1,p} \leq \epsilon, \tag{7}$$

In (7), for a fixed $k$ and $t$, $\boldsymbol{\lambda}_S$ has exactly $t$ non-zero elements and $S$ iterates through all possible combinations of $\{C\setminus k\}$ of size $t$. Therefore, the sum across all $\boldsymbol{\lambda}_S$ (for $S \in \binom{\mathcal{C}\setminus k}{t}$) would be equal to the sum across all binary vectors $\boldsymbol{\lambda} \in \{0,1\}^{|\mathcal{C}|}$ such that exactly $t$ elements are non-zero (i.e., $\|\lambda\|_0 = t$) **and** $\boldsymbol{\lambda}_k = 0$. Where, $\|\cdot\|_0$ denotes the $\ell_0$-norm, which counts the number of nonzero elements of the given input. So we can write (7) as

$$\max_{\boldsymbol{U}} \sum_{k\in\mathcal{C}} \sum_{t=0}^{|\mathcal{C}|-1} \sum_{\substack{\boldsymbol{\lambda}\in\{0,1\}^{|\mathcal{C}|} \\ \|\boldsymbol{\lambda}\|_0=t, \boldsymbol{\lambda}_k=0}} \Big[ \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \tag{8}$$
$$- \eta \sum_{q\in\{\mathcal{C}\setminus k\}} \delta(\boldsymbol{\lambda}_q = 0) \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \Big] \text{ s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I}, \|\boldsymbol{U}\|_{1,p} \leq \epsilon,$$

where $\delta$ is the indicator function which is 1 if the input condition is True and 0 otherwise. As $\boldsymbol{\lambda}$ is $|\mathcal{C}|$ dimensional, the second loss term in (8) iterates through each index of $\boldsymbol{\lambda}$ i.e., ($q \in \{\mathcal{C}\setminus k\}$) and computes the loss of label $q$ if $\boldsymbol{\lambda}_q = 0$.

We investigate two ways to approximately solve (8). In the first approach, we solve the lower bound of (8) as

$$\text{CMLU}_\alpha\text{: } \max_{\boldsymbol{U}} \sum_{k\in\mathcal{C}} \sum_{t\in\mathcal{T}} 2^{|\mathcal{C}|} \min_{\substack{\boldsymbol{\lambda}\in\{0,1\}^{|\mathcal{C}|} \\ \|\boldsymbol{\lambda}\|_0=t, \boldsymbol{\lambda}_k=0}} \Big[ \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \tag{9}$$
$$- \eta \sum_{q\in\{\mathcal{C}\setminus k\}} \delta(\boldsymbol{\lambda}_q = 0) \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \Big] \text{ s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I}, \|\boldsymbol{U}\|_{1,p} \leq \epsilon$$

Where, $\mathcal{T} = \{0, 1, \cdots, |\mathcal{C}| - 1\}$. (9) requires maximizing the minimal loss. To solve the inner minimization problem, for a fixed $k$ and $t$, we randomly initialize $\boldsymbol{\lambda}$ and iteratively update it for a fixed number of iterations. We then project $\boldsymbol{\lambda}$ on the feasible set by keeping the largest $t$ values as 1 and setting the rest to 0. We then keep the $\boldsymbol{\lambda}$ fixed and solve the outer maximization in (9) w.r.t. $\boldsymbol{U}$. In practice, instead of iterating through all $t \in \mathcal{T}$, for each $k$, we randomly choose $t$. The steps are shown in Algorithm 1.

In the second approach, we consider the continuous relaxation of $\boldsymbol{\lambda}$ and approximately solve it by uniformly randomly sampling $\boldsymbol{\lambda} \in [0, 1]^{|\mathcal{C}|}$,

$$\text{CMLU}_\beta\text{: } \max_{\boldsymbol{U}} \sum_{k\in\mathcal{C}} \sum_{t\in\mathcal{T}} \sum_{\substack{\boldsymbol{\lambda}\in[0,1]^{|\mathcal{C}|} \\ \|\boldsymbol{\lambda}\|_0=t, \boldsymbol{\lambda}_k=0}} \Big[ \mathcal{H}_k(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \tag{10}$$
$$- \eta \sum_{q\in\{\mathcal{C}\setminus k\}} \delta(\boldsymbol{\lambda}_q = 0) \mathcal{R}_q(\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}) \Big] \text{ s.t. } \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{U}} = \boldsymbol{I}, \|\boldsymbol{U}\|_{1,p} \leq \epsilon.$$

Eq. (9) and (10) can be explained intuitively. The first loss term $\mathcal{H}_k$ optimizes each label-wise universal perturbation $\boldsymbol{u}_k$ such that it is robust to the noise generated by composing other universal perturbations. Specifically, it aims to maximize the loss of class $k$, regardless of which other class universals have been combined. The second term ensures that $\boldsymbol{u}_k$ does not change the prediction of classes whose universal vectors have not been used in the composition.

**Remark:** *A key aspect of Eq. (9) and (10) is that $\boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}$ represents points sampled from an affine convex cone. Specifically, these formulations aim to maximize a function over points sampled from $|C|$ affine convex cones (generated by the columns of $\boldsymbol{U}$). Consequently, we can interpret our assumption in (4) as aiming to learn the universal adversarial regions of each class as affine cones, where the vector $\boldsymbol{u}_\Omega$, as outlined in (4), lies at the intersection of the affine cones of the classes in $\Omega$. This is illustrated in Fig. 2 (right), where $\boldsymbol{C}_k$ and $\boldsymbol{C}_i$ represent convex cones for classes $k$ and $i$ respectively, and $\boldsymbol{C}_{ik}$ denotes their intersection.*

For a comparison among Oracle, CMLU$_\alpha$, and CMLU$_\beta$, Algorithm 1 shows the steps of a single training epoch of Oracle of target size $|\Omega| = 1$ and the corresponding steps for our methods. Note that in general, Oracle has $\mathcal{O}(\binom{|\mathcal{C}|}{|\Omega|})$ training time complexity, which increases with the size of $\Omega$. For instance, for $|\Omega| = 2$, the complexity of the Oracle's inner loop becomes $\mathcal{O}(\binom{|\mathcal{C}|}{2})$, as it needs to handle all possible combinations of classes of a fixed size $|\Omega|$. However, CMLU$_\alpha$ has $\mathcal{O}(N_\alpha|\mathcal{C}|)$ and CMLU$_\beta$ has $\mathcal{O}(|\mathcal{C}|)$ complexity and is independent of $|\Omega|$. Note that the complexity of our method for learning composable universal perturbations for all possible target set sizes is equivalent to the complexity of Oracle trained for $|\Omega| = 1$ (linear in the number of classes). Oracle requires iterating over the dataset for each combination of target classes $\Omega$, leading to exponential complexity.

**Algorithm 1:** Algorithms of Oracle, CMLU$_\alpha$, CMLU$_\beta$

**Input :** $\mathcal{D}$: Training Data, $\mathcal{F}$: Classifier, $\epsilon$: Norm budget, $(\xi, \alpha)$: Step sizes
**Output:** $\boldsymbol{U}$: Learnt Universal Perturbations

Randomly initialize $\boldsymbol{U}$
/* $\mathcal{B} = \{\mathcal{B}^{(k)}\}_{k=1}^{|\mathcal{C}|}$, $\mathcal{B}^{(k)}$: images of class $k$ */
**Oracle:**
  **for** *Batch* $\mathcal{B}$ *in* $\mathcal{D}$ **do**
    /* One iteration through the dataset */
    **for** $k$ *in* $\mathcal{C}$ **do** /* $\mathcal{O}(|\mathcal{C}|)$ */
      $\mathcal{L}_k$: Compute loss in (1) using $\boldsymbol{u}_k$
      Compute the gradient: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{U}_k} l_k$
      Update $\boldsymbol{U}_k \leftarrow \Pi_\epsilon\left(\boldsymbol{U}_k + \xi * Sgn(\boldsymbol{g})\right)$
**for** *Batch* $\mathcal{B}$ *in* $\mathcal{D}$ **do**
  Set $\boldsymbol{g}$ to be all zeros of the same size as $\boldsymbol{U}$
  **for** $k$ *in* $\mathcal{C}$ **do** /* $\mathcal{O}(|\mathcal{C}|)$ */
    Randomly sample $t \sim \mathcal{T}$ and $\boldsymbol{\lambda} \in [0,1]^{|\mathcal{C}|}$;
    **CMLU$_\alpha$:** /* $\mathcal{O}(N_\alpha)$ */
      **for** $N_\alpha$ *num of iterations* **do**
        Set $\boldsymbol{v}_k = \boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}$
        $\mathcal{L}_k$: Compute loss in (9) using $\boldsymbol{v}_k$
        Update: $\boldsymbol{\lambda} = \boldsymbol{\lambda} - \alpha\, \nabla_{\boldsymbol{\lambda}} \mathcal{L}_k$
      Project $\boldsymbol{\lambda}$ on the feasible set $\|\boldsymbol{\lambda}\|_0 = t$
      Set $\boldsymbol{\lambda}_k = 0$ and $\boldsymbol{v}_k = \boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}$
      $\mathcal{L}_k$: Compute loss in (9) using $\boldsymbol{v}_k$
    **CMLU$_\beta$:** /* $\mathcal{O}(1)$ */
      Randomly set $|\mathcal{C}| - t$ values in $\boldsymbol{\lambda}$ to 0.
      Set $\boldsymbol{\lambda}_k = 0$ and $\boldsymbol{v}_k = \boldsymbol{u}_k + \boldsymbol{U}\boldsymbol{\lambda}$
      $\mathcal{L}_k$: Compute loss in (10) using $\boldsymbol{v}_k$
    Accumulate: $\boldsymbol{g} \leftarrow \boldsymbol{g} + \nabla_{\boldsymbol{U}_k} \mathcal{L}_k$
  Update $\boldsymbol{U} \leftarrow \Pi_\epsilon\left(\boldsymbol{U} + \xi * Sgn(\boldsymbol{g})\right)$

## 4. Evaluation

### 4.1. Experimental Setup

**Datasets** We perform experiments using NUS-WIDE [16], MS-COCO [56], and OpenImages [44] datasets to assess the effectiveness of our proposed method. We compare it with oracle and baseline methods. For NUS-WIDE and MS-COCO, we test target set sizes ranging from 1 to 5 and for OpenImages, we experiment with larger target set sizes, going up to 20. Due to the computational expense of learning oracle model, we choose a subset of labels from each dataset (25 classes from NUS-WIDE, 40 classes from MS-COCO, and 100 classes from OpenImages). Even with the small subset of classes, there can be a large number of possible combinations e.g., there can be more than 53k combinations of $|\Omega| = 5$ using 25 classes from NUS-WIDE. Therefore, we perform experiments on a subset of those possible combinations for each target set size $|\Omega|$. For all of our experiments, we learn universal vectors with maximum infinity norm $\epsilon = 0.05$ (for image values range of $[0, 1]$) and using SGD with fix step size $\xi = \epsilon/25$. We set $\gamma = \kappa/|\bar{\Omega}|$ for training oracle and baselines, and $\eta = \kappa/\|\boldsymbol{\lambda}\|_0$ in each iteration of our proposed method. We show ablation experiments for different values of $\kappa$ in section 4.2.1.

**Multi-Label Recognition Models** We evaluate the baselines and our method on state-of-the-art multi-label recognition models (using the officially provided weights).

– **Asymmetric Loss (ASL) [79]** is a novel loss function to address the imbalance between numerous negative labels and only a few positive labels associated with each sample. This method balances the probabilities of different positive and negative samples. In particular, the loss performs hard thresholding by dynamically down-weighting easy negative samples and can discard mislabeled samples.

– **ML-Decoder [80]** is an efficient, attention-based multi-label classifier that has a novel decoder architecture and

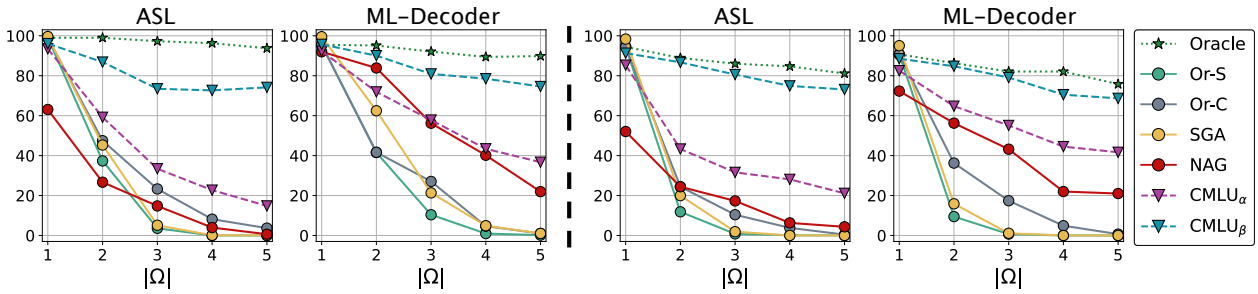


Figure 3. Fooling success rates ($FR$) on various datasets using ASL and ML-Decoder as we increase the target set size $|\Omega|$. The left two figures show results on NUS-WIDE and the right two figures show results on MS-COCO.

group-decoding scheme, which enables it to efficiently utilize the spatial information in images. The model is highly efficient as it can scale to large number of classes. Moreover, using word queries, it can generalize to unseen classes.

**Oracle and Baselines** We evaluate our method against three baselines and an oracle. The computational complexity of each of these methods is reported in Table 1 and their wall clock time in supplementary.

– **Oracle** directly optimizes (1) to learn a universal perturbation $\boldsymbol{u}_\Omega$ to attack a specific target set $\Omega$. This provides the upper bound on the attack performance for our method. It requires $2^{|C|}$ $d$-dimensional universal vectors and for each vector, iterate through the training set for $E$ epochs.

– **Oracle Sum (Or-S)** directly learns single-class universal perturbation vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots \boldsymbol{u}_{|C|}\}$ using (1) and generates a universal perturbation to attack set $\Omega$ as $\boldsymbol{u}_\Omega = \sum_{i\in\Omega} \boldsymbol{u}_i$. It only needs to learn $|C|$ $d$-dimensional vectors.

– **Oracle Combination (Or-C)** first learns single-class universal perturbation vectors and learns to combine them to attack a specific target set $\Omega$ i.e., $\boldsymbol{u}_\Omega = \sum_{i\in\Omega} a_i^\Omega \boldsymbol{u}_i$. Where, $a_i^\Omega \in \mathbb{R}$ is associated with $i^{th}$ class in specific set $\Omega$, and is learnt by iterating through the training set. It needs to learn $|C|$ $d$-dim vectors and $2^{|C|}$ scalars to combine them.

– **SGA [59]** addresses the gradient vanishing and poor local optima problem to improve the generalization of UAPs. It enhances the gradient stability by aggregating gradients from inner pre-search. We use SGA to learn single-class UAPs and use (4) to generate UAPs for target labels.

– **NAG [68]** models the distribution of universal adversarial perturbations using GANs, originally designed for untargeted perturbations in multi-class setting. To adapt this for multi-label universal perturbations, we condition the GAN on class vectors. During training, we generate label-wise universal perturbations. At test time, we generate perturbations for multiple classes by conditioning the GAN on a combined representation vector of the target classes. For this method, we have linear time complexity as we train to generate label-wise perturbations and the space requirement is the size of the generator.

**Evaluation Metrics** Given a set of classes $\Omega$ to attack, let $\mathcal{I}_\Omega$ represent all the images that contain the classes $\Omega$ and were attacked. Let $\mathcal{A}_\Omega \subseteq \mathcal{I}_\Omega$ represent the set of images that were successfully attacked. To evaluate the universal attack performance, we define the attack **fooling success rate**, $FR$, to measure the percentage of successfully attacked images. Since we want to affect the least number of non-targeted labels while achieving high fooling rates, we also report **non-target flip rate**, $NT_R$, to measure the percentage of non-targeted labels (labels in $\bar{\Omega}$) which were flipped by the attack. Our goal is to achieve high $FR$ while maintaining low $NT_R$ for attacking target set of any size.

$$\text{FR} = \frac{|\mathcal{A}_\Omega|}{|\mathcal{I}_\Omega|}, \text{NT}_R = \frac{1}{|\mathcal{A}_\Omega|} \sum_{k\in\mathcal{A}_\Omega} \frac{\sum_{i\in\bar{\Omega}} (1 - \delta(\chi(f_i^{(k)}), y_i^{(k)}))}{|\bar{\Omega}|} \tag{11}$$

where, $\delta$ is kronecker delta that equals 1 when the two inputs are equal and 0 otherwise. $\chi(v)$ is the step function which is 1 if $v > 0$ and 0 otherwise. $y_i^{(k)}, f_i^{(k)}$ are the predictions on clean and perturbed images respectively of $i^{th}$ non-target class of $k^{th}$ successfully attacked image. Note that $FR$ is calculated using *all* attacked images while $NT_R$ is calculated using the set of *successfully* attacked images.

## 4.2. Experimental Results

In this section, we evaluate our compositional framework for universal perturbations. Fig. 3 shows the fooling success rates for different methods. The results demonstrate consistent performance trends across datasets and models. Particularly, the success rate(FR) of baselines decline as we increase the target size. On the other hand, the Oracle method achieves the best performance but suffers from exponential complexity, making it impractical for larger target sets.

Among the baselines, Or-C, Or-S, and SGA perform the worst for large target set sizes. Since the performance of these methods depends on the label-wise universal perturbations, we can conclude that the low success rate of these methods implies that the the subspace spanned by label-wise perturbations does not contain the intersection of their respective adversarial regions. This is further elaborated in Fig. 5. In all cases, Or-C is better than Or-S since it op-

Table 1. Non-target flip rate ($NT_R$) on NUS-WIDE and MS-COCO, as we increase the target set size from 1 to 5. We show (-) when $|\mathcal{A}_\Omega| = 0$. We also show the space and time complexity of each method. $E$ represents the number of epochs for training, $d$ is the size of the perturbation (equal to the image size), $G_p$ is the size of GAN's parameters, and $|\mathcal{C}|$ is the number of classes.

| | | | ASL | | | | | | ML-Decoder | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Computational Complexity | | NUS-WIDE | | | MS-COCO | | | NUS-WIDE | | | MS-COCO | | |
| **Method** | **Space** | **Time** | **1** | **3** | **5** | **1** | **3** | **5** | **1** | **3** | **5** | **1** | **3** | **5** |
| **Oracle** | $\mathcal{O}(d\,2^{\lvert C\rvert})$ | $\mathcal{O}(E\,2^{\lvert C\rvert})$ | 6.17 | 7.42 | 8.10 | 6.50 | 7.73 | 8.66 | 2.86 | 3.10 | 3.01 | 7.38 | 8.58 | 9.81 |
| **Or-S** | $\mathcal{O}(d\,\lvert C\rvert)$ | $\mathcal{O}(E\,\lvert C\rvert)$ | 6.17 | 6.03 | - | 6.50 | 2.87 | 2.96 | 2.86 | 2.22 | - | 7.38 | 2.96 | - |
| **Or-C** | $\mathcal{O}(d\lvert C\rvert+2^{\lvert C\rvert})$ | $\mathcal{O}(E\,2^{\lvert C\rvert})$ | 6.17 | 6.86 | 2.74 | 6.50 | 5.23 | 3.20 | 2.86 | 2.45 | 0.60 | 7.38 | 6.40 | - |
| **SGA** | $\mathcal{O}(d\,\lvert C\rvert)$ | $\mathcal{O}(E\,\lvert C\rvert)$ | 7.69 | 7.31 | - | 7.77 | 4.30 | - | 2.89 | 2.47 | 1.27 | 8.62 | 8.94 | 9.37 |
| **NAG** | $\mathcal{O}(G_p)$ | $\mathcal{O}(E\,\lvert C\rvert)$ | 8.32 | 5.86 | 1.32 | 10.1 | 10.2 | 7.21 | 3.96 | 3.45 | 2.55 | 11.8 | 12.1 | 13.7 |
| **CMLU$_\alpha$** | $\mathcal{O}(d\,\lvert C\rvert)$ | $\mathcal{O}(EN_\alpha\lvert C\rvert)$ | 5.63 | 7.65 | 3.74 | 7.33 | 5.96 | 3.13 | 3.6 | 3.08 | 2.51 | 8.96 | 8.55 | 8.02 |
| **CMLU$_\beta$** | $\mathcal{O}(d\,\lvert C\rvert)$ | $\mathcal{O}(E\,\lvert C\rvert)$ | 7.96 | 8.13 | 7.93 | 8.90 | 8.86 | 9.59 | 4.23 | 3.74 | 3.02 | 9.65 | 9.83 | 10.5 |

Table 2. Experiments on large-scale dataset. We report fooling success rate (FR) and non-target flip rate (NT$_R$) on OpenImages using ASL model. The experiments were performed on large target set sizes, up to $|\Omega| = 20$. Higher FR (↑) and lower NT$_R$ (↓) are better.

| | FR (↑) | | | | | | | | NT$_R$ (↓) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **1** | **2** | **3** | **4** | **5** | **10** | **15** | **20** | **1** | **2** | **3** | **4** | **5** | **10** | **15** | **20** |
| **Oracle** | 97.6 | 97.9 | 96.8 | 97.9 | 95.7 | 96.3 | 96.8 | 96.8 | 0.64 | 0.80 | 0.87 | 0.93 | 0.96 | 1.05 | 1.17 | 1.47 |
| **Or-S** | 97.6 | 29.64 | 6.28 | 1.70 | 0.42 | 0.0 | 0.0 | 0.0 | 0.64 | 0.71 | 0.65 | 0.34 | 0.20 | - | - | - |
| **Or-C** | 97.6 | 49.4 | 22.8 | 23.3 | 13.4 | 0.0 | 0.0 | 0.0 | 0.64 | 0.71 | 0.67 | 0.75 | 0.48 | - | - | - |
| **SGA** | 98.9 | 41.4 | 9.64 | 6.11 | 0.92 | 0.15 | 0.0 | 0.0 | 0.77 | 0.81 | 0.73 | 0.74 | 0.32 | 0.12 | - | - |
| **NAG** | 95.9 | 92.1 | 88.8 | 88.4 | 89.7 | 79.9 | 80.9 | 84.2 | 0.81 | 0.89 | 0.93 | 1.01 | 1.07 | 1.42 | 1.53 | 1.84 |
| **CMLU$_\alpha$** | 92.2 | 66.7 | 44.0 | 46.2 | 39.4 | 39.9 | 32.4 | 32.3 | 0.72 | 0.76 | 0.83 | 0.89 | 0.83 | 0.86 | 0.82 | 0.77 |
| **CMLU$_\beta$** | 97.3 | 95.8 | 95.9 | 96.8 | 95.2 | 94.8 | 96.5 | 95.2 | 0.72 | 0.80 | 0.88 | 0.92 | 0.98 | 1.06 | 1.28 | 1.67 |

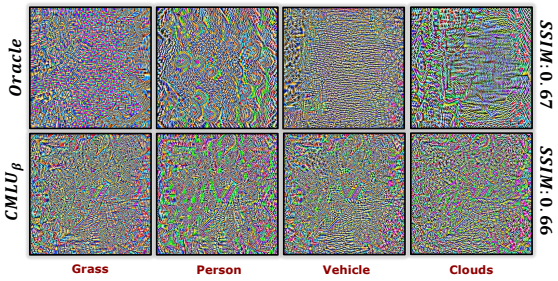


Figure 4. Visualizing label-wise universal perturbations learnt by Oracle and CMLU$_\beta$ on NUS-WIDE using ASL.

timizes to find the best combination of the label-wise universal perturbations and hence, is better than the naive summation of individual perturbations. Across all experiments, CMLU$_\beta$ is the closest to Oracle performance. This is despite the fact that CMLU$_\beta$ has linear complexity whereas the Oracle has exponential complexity. Moreover, CMLU$_\alpha$ achieves higher $FR$ and lower $NT_R$ than all baselines.

Table 1 shows the percentage of non-targeted affected labels (whose prediction was changed after the attack). Although the baselines have low NT$_R$ for target sizes greater than 1, they also have low fooling rates. Note that despite having linear complexity, the NT$_R$ of CMLU$_\beta$ is only slightly higher than Oracle. CMLU$_\alpha$ achieves lower $NT_R$ as compared to CMLU$_\beta$ but still outperforms the baselines. In all experiments, NAG performs well (close to CMLU$_\alpha$) but declines in performance as we increase the target size. It also has significantly higher NT$_R$ than the Oracle and CMLU$_\beta$. This implies that NAG learnt to attack multiple classes together rather than learning the distribution of label-wise universal perturbations. Therefore, even though it has high success rate, it also affects several other labels.

We further perform experiments on OpenImages, a large-scale dataset, and we show results in Tab. 2. Performance trends remain consistent with those observed on smaller datasets. The success rate of Or-S and Or-C consistently drops as we target more classes, and gets to 0 by target size 10. Note that NAG performs better in OpenImages experiments. As shown in the table, NAG affects a large percentage of non-targeted labels, which signifies that the method does not capture the class-specific directions and perturbs a large group of labels together. CMLU$_\alpha$ performs better than Or-C, Or-S, and SGA but worse than NAG. However, as shown in the table, CMLU$_\alpha$ still achieves significantly lower $NT_R$ for large target set sizes. In all experiments, CMLU$_\beta$ achieves consistently close performance to the Oracle while also maintaining low NT$_R$.

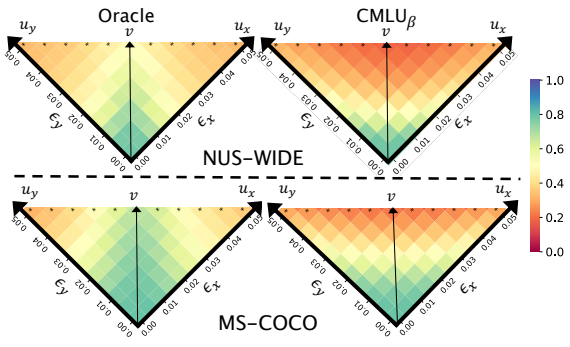


Figure 5. Comparison of the regions (**convex cones**) spanned by universal perturbations learnt using Oracle and CMLU$_\beta$ on NUS-WIDE (**top**) and MS-COCO (**bottom**). Each point in the heatmap is the average model confidence computed across images using the vector $(\epsilon_x * \boldsymbol{u}_x + \epsilon_y * \boldsymbol{u}_y)$ for all label combinations of size $|\Omega| = 2$, where $\boldsymbol{u}_x, \boldsymbol{u}_y$ are universal vectors of classes in $\Omega$. The asterisks$(*)$ show the region where the $L_\infty$ norm is $0.05$.

#### 4.2.1. Ablation Study

We perform ablation experiments on Oracle and CMLU$_\beta$ to investigate the effect of hyperparameters on their performance. Tab. 3 shows the results as we increase/decrease the values of $\epsilon$ and $\kappa$. Increasing $\epsilon$ provides a larger perturbation budget for better performance; however, to keep the changes to the inputs imperceptible, we set $\epsilon = 0.05$. Meanwhile, increasing $\kappa$ emphasizes maintaining non-targeted labels, which lowers NT$_R$ but also reduces the success rate (FR). This is because many labels can be correlated in MLL and attacking a set of labels while fixing non-targeted but correlated labels lead to sub-optimal perturbations [63].

#### 4.2.2. Analysis of Universal Adversarial Regions

The label-wise universal perturbations learnt by Oracle and CMLU$_\beta$ are visualized in Fig. 4. We quantitatively assess the visual impact of these perturbations using Structural Similarity Index (SSIM), computed between original and perturbed images. The average SSIM values across all images, as shown in Fig. 4, demonstrate comparable levels of distortion for Oracle and CMLU$_\beta$. Note that the pairwise dot product of the Oracle and CMLU vectors is $\approx 0$, indicating that CMLU$_\beta$ discover distinct attack directions, different from the Oracle. Additional analysis of pairwise dot products and visualizations are provided in supplementary.

To further highlight and analyze the distinction between the perturbations learnt using different methods, we focus on Oracle and CMLU$_\beta$. Specifically, we examine the universal adversarial regions generated by those methods. Fig. 5 shows the average model confidence on the input space region spanned by the pairs of universal perturbation vectors. In the figure, perturbing inputs to blue and green regions would not change the model prediction (does not attack any class), yellow region would attack one class, and red region would attack both classes in $\Omega$. Note that there is no prominant red region (to attack both classes) in the cross-section of Oracle vectors. This explains the low success rate of Or-S and Or-C which optimize to find red region in such a cross-section. More details on computing these cones is provided in supplementary. We also investigate how well those red regions computed using CMLU$_\beta$ can transfer to other models. For that, we perform transferability attacks and the results are shown in Tab. 4. From the table, we observe that CMLU$_\beta$ (linear complexity) achieves fooling success rates closer and often better than Oracle (exponential complexity). This shows that the convex cone regions corresponding to different labels are often common across models. This can be effectively used to generate black-box attacks. We hope that future works explore this further.

Table 3. Ablation experiments for hyperparameters on NUS-WIDE. The experiments are conducted for $|\Omega| = 1$.

| $\epsilon$ | Oracle | | CMLU$_\beta$ | | $\kappa$ | Oracle | | CMLU$_\beta$ | |
|---|---|---|---|---|---|---|---|---|---|
| | FR | NT$_R$ | FR | NT$_R$ | | FR | NT$_R$ | FR | NT$_R$ |
| **0.01** | 63.5 | 5.30 | 38.5 | 5.04 | **15** | 99.1 | 6.50 | 98.5 | 8.32 |
| **0.03** | 94.9 | 6.29 | 84.2 | 7.09 | **20** | 98.9 | 6.17 | 97.6 | 7.96 |
| **0.05** | 98.9 | 6.17 | 97.6 | 7.96 | **25** | 93.5 | 4.70 | 89.3 | 6.97 |
| **0.07** | 99.7 | 5.10 | 99.3 | 7.54 | **30** | 93.1 | 4.40 | 85.5 | 6.58 |

Table 4. Transferability Evaluation: $S \rightarrow T$ shows the performance of universal perturbations learnt on model **S** and evaluated on model **T**. (**A**: ASL, **D**: ML-Decoder)

| **Transferability of Universal Attacks** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | NUS-WIDE | | | MS-COCO | | |
| **Models** | **Method** | **1** | **3** | **5** | **1** | **3** | **5** |
| **A** $\rightarrow$ **D** | **Oracle** | **90.1** | **83.7** | **77.3** | 34.6 | 10.6 | 5.40 |
| | **CMLU$_\beta$** | 89.9 | 78.4 | 75.4 | **42.9** | **17.6** | **6.4** |
| **D** $\rightarrow$ **A** | **Oracle** | 24.6 | **1.33** | 0.0 | 64.4 | 28.3 | 19.6 |
| | **CMLU$_\beta$** | **33.5** | 0.16 | 0.0 | **64.8** | **34.5** | **21.2** |

## 5. Conclusion

We addressed the problem of generating targeted multi-label universal perturbations. We developed a compositional framework to generate exponential number of multi-label universal perturbations in constant time. The independence assumption on label-wise universal perturbations naturally leads to a formulation that requires learning convex cones. Through extensive experiments on multiple datasets and models, we show that our method outperforms all baseline methods and achieves performance comparable to the Oracle (an exponential complexity method).

## 6. Acknowledgements

This work is sponsored by ARPA-H (1AY2AX000062), DARPA PTG (HR00112220001), NSF (IIS-2115110) and ARO (W911NF2110276). Content does not necessarily reflect the position/policy of the Government.

## References


[1] Abhishek Aich, Calvin-Khang Ta, Akash Gupta, Chengyu Song, Srikanth Krishnamurthy, Salman Asif, and Amit Roy-Chowdhury. Gama: Generative adversarial multi-object scene attacks. *Advances in Neural Information Processing Systems*, 35:36914–36930, 2022. 1, 3

[2] Abhishek Aich, Shasha Li, Chengyu Song, M Salman Asif, Srikanth V Krishnamurthy, and Amit K Roy-Chowdhury. Leveraging local patch differences in multi-object scenes for generative adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1308–1318, 2023. 1

[3] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv*, 2018. 2

[4] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. 2018. 2

[5] Yuanhao Ban and Yinpeng Dong. Pre-trained adversarial perturbations. In *Advances in Neural Information Processing Systems*, pages 1196–1209, 2022. 2

[6] Emanuel Ben-Baruch, Tal Ridnik, Itamar Friedman, Avi Ben-Cohen, Nadav Zamir, Asaf Noy, and Lihi Zelnik-Manor. Multi-label classification with partial annotations using class-aware selective loss. 2022. 2

[7] P. Benz, C. Zhang, T. Imtiaz, and I.-S. Kweon. Double targeted universal adversarial perturbations. *Asian Conference on Computer Vision*, 2020. 1, 2

[8] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Universal adversarial training with class-wise perturbations. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 2

[9] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *Workshop on Artificial Intelligence and Security*, 2017. 1

[10] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*, 2017. 2

[11] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. Unlabeled data improves adversarial robustness. *Neural Information Processing Systems*, 2019. 2

[12] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *AAAI Conference on Artificial Intelligence*, 2018. 2

[13] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin. Learning semantic-specific graph representation for multi-label image recognition. *IEEE International Conference on Computer Vision*, 2019. 2

[14] Zhao-Min Chen, Xiu-Shen Wei, Xin Jin, and Yanwen Guo. Multi-label image recognition with joint class-aware map disentangling and label correlation embedding. *IEEE International Conference on Multimedia and Expo*, 2019. 1

[15] Z. M. Chen, X. S. Wei, P. Wang, and Y. Guo. Multi-label image recognition with graph convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, abs/1904.03582, 2019. 1

[16] T. S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. T. Zheng. Nus-wide: A real-world web image database from national university of singapore. *ACM International Conference on Image and Video Retrieval*, 2009. 5

[17] S. D. Dao, E. Zhao, D. Phung, and J. Cai. Multi-label image classification with contrastive learning. *arXiv preprint, arXiv:2107.11626*, 2021. 2

[18] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. *European Conference on Computer Vision*, 2014. 2

[19] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018. 2

[20] Zixuan Ding, Ao Wang, Hui Chen, Qiang Zhang, Pengzhang Liu, Yongjun Bao, Weipeng Yan, and Jungong Han. Exploring structured semantic prior for multi label recognition with incomplete labels. 2023. 1

[21] Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24678–24687, 2023. 2

[22] Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020. 2

[23] L. Feng, B. An, and S. He. Collaboration based multi-label learning. *AAAI Conference on Artificial Intelligence*, 2019. 1

[24] Zongyuan Ge, Dwarikanath Mahapatra, Suman Sedai, Rahil Garnavi, and Rajib Chakravorty. Chest x-rays classification: A multi-label and fine-grained problem. *arXiv*, 2018. 1

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 1

[26] Emily Hand, Carlos Castillo, and Rama Chellappa. Doing the best we can with what we have: Multi-label balancing with selective learning for attribute prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1

[27] J. Hayes and G. Danezis. Learning universal adversarial perturbations with generative models. *IEEE Security and Privacy Workshops*, 2018. 2

[28] W. He, J. Wei, X. Chen, N. Carlini, and D. Song. Adversarial example defense: Ensembles of weak defenses are not strong. *USENIX Workshop on Offensive Technologies*, 2017. 2

[29] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. 2019. 2

[30] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 1

[31] Lei Hsiung, Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24658–24667, 2023. 1

[32] Shu Hu, Lipeng Ke, Xin Wang, and Siwei Lyu. Tkml-ap: Adversarial attacks to top-k multi-label learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7649–7657, 2021. 3

[33] D. Huynh and E. Elhamifar. Interaction compass: Multi-label zero-shot learning of human-object interactions via spatial relations. *International Conference on Computer Vision*, 2021. 1

[34] D. T. Huynh and E. Elhamifar. Interactive multi-label cnn learning with partial labels. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[35] Tooba Imtiaz, Morgan Kohler, Jared Miller, Zifeng Wang, Mario Sznaier, Octavia I Camps, and Jennifer G Dy. Saif: Sparse adversarial and interpretable attack framework. *arXiv preprint arXiv:2212.07495*, 2022. 2

[36] J. Li R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian. Universal perturbation attack against image retrieval. *International Conference on Computer Vision*, 2019. 1

[37] Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. Multiguard: Provably robust multi-label classification against adversarial examples. *Advances in Neural Information Processing Systems*, 2022. 3

[38] V. Khrulkov and I. Oseledets. Art of singular vectors and universal adversarial perturbations. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[39] Youngwook Kim, Jae Myung Kim, Zeynep Akata, and Jungwoo Lee. Large loss matters in weakly supervised multi-label classification. 2022. 1

[40] Takumi Kobayashi. Two-way multi-label loss. 2023. 1

[41] Linghao Kong, Wenjian Luo, Hongwei Zhang, Yang Liu, and Yuhui Shi. Evolutionary multilabel adversarial examples: An effective black-box attack. 2023. 3

[42] Linghao Kong, Wenjian Luo, Zipeng Ye, Qi Zhou, and Yan Jia. Multi-label black-box adversarial attacks only with predicted labels. 2024. 1

[43] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017. 1

[44] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 2016. 5

[45] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi. General multi-label image classification with transformers. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[46] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. 2018. 2

[47] Guofa Li, Zefeng Ji, Yunlong Chang, Shen Li, Xingda Qu, and Dongpu Cao. Ml-anet: A transfer learning approach using adaptation network for multi-label image classification in autonomous driving. *Chinese Journal of Mechanical Engineering*, 2021. 1

[48] Maosen Li, Yanhua Yang, Kun Wei, Xu Yang, and Heng Huang. Learning universal adversarial perturbation by adversarial example. *AAAI Conference on Artificial Intelligence*, 2022. 3

[49] Peng Li, Peng Chen, Yonghong Xie, and Dezheng Zhang. Bi-modal learning with channel-wise attention for multi-label image classification. *IEEE Access*, 2020. 2

[50] Q. Li, M. Qiao, W. Bian, and D. Tao. Conditional graphical lasso for multi-label image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[51] Q. Li, X. Peng, Y. Qiao, and Q. Peng. Learning label correlations for multi-label image recognition with graph networks. *Pattern Recognition Letters*, 2020. 2

[52] X. Li, F. Zhao, and Y. Guo. Multi-label image classification with a probabilistic label enhancement model. In *UAI*, 2014. 2

[53] Y. Li, Y. Song, and J. Luo. Improving pairwise ranking for multi-label image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[54] Z. Li, W. Lu, Z. Sun, and W. Xing. Improving multi-label classification using scene cues. *Multimedia Tools and Applications*, 2017. 2

[55] Dekun Lin. Probability guided loss for long-tailed multi-label image classification. 2023. 1

[56] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *European Conference on Computer Vision*, 2014. 5

[57] An-An Liu, Zhuang Shao, Yongkang Wong, Junnan Li, Su Yu-Ting, and Mohan Kankanhalli. Lstm-based multi-label video event detection. *Multimedia Tools and Applications*, 78, 2019. 1

[58] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*, 2021. 2

[59] Xuannan Liu, Yaoyao Zhong, Yuhang Zhang, Lixiong Qin, and Weihong Deng. Enhancing generalization of universal adversarial perturbation through gradient aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 6

[60] Yiran Liu, Xin Feng, Yunlong Wang, Wu Yang, and Di Ming. Trm-uap: Enhancing the transferability of data-free universal adversarial perturbation via truncated ratio maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4762–4771, 2023. 2

[61] Mingzhi Ma, Weijie Zheng, Wanli Lv, Lu Ren, Hang Su, and Zhaoxia Yin. Multi-label adversarial attack based on label correlation. In *2023 IEEE International Conference on Image Processing (ICIP)*, 2023. 3

[62] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. 2

[63] Hassan Mahmood and Ehsan Elhamifar. Semantic-aware multi-label adversarial attacks. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 24251–24262, 2024. 1, 3, 8

[64] S. Melacci, G. Ciravegna, A. Sotgiu, A. Demontis, B. Biggio, M. Gori, and F. Roli. Domain knowledge alleviates adversarial attacks in multi-label classifiers. 2021. 3

[65] J.-H. Metzen, M.-C. Kumar, T. Brox, and V. Fischer. Universal adversarial perturbations against semantic image segmentation. *International Conference on Computer Vision*, 2019. 1, 2

[66] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *IEEEConference on Computer Vision and Pattern Recognition*, 2016. 2

[67] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[68] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*, 2017. 6

[69] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018. 1, 2

[70] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz. Maximizing subset accuracy with recurrent neural networks in multi-label classification. *Neural Information Processing Systems*, 2017. 2

[71] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu. Improving adversarial robustness via promoting ensemble diversity. *International Conference on Machine learning*, 2019. 2

[72] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv*, 2020.

[73] T. Pang, X. Yang, Y. Dong, K. Xu, H. Su, and J. Zhu. Boosting adversarial training with hypersphere embedding. *Neural Information Processing Systems*, 2020. 2

[74] Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016. 1

[75] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*, 2015. 1

[76] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 2

[77] Tao Pu, Tianshui Chen, Hefeng Wu, and Liang Lin. Semantic-aware representation blending for multi-label image recognition with partial labels. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1

[78] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. *Advances in Neural Information Processing Systems*, 35: 29845–29858, 2022. 2

[79] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 82–91, 2021. 5

[80] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 32–41, 2023. 5

[81] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 2

[82] J. Cohenand E. Rosenfeld and Z. Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine learning*, 2019. 2

[83] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! *Neural Information Processing Systems*, 2019.

[84] Nasim Shafiee and Ehsan Elhamifar. Zero-shot attribute attacks on fine-grained recognition models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 262–282. Springer, 2022. 1, 2

[85] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. Deeply learned attributes for crowded scene understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[86] Nitish Shukla and Sudipta Banerjee. Generating adversarial attacks in the latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 730–739, 2023. 1

[87] Q. Song, H. Jin, X. Huang, and X. Hu. Multi-label adversarial perturbations. *IEEE International Conference on Data Mining*, 2018. 3

[88] Fengguang Su, Ou Wu, and Weiyao Zhu. Multi-label adversarial attack with new measures and self-paced constraint weighting. 2024. 1

[89] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 2

[90] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *Intenational Journal Data Warehousing and Mining*, 3, 2007. 2

[91] N. Tursynbek, A. Petiushko, and I. Oseledets. Geometry-inspired top-k adversarial perturbations. *arXiv*, 2020. 2

[92] J. Uesato, J. B. Alayrac, P. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? *Neural Information Processing Systems*, 2019. 2

[93] Thomas Verelst, Paul K Rubenstein, Marcin Eichner, Tinne Tuytelaars, and Maxim Berman. Spatial consistency loss for training multi-label classifiers from single-label annotations. 2023. 2

[94] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[95] Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. Multi-label classification with label graph superimposing. In *Proceedings of the AAAI conference on artificial intelligence*, pages 12265–12272, 2020.

[96] Z. Wang, T. Chen, G. Li, G. Li, and L. Lin. Multi-label image recognition by recurrently discovering attentional regions. *IEEE International Conference on Computer Vision*, 2017. 2

[97] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 2

[98] Y. Wu, H. Liu, S. Feng, Y. Jin, G. Lyu, and Z. Wu. Gmmlic: Graph matching based multi-label image classification. *International Joint Conference on Artificial Intelligence*, 2021. 2

[99] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. Yuille. Improving transferability of adversarial examples with input diversity. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[100] Ming-Kun Xie, Jiahao Xiao, and Sheng-Jun Huang. Label-aware global consistency for multi-label learning with single positive labels. *Advances in Neural Information Processing Systems*, 35:18430–18441, 2022. 1

[101] J. Xu, H. Tian, Z. Wang, Y. Wang, W. Kang, and F. Chen. Joint input and output space learning for multi-label image classification. *IEEE Transactions on Multimedia*, 2020. 2

[102] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *Network and Distributed Systems Security Symposium*, 2018. 2

[103] Zhuo Yang, Yufei Han, and Xiangliang Zhang. Characterizing the evasion attackability of multi-label classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10647–10655, 2021. 3

[104] Zhuo Yang, Yufei Han, and Xiangliang Zhang. Attack transferability characterization for adversarially robust multi-label classification. In *Machine Learning and Knowledge Discovery in Databases.*, pages 397–413, 2021. 3

[105] V. O. Yazici, A. Gonzalez-Garcia, A. Ramisa, B. Twardowski, and J. v. d. Weijer. Orderless recurrent models for multi-label classification. *European Conference on Computer Vision*, 2020. 2

[106] J. Ye, J. He, X. Peng, W. Wu, and Y. Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. *European Conference on Computer Vision*, 2020.

[107] R. You, Z. Guo, L. Cui, X. Long, S. Y. Bao, and S. Wen. Cross-modality attention with semantic graph embedding for multi-label classification. *AAAI Conference on Artificial Intelligence*, 2020. 2

[108] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2148–2157, 2019. 1

[109] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Cd-uap: Class discriminative universal adversarial perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6754–6761, 2020. 2

[110] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[111] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. In *Proceedings of the AAAI conference on artificial intelligence*, 2021. 2

[112] ML. Zhang and Z. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 2006. 2

[113] Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. Use all the labels: A hierarchical multi-label contrastive learning framework. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[114] Zhe Zhao, Guangke Chen, Jingyi Wang, Yiwei Yang, Fu Song, and Jun Sun. Attack as defense: characterizing adversarial examples using robustness. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, page 42–55. Association for Computing Machinery, 2021. 2

[115] Donghao Zhou, Pengfei Chen, Qiong Wang, Guangyong Chen, and Pheng-Ann Heng. Acknowledging the unknown for multi-label learning with single positive labels. *European Conference on Computer Vision*, 2022. 1

[116] N. Zhou, W. Luo, X. Lin, P. Xu, and Z.. Zhang. Generating multi-label adversarial examples by linear programming. *International Joint Conference on Neural Networks*, 2020. 3

[117] N. Zhou, W. Luo, J. Zhang, L. Kong, and H. Zhang. Hiding all labels for multi-label images: An empirical study of adversarial examples. *International Joint Conference on Neural Networks*, 2021. 3

[118] Y. Zhu, J. T. Kwok, and Z. Zhou. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering*, 2018. 2