Multi-Label Black-box Attacks via Evolutionary Structured Many-objective Adversarial Perturbations

Anonymous Authors

***Abstract*—Multi-label learning poses significant challenges due to the complexities of co-occurring labels. Adversarial examples are critical in safety-sensitive domains, where malicious tampered data can compromise models. Yet, their application to tabular multi-label learning remains under-explored, presenting a potential security risk. This paper introduces an adversarial training framework leveraging Evolutionary Computation, specifically the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), to craft structured and concealable adversarial examples for tabular multi-label classifiers. Our contributions centre on an effective adversarial method tailored for tabulated multi-label learning, and a many-objective framework to balance the conflict between multi-label attack success, attack robustness, and attack concealability. We extend multi-label adversarial training to cope with tabulated data necessitating novel methods for generating structured adversarial examples and assessing attack concealability compared to image-based approaches—pioneering future research in tabulated multi-label adversarial training.Our framework also simulates real-world black-box attack scenarios where true model information is unknown. Our approach trains adversarial examples without prior knowledge of the target model by competing with a proxy model, progressively training more robust adversarial examples. Experiments show a high attack success rate (81.3–100%) across large datasets, significantly reducing multi-label classification performance post-perturbation and confirming the concealability of the attacks. Our results highlight the robustness of our approach, advancing adversarial training for multi-label, tabulated data.**

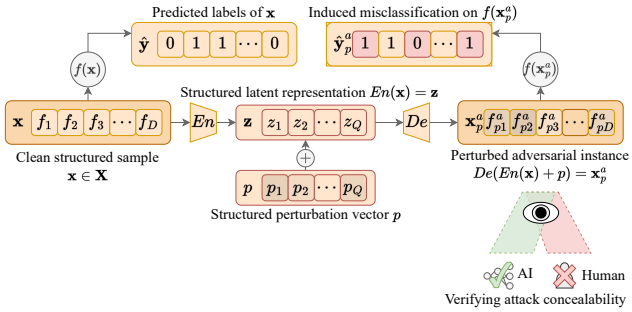
***Index Terms*—Adversarial perturbations, black-box attacks, evolutionary many-objective, multi-label, classification**

I. Introduction

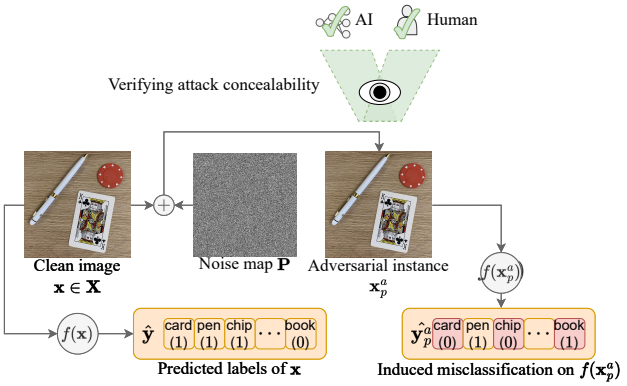
MULTI-LABEL classification has become a contemporary machine learning problem that involves predicting a *set* of class labels for instances, belonging to a variety of domains including text or semantic analysis [1]–[5] and computer vision [6]–[13]. State-of-the-art research in multi-label classification has primarily advanced through deep-learning-based methods in contemporary domains such as computer vision and tabulated (structured) problems. Before the realisation of adversarial training [14], [15], traditional deep-learning methods are trained on *clean* instances (referring to instances that have not been modified by a third party with malicious intent), without any conditioning to withstand a potential attack. In safety-critical applications, the robustness of a trained model against adversarial attacks has become a critical point of contention. To this date, adversarial training has mostly focused on computer vision (by focusing on adversarial images), leaving adversarial training on tabulated problems, to the best of our knowledge, mostly unexplored. This is particularly important as tabulated multi-label learning represents a broader spectrum of multi-label domains, each flattened to a tabulated representation via pre-trained models or pre-specified methods [3]. Adversarial methods, largely designed for images, do not translate well to the structured nature of tabulated multi-label learning. The structured, high-dimensional nature of tabulated multi-label data introduces unique challenges in crafting perturbations that balance effectiveness and concealability. The subtleties of perturbation magnitude are also critical in tabulated contexts, where changes must be meticulous to avoid detection by statistical or machine learning defenses. This presents a critical safety risk for much of multi-label deep learning, as malicious third parties can inject tampered tabulated data to induce misclassification, even without prior knowledge of the target model structure or its decision boundaries [14], [15]. This is especially critical in tabulated problems where misclassification can have profound consequences. In multi-label scenarios, such errors could result in inaccurate medical diagnoses or the improper assignment of semantic labels, potentially allowing hate speech or sensitive content to go unchecked, or mistakenly censoring normal content.

Adversarial attacks are achieved through adversarial training, which learns optimal perturbations of instances that aim to *fool*, *i.e.,* induce misclassification of a deep learning model by exploiting its decision boundaries [16]. There are primarily two types of attacks, white-box and black-box attacks [17]. White-box attacks are achieved by accessing model parameters, in contrast, black-box attacks only rely on model outputs and model inputs (without any access to the underlying model weights, gradients, or structure), thus simulating real-world attack scenarios where the attacker does not possess access to critical model information [17]. The key ideas behind tabulated multi-label and image-based multi-label adversarial training are respectively illustrated in Fig. 1 (a) and (b). Perturbations that ultimately mislead a multi-label classifier's output are challenging to find in multi-label problems. The complexity of multi-label classification necessitates that adversarial attacks inducing misclassification must account for the intricate interactions between features and multiple class labels. This complexity poses additional challenges compared to single-label classification, as perturbations may need to add, remove, or swap a set of relevant or irrelevant labels to mislead the classifier's output all while keeping the attack concealed. Therefore, multi-label adversarial attacks must consider the interactions and dependencies among multiple labels. Image-based adversarial attacks, shown in Fig. 1 (b)

Manuscript received X



(a) Tabular (structured) multi-label adversarial training via perturbation-based methods. Perturbations $\mathbf{p} \in \mathbb{R}^Q$ are applied to latent representations $z \in \mathbb{R}^Q$ of tabulated instances $\mathbf{x} \in \mathbb{R}^D$ to induce the misclassification of multiple class labels.



(b) Image-based multi-label adversarial training via noise map $\mathbf{P}$ to generate image $\mathbf{x}_p^a$ that induces misclassification over multiple class labels.

Fig. 1: Subfigures (a) and (b) respectively show the key differences between multi-label adversarial training on tabular (structured) data and contemporary (image) data. In either case, a structured perturbation vector $\mathbf{p}$, or noise map $\mathbf{P}$, is learned to perturb a clean instance $\mathbf{x}$ to induce misclassification over multiple class labels on a model f trained on a set of clean instances $\mathbf{X}^{N\times D}$ given N instances and D features.

can be practical, where concealability can be determined using human verification or machine learning. In contrast, tabulated multi-label problems lack the same practicality, and usually cannot be verified as concealable by the human eye due to the structured representation (especially with hundreds or thousands of features), therefore the need for statistical or machine learning methods to detect an attack makes them particularly dangerous.

A promising attach of multi-label data is to encode an instance using an autoencoder, introduce perturbations into the latent representation, and then decode the instance back into the original feature space. However, standard autoencoders tend to produce multiple entangled latent dimensions that may change in tandem when introducing perturbations, thus limiting control. Advances in variational autoencoders (VAE), *e.g.,* β-VAE [18], have enabled disentangled latent representation learning. This allows changes in one latent dimension to be relatively invariant to changes in other latent dimensions, allowing an adversarial attack by manipulating the latent representations of the input data. Generating *convincing* or *robust* adversarial examples during adversarial training also necessitates balancing the adversarial examples' attack success (quantifying its ability to mislead a classifier), and the classifier's robustness against the adversarial example. Adversarial training can be framed as a many-objective optimisation problem since attack success and robustness are directly conflicting objectives [16], [19]. Furthermore, it is desirable to conceal the attack attempt via invisible perturbations without sacrificing the attack's success, which can introduce additional conflicting objectives during adversarial training. Evolutionary computation (EC) has been widely used to optimise multiple conflicting objectives; thus, we aim to leverage EC, particularly the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), in this work.

The aforementioned challenges of multi-label adversarial training raise several interesting research questions. First, how can we design a robust adversarial training framework for tabulated multi-label problems, necessitating new methods to generate structured perturbations? Second, how can we learn convincing perturbations where the model information is not available, *i.e.,* a black-box attack? Third, how can we optimise the attack success, classifier robustness, and attack invisibility as adversarial training objectives simultaneously? These research questions motivate the design of *Multi-Label black-box attacks via Many-objective Adversarial Perturbations* (ML-MAP), which constitutes the following major contributions.

A. Contributions

Our work proposes the Multi-Label black-box attacks via Many-objective Adversarial Perturbations (ML-MAP) framework, which leverages CMA-ES [20] to generate robust adversarial examples without prior model information. ML-MAP utilizes CMA-ES to optimise structured perturbations, ensuring these adversarial instances remain effective and undetected. The primary contributions of this paper can be summarised as follows:

1) We develop a novel adversarial training framework, incorporating CMA-ES to generate structured perturbations for tabular multi-label data.
2) ML-MAP demonstrates an effective black-box attack method using CMA-ES and many-objective optimisation, producing adversarial instances that balance attack success, robustness, and concealability.
3) Through CMA-ES optimisation, ML-MAP efficiently exposes vulnerabilities in current state-of-the-art tabular

multi-label models, creating adversarial samples that remain statistically undetected.

4) ML-MAP's perturbations can exploit decision boundary vulnerabilities to enhance misclassification, providing potential insights into how adversarial strategies can indirectly influence label confidence without prior knowledge of the attacked model.

II. Background

A. Multi-label classification

Multi-label classification is emerging as the predominant classification paradigm prevalent in many domains such as computer vision [6]–[13] and tabulated learning [1]–[5]. Traditional multi-label classification algorithms transformed a multi-label problem into a series of single-label problems [21], [22], however, the efficacy of such methods was limited as valuable label interactions were lost after transformation. In response, deep-learning has been the primary influence in advancing the multi-label classification field by not only learning to predict all labels simultaneously but also learning the various interactions between them [1], [23], [24].

Many state-of-the-art deep-learning methods have since been proposed for tabulated multi-label classification. Dual perspective label-specific feature learning for multi-label classification (DELA) [4] was proposed to train networks that were robust against non-informative features through a perturbation-based feature training framework. DELA's framework identified noninformative features for each label and made the discrimination process invariant to feature changes. This was achieved by perturbing label-specific features during training while simultaneously identifying non-informative features, optimised by a relaxed expected risk minimisation problem. Another state-of-the-art method, collaborative learning of label semantics and deep label-specific features (CLIF) [3], embedded label interactions into the weights of the neural network by superimposing a label graph using a graph auto-encoder. This was achieved by jointly encoding a label relationship graph into semantic embeddings and encoding label-specific features using a disentanglement module. End-to-end probabilistic label-specific feature learning for multi-label classification (PACA) [25] introduced a unified framework to jointly perform clustering analysis of each positive/negative instance of each class label to obtain positive/negative prototypes; then, label-specific features are learnt by measuring distances between the original instances and the prototypes; and finally, classifiers are induced based on the label-specific features.

Generally speaking, DELA, CLIF, and PACA compared to many well-known multi-label learners: multi-label learning with label-specific features (LIFT) [26], learning label-specific features for multi-label classification (LLSF) [27], joint feature selection and classification for multi-label learning (JFSC) [28], learning deep latent spaces for multi-label classification (C2AE) [1], and disentangled variational autoencoder-based multi-label classification with covariance-aware multivariate probit model (MPVAE) [2], have achieved state-of-the-art results on a diverse set of tabulated multi-label domains. Therefore, deep-learning has been solidified as the predominant methodology for contemporary multi-label problems.

B. Adversarial training

Deep-learning methods are known to be susceptible to attack by adversarial examples, thereby enabling attackers to induce miss-classification. Many of the existing studies on adversarial training are focused on multi-class classification, which cannot directly be applied to the multi-label scenario due to the additional complexities of having multiple class labels per instance [17]. Traditional methods for generating adversarial examples have utilised gradient-based methods such as the fast gradient sign method (FGSM) [16] to rapidly generate adversarial examples. Following the FGSM study, many existing gradient-based methods for generating adversarial examples have since been proposed [29], [30]. Studies on adversarial training for multi-label classification have been overwhelmingly applied to multi-label image datasets. Song et. al. [31] proposed a multi-label variant of Carlini & Wagner [32] attack (ML-CW) and DeepFool [33] (ML-DF). ML-CW is an optimisation-based method that combines both an ℓ_2-norm of perturbations and hinge-loss of attack targets for images. ML-DF is defined as a constraint optimisation algorithm to superimpose a perturbation on an image until it induces misclassification. Furthermore, two ranking-based methods were also introduced in the same study, namely ML-Rank I and ML-Rank II.

In any case, the proposed methods were designed specifically for images and assume certain properties of the target classifier for an attack, *i.e.,* a white-box attack, thus rendering them inapplicable to black-box attacks on structured, tabulated data. To enable black-box attacks in multi-label data, Kong et. al. [17] proposed an evolutionary adversarial training method for multi-label images with differential evolution (MLAE-DE), utilising new fitness and crossover operators to generate perturbed adversarial images. Recently, Wang et. al. [19] proposed to learn structured perturbations for latent representations of single-label images. The perturbations are trained in a generative adversarial network-based (GAN) framework and compared against several single-label implementations of FGSM, CW, and DF. However, in both papers, the adversarial training method is again tailored for image data, moreover, the methodology is not designed to account for the conflict between model robustness, adversarial attack success, perturbation magnitude, and attack concealability.

The existing studies on adversarial training highlight the critical gap in research between tabulated multi-label learning and adversarial training for black-box attacks, despite the many state-of-the-art deep-learning-based multi-label learners [1]–[5]. This leaves an important area of multi-label learning unexplored and presents an important opportunity to propose a robust multi-label adversarial training algorithm to mimic potential real-world attacks for safety-critical systems via black-box scenarios. To the best of our knowledge, we propose the first multi-label tabulated adversarial training method for black-box attacks.

C. Many-objective optimisation

Generating convincing adversarial examples requires a balance between model classification performance and attack success, *i.e.,* an attacked model and adversarial example should

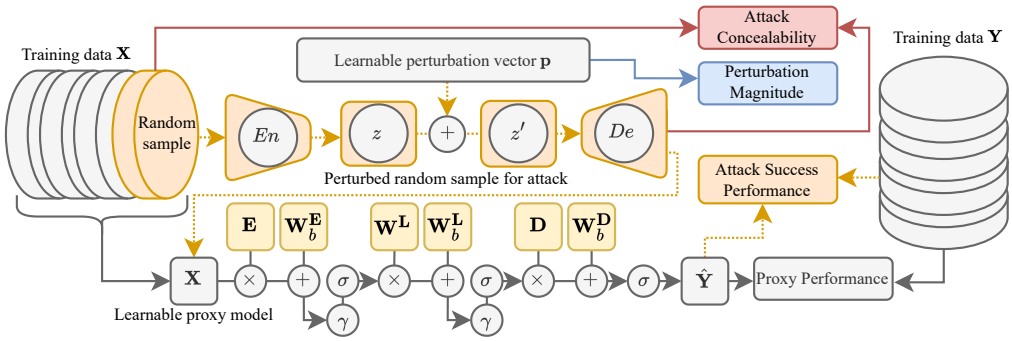


Fig. 2: The overall proposed approach of ML-MAP that jointly optimises a proxy model and perturbation vector $\mathbf{p}$ using input training data $\mathbf{X}$. During training, the proxy performance is recorded by classifying labels for $\mathbf{X}$. Concurrently, a set of randomly chosen samples from $\mathbf{X}^a \subseteq \mathbf{X}$ are chosen to generate adversarial samples $\mathbf{X}_p^a$ using a pre-trained β-VAE and the learned perturbation vector $\mathbf{p}$. The adversarial samples are used to estimate attack success on the proxy model. Furthermore, the attack concealability of the newly generated adversarial samples and the perturbation magnitude is calculated, resulting in a four-objective minimisation problem.

compete in a GAN-based framework. In this framework, the generated adversarial example should continually become more convincing, and harder to detect, while the attacked model should continually improve its robustness against attack while maintaining acceptable classification performance on real, unaltered examples. Here, many-objective adversarial training emerges, involving a strategic optimisation where multiple conflicting objectives—such as attack success, attack robustness, and perturbation concealability—must be simultaneously considered and optimised. In this scenario, adversarial training presents itself as a many-objective optimisation problem (with more than three conflicting objectives that are described in the following sections) that conflict with each other. Formally, an o-objective minimisation problem can be written as follows.

Definition II.1 (Many-objective optimisation)**.**

$$\text{minimise } \mathbf{o}(f) = (o_1(f), o_2(f), ..., o_o(f)) \tag{1}$$

where $\mathbf{o}(f)$ is a vector of objective values for function f, $o_i(f)$ is the i^{th} objective value, $f \in \Omega$ is represented by a decision vector (*i.e.,* learnable parameters of a deep neural network drawn from Ω). The quality of a solution is based on the trade-offs between the objectives. A solution f dominates solution f' (*i.e.*, $f \prec f'$) if:

$$\forall i : o_i(f) \leq o_i(f') \text{ and } \exists j : o_j(f) < o_j(f'). \tag{2}$$

A solution is called a Pareto optimal solution if it is not dominated by any other feasible solution. All the Pareto optimal solutions form a trade-off surface called a Pareto optimal set.

Definition II.2 (Pareto optimal set)**.** A Pareto optimal set of solutions $\mathbb{P}^B$ contain the following:

$$\mathbb{P}^B = \{f : \{f' : f' \prec f, \quad \forall f', f \in \Omega, f' \neq f\} = \emptyset\}. \tag{3}$$

The task of a many-objective optimisation algorithm is to approximate the Pareto front.

III. The ML-MAP approach

The overall proposed approach for *Multi-Label black-box attacks via Many-objective Adversarial Perturbations* (ML-MAP) is outlined in the following section. ML-MAP learns many-objective adversarial perturbations, which are *structural* perturbations that aim to generate convincing attacks on state-of-the-art tabulated multi-label classifiers. Namely, ML-MAP utilises a pre-trained β-VAE to encode input samples and learns by perturbing latent representations to generate adversarial examples. The overall ML-MAP approach is detailed in Fig. 2. First, the proxy model to estimate attack success is detailed, followed by the proposed objective functions to estimate attack success, proxy model performance, and attack invisibility. Next, a many-objective fitness function is proposed to optimise all of the proposed objective functions simultaneously.

A. Notations for multi-label classification

Multi-label classification is a supervised machine learning problem, where an instance can be associated with multiple class labels simultaneously. Let $\mathcal{X} \in \mathbb{R}^D$, $\mathcal{Y} \in \{0,1\}^K$, and $\Omega \in \mathbb{R}^L$ respectively denote the input, output, and learnable parameter space for D features, K labels, and L parameters. Let $\mathcal{P}$ be a joint probability distribution of samples over $\mathcal{X} \times \mathcal{Y}$ and $\theta : \mathbb{R}^D \rightarrow \mathbb{R}^K$ represent a deep neural network drawn from $\Omega \in \mathbb{R}^L$, and trained on N samples drawn from $\mathcal{P}$. An input vector $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X} \in \mathbb{R}^D$, can be associated with an output vector that is a subset of $\mathcal{Y} \in \{0,1\}^K$, *i.e.*, $\mathbf{y} = \{y_1, ..., y_K\}$, where $y_l = 1$ if label l is associated with $\mathbf{x}$, and is otherwise zero. The input feature and label data are defined as $\mathbf{X} \in \mathcal{X}^N$ and $\mathbf{Y} \in \mathcal{Y}^N$, respectively. In this paper, we define Q as the number of perturbation dimensions.

B. Proxy configuration and perturbation parameterisation

A standard feedforward model is used in this paper to estimate the model classification performance and attack success. Due to the tabular nature of the data, our model takes matrices

as inputs and outputs, allowing us to handle all samples simultaneously, rather than individual vectors in standard feedforward networks. The encoding layer $\mathbf{E} : \mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{N\times C}$ with bias $\mathbf{W}_b^{\mathbf{E}}$, maps the input to C embedding dimensions, where $C << D$. The mapped input is row-standardised (γ) before being passed through a feedforward layer ($\mathbf{L}$) with weights $\mathbf{W}^{\mathbf{L}}$ and bias $\mathbf{W}_b^{\mathbf{L}}$. Standardisation is repeated before the decoder $\mathbf{D} : \mathbb{R}^{N\times C} \rightarrow \mathbb{R}^{N\times K}$ with bias $\mathbf{W}_b^{\mathbf{D}}$. Sigmoid activation ($\sigma$) is applied at each layer (and in particular the output layer). The full equation for generating the prediction matrix $\hat{\mathbf{Y}}$ is given by:

$$\hat{\mathbf{Y}} = \sigma(\sigma(\gamma(\sigma(\gamma(\mathbf{XE} + \mathbf{W}_b^{\mathbf{E}}))\mathbf{W}^{\mathbf{L}} + \mathbf{W}_b^{\mathbf{L}}))\mathbf{D} + \mathbf{W}_b^{\mathbf{D}}). \quad (4)$$

The sigmoid function ensures bounded activations, suiting the shallow matrix-based representation, and ReLU and GELU are more tailored to deeper architectures and address specific issues such as vanishing gradients. The tight-bound complexity estimate scales *linearly* with the parameters in the encoding $\Theta(NDC)$ and decoding stages $\Theta(NKC)$, and quadratically with C in the intermediary feedforward step $\Theta(NC^2)$ assuming naive implementation of matrix multiplication.

Furthermore, a perturbation vector $\mathbf{p} \in \mathbb{R}^{Q}$ is learned to generate the adversarial examples via a pre-trained β-VAE following a similar setup as [19]. In an attack scenarios, a set of samples are chosen *at random* and are perturbed using the perturbation vector. This can first be achieved through encoding the adversarial sample into its latent representation z, then adding the perturbation before decoding $z' = z + \epsilon\mathbf{p}$. A scaling factor ϵ is used to control the intensity of the perturbation.

C. Objective functions

Generating convincing multi-label adversarial examples requires a balance between attack success, proxy model performance, attack concealability, and perturbation magnitude. In this paper, we estimate attack success in the following way.

1) Untargeted attack success: The first type of attack simulates real-world adversarial attack scenarios where the intent or strategy of the attacker is unknown, *i.e.,* the attacker can either add, remove, or swap class labels in an attack for a random subset of instances $\mathbf{X}^a \subseteq \mathbf{X}$, $\mathbf{Y}^a \subseteq \mathbf{Y}$. This type of attack manifests as an untargeted attack. To determine attack success, we use the micro-F_1 ($\mathcal{L}^{F_1}$) score to measure the difference between precision and recall of predicted labels of samples pre ($\mathbf{X}^a$) and post-perturbation ($\mathbf{X}_p^a$), averaged across all labels. Formally, given a proxy model f, untargeted attack loss ($\mathcal{L}^U$) is expressed in Eq. (5).

$$\mathcal{L}^U = \min(1 - (\mathcal{L}^{F_1}(f(\mathbf{X}^a), \mathbf{Y}^a) - \mathcal{L}^{F_1}(f(\mathbf{X}_p^a), \mathbf{Y}^a)), 1) \quad (5)$$

The behavior of the untargeted attack loss is that perturbations that *reduce* the classification accuracy, *i.e.,* $\mathcal{L}^{F_1}(f(\mathbf{X}^a), \mathbf{Y}^a)) < \mathcal{L}^{F_1}(f(\mathbf{X}_p^a), \mathbf{Y}^a)$, will reduce the loss value (ranging between 0 and 1), indicating a successful untargeted attack.

2) Proxy performance: To balance attack success with model robustness, we propose to jointly train f and $\mathbf{p}$ in a GAN-like framework. In this scenario, f will have to withstand untargeted attacks from perturbed instances while maintaining good performance on unaltered instances. This can be formulated as the proxy model micro-F_1 loss ($\mathcal{L}^S$), which balances the precision and recall, written in Eq. (6).

$$\mathcal{L}^S = \mathcal{L}^{F_1}(f(\mathbf{X}), \mathbf{Y}) \quad (6)$$

3) Attack concealability: Tabulated data is harder to visualise than image data for the human eye, therefore quantifying the invisibility of tabulated perturbations is harder to define. In this paper, use utilise a distance-based metric to measure the similarity between adversarial samples pre and post-perturbation. This is given by the concealability loss ($\mathcal{L}^C$) Eq. (7).

$$\mathcal{L}^C = \frac{1}{1 + (e^{-\mu_{\mathbf{X}^a}^{MSE}})} \quad (7)$$

where $\mu_{\mathbf{X}^a}^{MSE}$ prescribes the average MSE between pre and post-perturbation adversarial samples.

$$\mu_{\mathbf{X}^a}^{MSE} = \frac{\sum_{i=1}^{|\mathbf{X}^a|} MSE(\mathbf{X}_{i:}^a, \mathbf{X}_{p_{i:}}^a)}{|\mathbf{X}^a|} \quad (8)$$

4) Perturbation magnitude: The final objective function calculates the magnitude of the perturbation vector $\mathbf{p}$, to discourage arbitrarily large perturbations that deviate from the latent representation of each sample. To achieve this, we introduce an ℓ_2-norm-based loss ($\mathcal{L}^N$) bounded between 0 and 1 in Eq. (9).

$$\mathcal{L}^N = \frac{1}{1 + e^{-\|p\|}} \quad (9)$$

D. Many-objective fitness function

All objective functions in Section III-C are designed to be minimised and range between 0 and 1. Therefore, we propose a four-objective minimisation problem consisting of attack success, proxy performance, attack concealability, and perturbation magnitude. The attack success is formulated as the discrepancy between classification performance pre and post-perturbation. We use a Lebesgue measure-based fitness function to navigate the complex landscape of many-objective adversarial training, extending the state-of-the-art Consistent Lebesgue Measure-based Multi-label Learner (CLML) framework [34]. This choice is motivated by the Lebesgue measure's wide use in multi-objective optimisation [35]–[37]. In comparison to traditional multi-objective methods such as dominance-based sorting, the Lebesgue measure can be proven consistent with the underlying desired objective functions [34], which is a desirable property when directly optimising non-convex or discontinuous objective functions such as the micro-F_1 untargeted attack loss [38].

To optimise this fitness landscape, we represent both model weights (f) and perturbation vectors ($\mathbf{p}$) as flattened vectors $[f, p]$, leveraging the covariance matrix adaptation evolutionary strategy (CMA-ES) [20], a robust and widely-used non-convex optimiser [39]–[41]. Extending the CLML framework

allows ML-MAP to optimise the proxy model weights and perturbation vector jointly based on the several non-convex losses proposed in Section III-C. This optimisation framework also motivates the choice of the proxy model, where weights can be sampled from the multi-variate distribution learned by CMA-ES.

Definition III.1 (Lebesgue measure)**.** Given $o = 4$, let $\mathcal{L}^U, \mathcal{L}^S, \mathcal{L}^C$, and $\mathcal{L}^N$ respectively, without loss of generality, represent the untargeted attack loss, proxy performance loss, concealability loss, and perturbation magnitude loss. Let $Z = \{z \in \mathbb{R}^o | z_i \in (0,1), \forall i = 1, \cdots, o\}$ be a set of all possible loss vectors; $R \subset Z$ denote a set of mutually non-dominated loss vectors $\boldsymbol{\mathcal{L}} = (\mathcal{L}^U, \mathcal{L}^S, \mathcal{L}^C, \mathcal{L}^N) \in \mathbb{R}^o$; F denotes the learned function and perturbation $\psi = (f, p) \in F$ that produce $\boldsymbol{\mathcal{L}}$, and a mapping function $g(\psi) = \boldsymbol{\mathcal{L}}$, $g : F \rightarrow \mathbb{R}^o$ (representing the mapping performed by ML-MAP in Fig. 2); and $H(F,R) \subseteq Z$ denotes the set of loss vectors that dominate at least one element of R and are dominated by at least one element of F:

$$H(F,R) := \{z \in Z \mid \exists \psi \in F, \exists r \in R : g(\psi) \prec z \prec r\}. \tag{10}$$

R is initialised to a set containing the unit loss vector $\{1\}^4$, which denotes the worst possible performance on all objective functions and is updated afterward. The Lebesgue measure, $\lambda(H(F,R))$, is quantified as the measure of space within $\mathbb{R}^o$ that is dominated by F over R, representing the "area" where F produces better loss vectors compared to those in R. Formally, it is defined by the integral:

$$\int_{\mathbb{R}^o} \mathbf{1}_{H(F,R)}(\mathbf{z}) d\mathbf{z}. \tag{11}$$

where $\mathbf{1}_{H(F,R)}$ is the indicator function of $H(F,R)$. The indicator function takes the value 1 for all $z \in H(F,R)$, indicating points where the loss vector z falls within the dominated space of F over R, and 0 otherwise.

Definition III.2 (Lebesgue contribution)**.** The contribution of ψ toward the improvement (minimisation) of a set of loss functions $\boldsymbol{\mathcal{L}}$ can be quantified by first measuring the improvement of ψ via the partition function $P(\psi)$:

$$P(\psi) = H(\{\psi\}, R) \backslash H(F \backslash \{\psi\}, R). \tag{12}$$

Hence, the Lebesgue contribution of ψ, $\lambda(P(\psi)) = \int_{\mathbb{R}^o} \mathbf{1}_{P(\psi)}(\mathbf{z}) d\mathbf{z}$, describes it's contribution to minimising $\boldsymbol{\mathcal{L}}$. Ultimately, $\lambda(H(F,R))$, and therefore $\boldsymbol{\mathcal{L}}$, is sought to be optimised via $\lambda(P(\psi))$, *i.e.*, ψ is guided by evaluating its Lebesgue contribution $\lambda(P(\psi))$, thus measuring the new marginal improvement of a loss vector over a set of previously found loss vectors.

To efficiently calculate the Lebesgue contribution (especially when the set of functions F and R are sparsely populated during the early stages of the optimisation), we estimate the Lebesgue measure using Monte Carlo sampling [36]. First, a sampling space $S \subseteq Z$ is defined that entirely contains $P(\psi)$, *i.e.,* $P(\psi) \subseteq S \subseteq Z$. The sampling space can be problem-specific, however, in this paper, it is defined to contain all possible loss vectors between $\{0\}^4$ and $\{1\}^4$. Following, g samples, s_i $i = 1, \cdots, g$, are drawn from S randomly and with uniform probability. Given $\{s_1, \cdots, s_g\}$, the Lebesgue contribution is estimated via $\hat{\lambda}(P(\psi))$ via the following:

$$\hat{\lambda}(P(\psi)) = \lambda(S(\psi)) = \frac{|\{s_i | s_i \in P(\psi)\}|}{g} \tag{13}$$

where $|\{s_i | s_i \in P(\psi)\}|$ is denoted as the number of randomly sampled solutions that exist in $P(\psi)$, also known as *hits*. The probability $\mathbf{p}$ of a sample being *hit* is i.i.d. Bernoulli distributed, therefore, $\hat{\lambda}(P(\psi))$ converges to $\lambda(P(\psi))$ with $\frac{1}{\sqrt{pg}}$ [42].

E. Optimisation process

1) Covariance matrix adaptation: Optimisation of ML-MAP is achieved with the state-of-the-art Consistent Lebesgue Measure-based Multi-label Learner framework (CLML) [34], which uses co-variance matrix adaptation evolutionary strategy (CMA-ES) [43] as a gradient-free numerical optimisation technique. CMA-ES is well-suited for non-convex and non-differentiable optimisation problems. Let $\boldsymbol{\theta}^f$ denote the vector consisting of the learnable parameters of a learned function f a perturbation vector $\mathbf{p}$, where $\boldsymbol{\theta} = [\boldsymbol{\theta}^f, \mathbf{p}]$. CMA-ES samples n solutions from a multi-variate normal distribution as follows:

$$\boldsymbol{\theta}_i \sim \mathbf{m} + \sigma \mathcal{N}_i(0, \mathbf{C}) \quad \forall i, \quad 1 \leq i \leq n \tag{14}$$

where $\boldsymbol{\theta}_i$ are the parameters of the i^{th} function, $1 \leq i \leq n$, $\mathbf{m}$ is the expected density of $\boldsymbol{\theta}_i$, σ the step-size, and $\mathbf{C}$ the covariance matrix. CMA-ES therefore iteratively updates $\mathbf{m}$ and $\mathbf{C}$ via the following:

$$\mathbf{m}^{t+1} = \mathbf{m}^t + \sigma \sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_i^{top} \tag{15}$$

$$\mathbf{C}^{t+1} = (1 - c_{cov})\mathbf{C}^t + c_{cov} \sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_i^{top} (\sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_i^{top})^T \tag{16}$$

where c_{cov} is the learning rate, $\sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_i^{top}$ is the weighted sum of the μ-top ranked solutions at iteration t, where the weights $w_1 > w_2 > \cdots > w_\mu > 0$ and $\sum_{i=1}^{\mu} w_i = 1$. It is also deemed that solutions $\boldsymbol{\theta}_i^{top} \sim \mathbf{m} + \sigma \mathcal{N}_i(0, \mathbf{C})$ are ranked such that $\boldsymbol{\theta}_1^{top} \prec \cdots \prec \boldsymbol{\theta}_\mu^{top}$ and that the μ ranked solutions are a subset of the total number of sampled solutions, i.e. $\mu < n$. This method is referred to as rank-one update.

2) Pseudocode of ML-MAP: The overall process of ML-MAP is detailed in Algorithm 1. The ML-MAP approach learns a covariance matrix for the parameters $\psi = [f, p]$ consisting of the proxy model weights and perturbation vector. The incumbent solution for ψ is optimised iteratively by updating the covariance matrix $\mathbf{C}$ and density vector $\mathbf{m}$ until the maximum number of generations T is met. Initially, the reference set R is set to a unit vector, and the set of function solutions F is set to an empty set. In each generation, λ new incumbent solution is generated from a multivariate normal distribution and added to F. A set of adversarial samples are generated from a random subset of the original instances for each solution. The attack loss and proxy loss are calculated using the training and validation set, and the perturbation magnitude and concealability are calculated. Once all loss

Algorithm 1 Multi-label black box attacks via many-objective adversarial perturbations (ML-MAP)

Input: Maximum epoch T, pre-trained β-VAE, $\mathbf{X}$, $\mathbf{Y}$;
Initialise R^0 to unit vector $\{1\}^4$;
Initialise $F^0 = \{\emptyset\}$;
Initialise $\mathbf{C}^0$ and $\mathbf{m}^0$;
Set $t = 0$;
Set $\psi^0 = [f^0, p^0] = \mathbf{m}^0$;
while $t < T$ **do**
  Generate $\psi^i \sim \mathbf{m}^t + \sigma\mathcal{N}_i(0, \mathbf{C}^t)$, $1 \leq i \leq n$;
  Set $F^{t+1} = \bigcup_{i=1}^{\lambda}\{\psi^i\}$;
  for $\psi^i \in F^{t+1}$ **do**
    Randomly sample $\frac{N}{10}$ solutions $\mathbf{X}^a \subseteq \mathbf{X}$ and $\mathbf{Y}^a \subseteq \mathbf{Y}$ and generate $\mathbf{X}_p^a$;
    Calculate the training (*tra*) and validation (*val*) loss values for $\mathcal{L}^U$, $\mathcal{L}^C$, $\mathcal{L}^S$, and $\mathcal{L}^N$;
    Estimate $\lambda(P(\psi^i))$ over $\mathcal{L}^U_{tra}(\psi^i)$, $\mathcal{L}^S_{tra}(\psi^i)$, $\mathcal{L}^C_{tra}(\psi^i)$, and $\mathcal{L}^N_{tra}(\psi^i)$, and prescribe it as the fitness for ψ^i;
    Archive $\mathcal{L}^U_{val}(\psi^i)$, $\mathcal{L}^S_{val}(\psi^i)$, $\mathcal{L}^C_{val}(\psi^i)$, $\mathcal{L}^N_{val}(\psi^i)$, and the corresponding ψ^i;
  end for
  Update $\mathbf{m}^{t+1}$ using solutions $\forall\psi^i \in F^{t+1}$; Update $\mathbf{C}^{t+1}$ using $\lambda(P(\psi^i))$ as fitness values $\forall\psi^i \in F^{t+1}$;
  Update R^{t+1} by calculating the mutually non-dominated solutions in $R^t \cup F^{t+1}$;
  Set ψ^{t+1} to the best solution in F^{t+1} according to its prescribed fitness value;
end while
Return: Incumbent solutions for each loss function from archive: $\mathcal{L}^{(U)}_{val}(\psi^i)$, $\mathcal{L}^S_{val}(\psi^i)$, $\mathcal{L}^C_{val}(\psi^i)$, $\mathcal{L}^N_{val}(\psi^i)$, and the final incumbent solution ψ^T;

TABLE I: Summary of datasets. D, N, and K correspond to the number of features, instances, and labels, respectively.

Dataset	N	D	K	K^μ	DK/K^μ
flags	194	19	7	3.392	39.21
CAL500	502	68	174	26.044	9,637.54
emotions	593	72	6	1.869	231.14
genbase	662	1186	27	1.252	25,576.68
enron	1702	1001	53	3.378	15,705.45
yeast	2417	103	14	4.237	340.335
tmc2007-500	28,596	500	22	2.158	5,097.31
mediamill	43,907	120	101	4.376	2,769.65
IMDB-F	120,900	1001	28	2.000	14,014

values have been calculated, the Lebesgue contribution is estimated and prescribed as the fitness of ψ^i. After each solution has been evaluated, the density and covariance matrix are updated, and the incumbent solution is updated and stored in an archive concerning each loss function. ML-MAP returns the best solutions for each loss function and the final incumbent solution.

IV. Experiment design

A. Datasets

We conduct the experiments on nine widely-used multi-label datasets, shown in Table I. K^μ (the cardinality) of an instance measures the average number of associated class labels; DK/K^μ, the theoretical maximum complexity of an instance, (*i.e.*, the average feature to label interactions per instance). For each dataset, 30% are partitioned to the test set [44]. The remaining 70% is further split such that 20% is used as a validation set, and the remaining is used for training. We apply normalisation to all numerical features before training. The β-VAE is pre-trained on the training portion of each dataset.

B. Comparative methods

The existing multi-label adversarial training methods are tailored for image data, and are not designed to find many-objective perturbations. Hence, we propose to attack three of the state-of-the-art multi-label deep learning models: DELA [4], PACA [25], and CLIF [3]. We analyse the classification performances of each of DELA, PACA, and CLIF trained on *clean* instances, and later attacked using 10% (up to a maximum of 1,000 instances) of the test set as both unaltered ($\mathbf{X}^a$) and carefully perturbed adversarial instances generated by ML-MAP ($\mathbf{X}_p^a$), *i.e.,* DELA+ML-MAP, PACA+ML-MAP, and CLIF+ML-MAP. To determine the effectiveness of ML-MAP, we also analyse the adversarial distributions pre ($\mathcal{X}^{pre}$) and post perturbation ($\mathcal{X}^{pos}$). This is initially achieved using a non-linear projection to reduce the adversarial samples into components IC_1 and IC_2 using Isomap [45]. Differences between distributions concerning ML-MAP are calcuated by the earth movers distance (EMD) [46] given in Eq. (17):

$$EMD(\mathcal{X}^{pre}, \mathcal{X}^{pos}) = \min_{\gamma} \sum_{i,j} \gamma_{i,j} d(\mathbf{x}_i^{pre}, \mathbf{x}_j^{pos}) \qquad (17)$$

where $d(\cdot)$ represents the Euclidean distance function, $\gamma_{i,j}$ the optimal transportation plan solved by numerical methods, and $\mathbf{x}_i^{pre} \sim \mathcal{X}^{pre}, \mathbf{x}_j^{pos} \sim \mathcal{X}^{pos}$. Due to true distributions of each dataset being unknown, the significance of the differences are determined by a non-parametric Kolmogorov-Smirnov test [47] using a permutation test [48] with a significance level of 5%. Permutation tests are generally robust to violations of assumptions such as normality. The permutation test conducts repeated comparisons between a random pair of adversarial examples sampled from $\mathcal{X}^{pre} \times \mathcal{X}^{pos}$ to estimate the statistical significance between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$. Moreover, by creating a null distribution via randomisation of the data, the permutation test is essentially distribution-free, meaning that it does not rely on any assumptions regarding the underlying characteristics of the distributions.

C. Parameters

Based on previous work on the Lebesgue measure-based optimiser [34], we set $O = 500$ (the maximum number of epochs). Furthermore, the embedding dimension of the proxy model, C, is set to $C = 20$ based on the recommendations in [34]. Moreover, based on initial trial and error, the scaling factor ϵ is set to 0.05. The parameter configurations for DELA, PACA, and CLIF are set to the recommended values in their respective papers [3], [4], [25].

TABLE II: Adversarial attack results. The untargeted attack results are presented in terms of micro-F_1 ($\mathcal{L}^{F_1}$), label ranking average precision ($\mathcal{L}^{AP}$), and Hamming-loss ($\mathcal{L}^{HL}$) of each clean model tested on $\mathbf{X}^a$ and $\mathbf{X}^a_p$ sampled from the test set.

Metric	Method	emotions	flags	CAL500	enron	genbase	yeast	IMDB-F	mediamill	tmc2007-500
$\mathcal{L}^{F_1}(\downarrow)$	CLIF	0.754	0.703	0.361	0.475	0.979	0.610	0.164	0.645	0.783
	CLIF+ML-MAP	**0.429**	**0.488**	**0.330**	**0.392**	**0.000**	**0.000**	**0.070**	**0.484**	**0.280**
	DELA	0.732	0.706	0.351	0.443	1.000	0.608	0.164	0.645	0.774
	DELA+ML-MAP	**0.429**	**0.702**	**0.321**	**0.392**	**0.000**	**0.000**	**0.093**	**0.467**	**0.334**
	PACA	0.778	0.667	0.372	0.491	1.000	0.606	0.100	0.635	0.727
	PACA+ML-MAP	**0.459**	**0.293**	**0.267**	**0.061**	**0.047**	**0.431**	**0.000**	**0.403**	**0.237**
$\mathcal{L}^{AP}(\downarrow)$	CLIF	0.909	0.791	**0.380**	0.603	1.000	0.723	0.590	0.801	0.862
	CLIF+ML-MAP	**0.580**	**0.508**	0.415	**0.450**	**0.519**	**0.601**	**0.578**	**0.576**	**0.396**
	DELA	0.894	0.866	**0.404**	0.611	1.000	0.723	0.608	0.781	0.863
	DELA+ML-MAP	**0.577**	**0.697**	0.416	**0.454**	**0.487**	**0.581**	**0.587**	**0.614**	**0.421**
	PACA	0.915	0.741	0.423	0.608	1.000	0.717	0.633	0.778	0.845
	PACA+ML-MAP	**0.580**	**0.457**	**0.272**	**0.091**	**0.182**	**0.373**	**0.615**	**0.550**	**0.257**
$\mathcal{L}^{HL}(\uparrow)$	CLIF	0.157	0.262	**0.165**	0.057	0.002	0.218	0.050	0.026	0.042
	CLIF+ML-MAP	**0.370**	**0.500**	0.141	**0.063**	**0.047**	**0.301**	**0.964**	**0.037**	**0.342**
	DELA	0.176	0.238	**0.167**	0.059	0.000	0.228	0.055	0.028	0.045
	DELA+ML-MAP	**0.370**	**0.270**	0.143	**0.063**	**0.047**	**0.301**	**0.648**	**0.044**	**0.249**
	PACA	0.148	0.286	0.158	0.062	0.000	0.233	0.041	0.029	0.055
	PACA+ML-MAP	**0.364**	**0.690**	**0.225**	**0.841**	**0.080**	**0.538**	**0.036**	**0.055**	**0.325**

TABLE III: Attack Success Rate (ASR) across datasets.

Method	emotions	flags	CAL500	enron	genbase	yeast	IMDB-F	mediamill	tmc2007-500
CLIF+ML-MAP	83.33%	100%	33.33%	66.0%	100%	84.7%	100%	82.1%	100%
DELA+ML-MAP	83.33%	83.33%	33.33%	66.0%	100%	73.6%	100%	81.3%	98.6%
PACA+ML-MAP	83.33%	100%	93.33%	100%	100%	98.6%	90.8%	91.2%	100%

V. Results and discussions

A. Attack performance

The adversarial attack performance of ML-MAP on CLIF, DELA, and PACA are shown in Table II. The results are shown in terms of micro-F_1 ($\mathcal{L}^{F_1}$), label ranking average precision ($\mathcal{L}^{AP}$), and hamming-loss ($\mathcal{L}^{HL}$). Each model is trained on clean instances and tested on both unaltered instances $\mathbf{X}^a$ and perturbed instances $\mathbf{X}^a_p$ (with both sets sampled from the test set). A successful untargeted attack is indicated by a lower classification performance score for both $\mathcal{L}^{F_1}$ and $\mathcal{L}^{AP}$, and a higher score for $\mathcal{L}^{HL}$. In almost all cases, CLIF, DELA, and PACA are fooled by the perturbed adversarial examples generated by ML-MAP. The only scenario where this is not the cases is on CAL500 for both DELA and PACA models on $\mathcal{L}^{AP}$ and $\mathcal{L}^{HL}$. On some datasets such as genbase and yeast, this can be as extreme as inducing *complete* misclassification (especially in terms of $\mathcal{L}^{F_1}$). This can be expected as genbase has a very low cardinality of approximately one label per instance, and yeast has a relatively small number of labels. In both cases, it is easier for ML-MAP to induce misclassification due to the less complex decision boundaries.

On the other hand, Table III shows the frequently-used Attack Success Rate (ASR) [16], [17], given in Eq. (18):

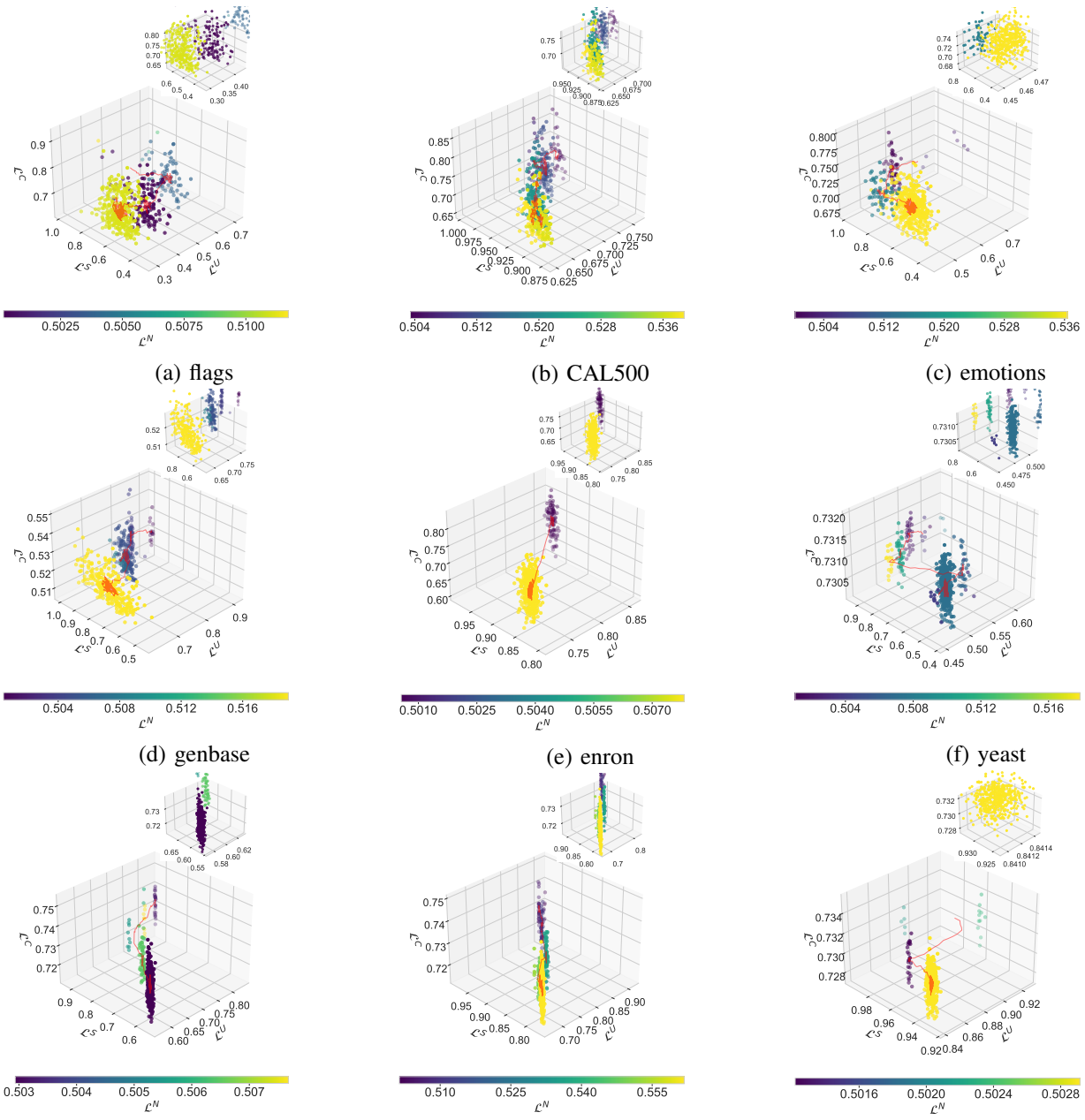
$$ASR = \frac{1(\mathcal{L}^{F_1}(\mathbf{X}^a; f) > \mathcal{L}^{F_1}(\mathbf{X}^a_p; f))}{|\mathbf{X}^a|} \times 100 \qquad (18)$$

where $f \in \{CLIF, DELA, PACA\}$ and $1(\cdot)$ being the indicator function that counts each successful attack, similar to the untargeted attack loss $\mathcal{L}^U$ in Eq. 5. In comparison to the results presented in Table II, the ASR's in Table III do not show the overall degree of induced misclassification, but rather *how many* of the perturbed instances induce misclassification. In most cases, ML-MAP can achieve between 80-100% ASR. On the CAL500 dataset, CLIF+ML-MAP and DELA+ML-MAP achieve only a 33.3% ASR, which coincides with the adversarial attack results in Table II. In terms of the large-scale datasets, ML-MAP can achieve between 90.8% and 100% ASR on IMDB-F, between 81.3% and 91.2% on mediamill, and between 98.6% and 100% on tmc2007-500. In general, based on the results in both Tables II and III, ML-MAP has demonstrated a successful attack on three state-of-the-art deep-learning models for tabulated multi-label classification. The next section analyses the training curves of ML-MAP to better understand the optimisation behaviour that leads to successful perturbations.

B. Many-objective training curves

We propose to analyse the relationships between the attack loss $\mathcal{L}^U$, proxy loss $\mathcal{L}^S$, attack concealability $\mathcal{L}^C$, and perturbation magnitude $\mathcal{L}^N$. Fig. 3 plots the training trajectory of ML-MAP concerning the four loss functions (with $\mathcal{L}^N$ represented in colour), while the red line traces the moving average trajectory of ML-MAP on the approximate loss landscape. Overall, the learning behaviour is a non-smooth descent over the learning-related loss functions $\mathcal{L}^U$ and $\mathcal{L}^S$, and the attack concealability $\mathcal{L}^C$. The perturbation magnitude $\mathcal{L}^N$, on all cases except on yeast and tmc2007-500, increases over time as other loss functions decrease. This is expected as convincing adversarial examples may require some movement from the original pre-perturbation latent space.

Despite the overall trend in minimisation, the loss landspace appears highly discontinuous and non-smooth. In most cases, several distinct clusters of vertically spread solutions are observed, as ML-MAP "jumps" between them during optimisation. These vertically spread clusters consist of similar learning loss functions values for $\mathcal{L}^U$ and $\mathcal{L}^S$, and greater variation in $\mathcal{L}^C$. This can indicate conflict among the loss functions, although $\mathcal{L}^C$ can remain variable even on solutions that exhibit good proxy performance and attack success. In any case, despite a degree of stochasticity, there is some indication



(a) flags (b) CAL500 (c) emotions

(d) genbase (e) enron (f) yeast

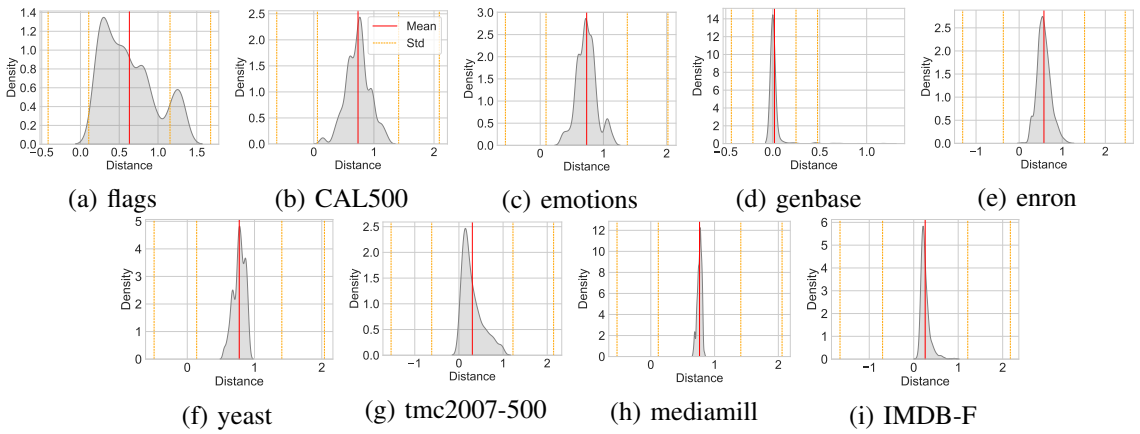
(g) tmc2007-500 (h) mediamill (i) IMDB-F

Fig. 3: The many-objective training curves of ML-MAP plotted against $\mathcal{L}^U$, $\mathcal{L}^S$, and $\mathcal{L}^C$. The colour represents the perturbation weight magnitude $\mathcal{L}^N$. The red line shows the moving average trajectory of ML-MAP. A zoom-in plot is presented at the top right of each subplot to highlight the area of convergence. Results are presented on flags through IMDB-F (a-i).

of non-smooth descent over $\mathcal{L}^U$, $\mathcal{L}^S$, and $\mathcal{L}^C$ loss functions. Recall that ML-MAP specifically optimises the Lebesgue *contribution* $\lambda(P(f))$, *i.e.*, the contribution toward the improvement of the Lebesgue measure $\lambda(H(F,R))$. In this case, by empirical observation, maximising the Lebesgue contribution directly corresponds to the minimisation of the desired loss functions, although conflicting behaviour of $\mathcal{L}^N$ is observed. Due to the tabulated nature of the data, visual examination of the perturbed adversarial samples to determine concealability is not easily achieved. Therefore, we propose to analyse the concealability of the perturbed tabulated adversarial samples in the following section through distribution measurement.

C. Quantifying concealability via distribution analysis

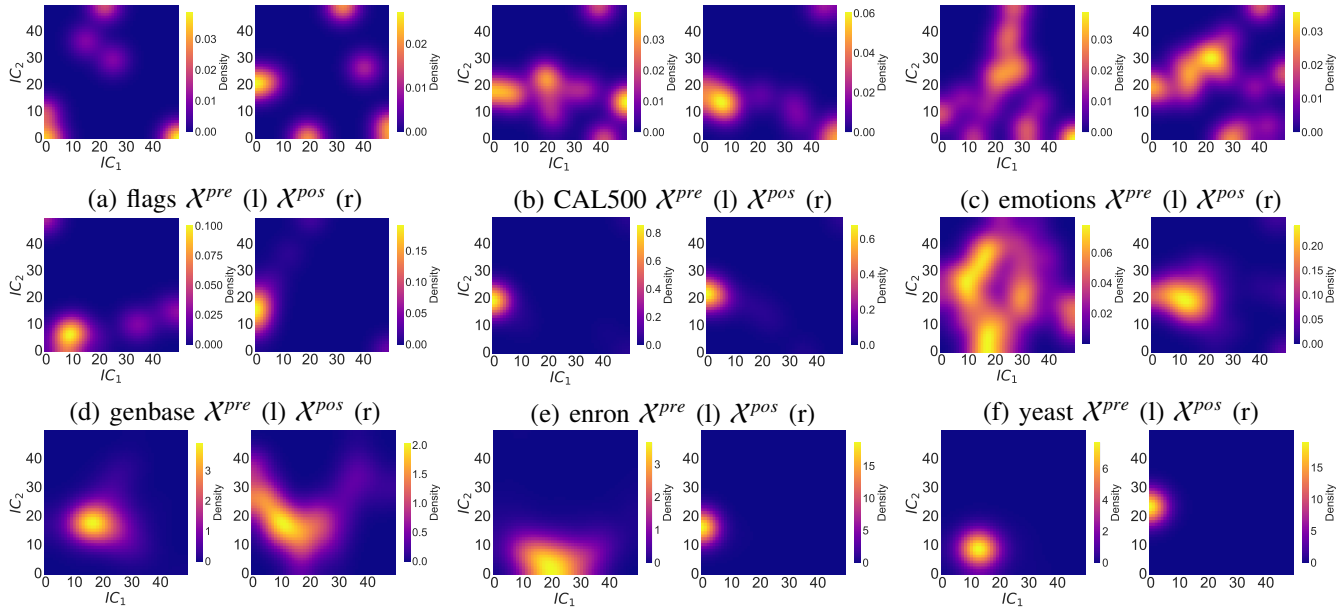
Fig. 4 shows the kernel density estimations of distributions of pairwise Euclidean differences of samples pre ($\mathbf{X}^a$) and post ($\mathbf{X}^a_p$) perturbation. The mean of the distributions are drawn in red, and standard deviations in yellow. The distributions of differences for all datasets are roughly symmetrical (and normal), except on flags. All distributions are centered around 0.5. The symmetry around 0.5 indicates that, on average, perturbations do not significantly shift samples in any direction (within one



(a) flags (b) CAL500 (c) emotions (d) genbase (e) enron

(f) yeast (g) tmc2007-500 (h) mediamill (i) IMDB-F

Fig. 4: Distributions of pairwise Euclidean distances between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$ modelled by kernel density estimation.



(a) flags $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (b) CAL500 $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (c) emotions $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r)

(d) genbase $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (e) enron $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (f) yeast $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r)

(g) tmc2007-500 $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (h) mediamill $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r) (i) IMDB-F $\mathcal{X}^{pre}$ (l) $\mathcal{X}^{pos}$ (r)

Fig. 5: Visualisation of Isomap projections of $\mathcal{X}^{pre}$ (left) and $\mathcal{X}^{pos}$ (right) with Guassian filtering ($\sigma = 4$) and bins ($B = 50$). Data is projected onto two Isomap components: IC_1 and IC_2.

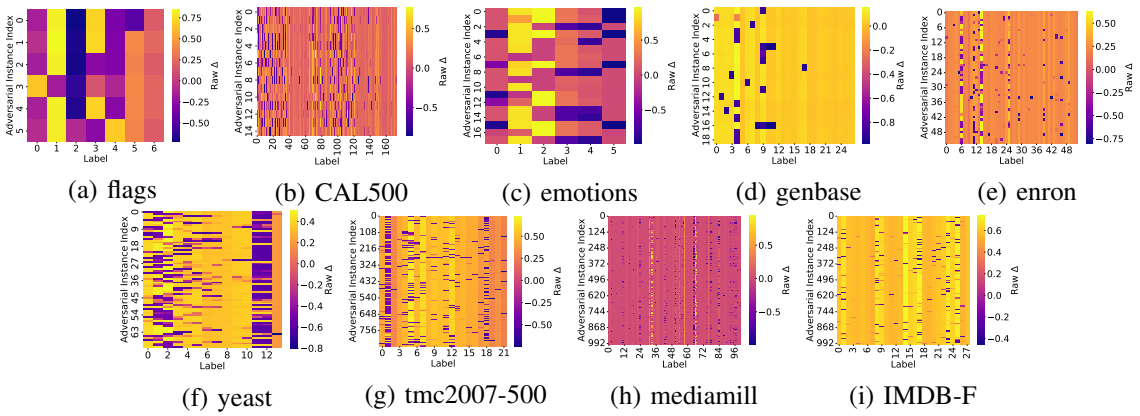
TABLE IV: Observed earth movers distance (EMD) and Kolmogorov-Smirnov test between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$ via a permutation test.

	flags	CAL500	emotions	enron	genbase	yeast	tmc2007-500	mediamill	IMDB-F
EMD	$< \epsilon$	0.0004	$< \epsilon$	0.0016	0.0008	0.0036	0.0172	0.1392	0.0568
p-value	**1.0**	**1.0**	**1.0**	**0.7908**	**0.8928**	**0.4863**	**0.0983**	$< \epsilon$	$< \epsilon$

unit norm). However, the higher degree of variation on the flags datasets suggests that perturbations have a difference effect on the distribution of samples.

To better understand the distributions of adversarial samples pre and post perturbation, Fig. 5 visualises the Isomap projections of two distributions: $\mathbf{X}^a \sim \mathcal{X}^{pre}$ and $\mathbf{X}^a_p \sim \mathcal{X}^{pos}$. Both pre and post perturbation data is projected onto two Isomap components: IC_1 and IC_2, and density is presented in colour. On datasets flags through tmc2007-500 (a-g), the two distributions in Isomap space share similar ranges and concentrations. For example, on CAL500, both $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$ share similar densities between $IC_2 \in [10, 20]$ and $IC_1 \in [0, 20]$; on emotions, between $IC_2 \in [20, 30]$ and $IC_1 \in [20, 30]$; on enron, between $IC_2 \in [10, 20]$ and $IC_1 \in [0, 10]$; on yeast, between $IC_2 \in [10, 30]$ and $IC_1 \in [10, 20]$; and on tmc2007-500, between $IC_2 \in [10, 20]$ and $IC_1 \in [10, 20]$. The scale and location of densities are similar in most cases, except for mediamill and IMDB-F. On mediamill, $\mathcal{X}^{pos}$ appears to shift by 10 units on IC_2 and -20 on IC_1, and on IMDB-F, by 15 on IC_2 and -10 on IC_1.

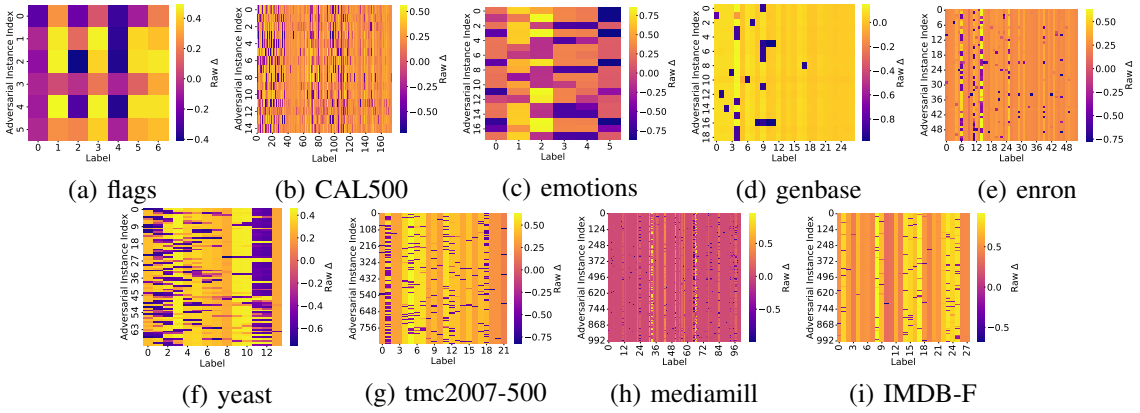
Based on these results, we investigate whether any statistically significant differences exists between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$.



(a) flags (b) CAL500 (c) emotions (d) genbase (e) enron

(f) yeast (g) tmc2007-500 (h) mediamill (i) IMDB-F

Fig. 6: Visualisation of the change in class label confidence of CLIF between $\mathbf{X}^{pre}$ and X^{pos}. Results are presented on flags through IMDB-F (a-i).



(a) flags (b) CAL500 (c) emotions (d) genbase (e) enron

(f) yeast (g) tmc2007-500 (h) mediamill (i) IMDB-F

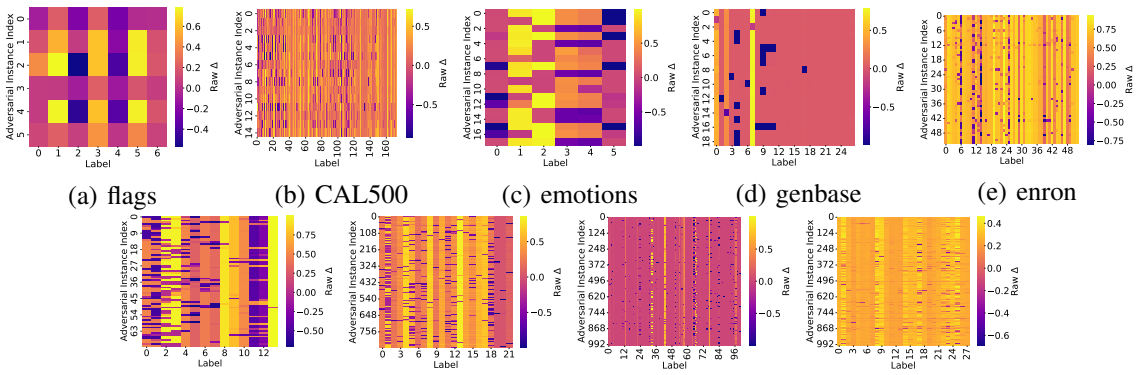
Fig. 7: Visualisation of the change in class label confidence of DELA between $\mathbf{X}^{pre}$ and X^{pos}. Results are presented on flags through IMDB-F (a-i).

Table IV shows the earth movers distance (EMD) along with the Kolmogorov-Smirnov test between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$ via a permutation test. On datasets flags through tmc2007-500, the EMD value is less than 0.02, and the p-value at significance level 5%, fails to reject the null hypothesis of any significant differences between $\mathcal{X}^{pre}$ and $\mathcal{X}^{pos}$. On flags through emotions, the p-value is 1.0, which is potentially due to the relatively small number of samples. On the larger datasets such as mediamill and IMDB-F, the p-values are less than $\epsilon = 1e-5$, which could be due to larger variations in the data that may lead to difficulties in generating perturbations that lie in-distribution while successfully learning the decision boundary vulnerabilities of the classifier.

D. Analysing changes in label confidence

Fig. 6, 7, and 8 plot the average raw change (Δ) in label confidence between $\mathbf{X}^a$ and $\mathbf{X}^a_p$ of each adversarial example for CLIF, DELA, and PACA, respectively. We proceed the analysis by discussing each dataset seperately.

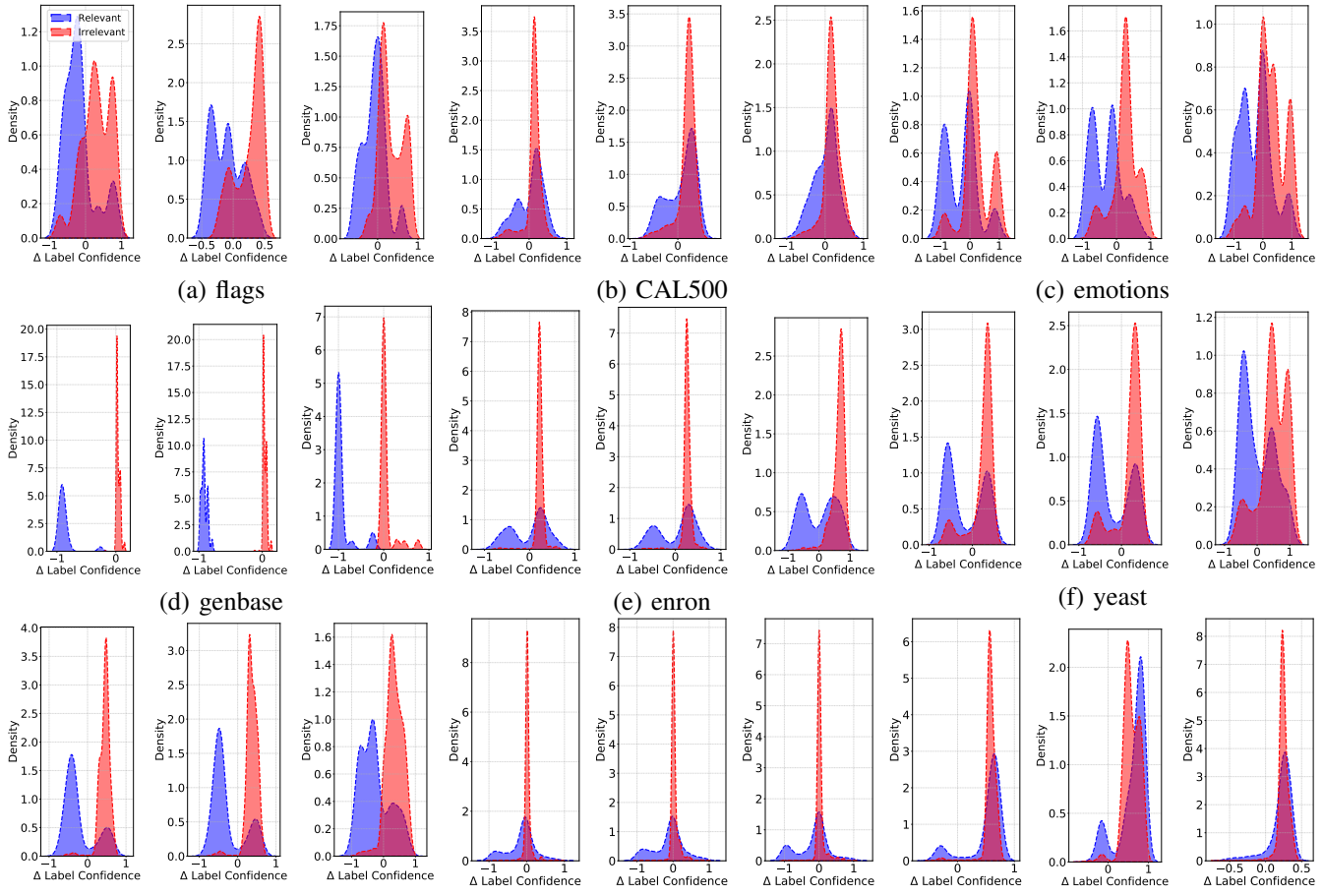
1) Adversarial examples on the flags dataset tend to induce a reduction in confidence for CLIF, DELA, and PACA on labels 0, 2 and 4. On the other hand, labels 1, 3, 5, and 6 tend to increase in confidence.
2) Perturbed adversarial examples on the CAL500 dataset tend to modify (both decrease and increase) class label confidences in bands between labels 0 and 40, and labels 80 and 120. These bands are more clear for CLIF, and less so for DELA and PACA.
3) On the emotions dataset, the most consistent trend is increase in confidences for labels 1, 2, 3, and 4 on all three models. Some examples see a significant decrease in confidence for labels 0, 3, 4, and 5, although this is not consistent across all adversarial examples.
4) On the genbase dataset, most labels are relatively unaltered, except for labels 0 through 12, which some examples tend to induce significantly lower confidence. This is expected since genbase has a lower cardinality than most datasets, which an average of one class label per instance, which suggests attacks can be easier to achieve by only modifying a small subset of labels.
5) On enron, there are several clear bands of labels that have been modified. Approximately label 6, 11, 14, and 24 tend to share in both increases and decreases of label confidence post perturbation.
6) On yeast, labels 0 through 6 tend to have varying increases and decreases in label confidence. Labels 2 and three tend to increase in label confidence, especially



(a) flags (b) CAL500 (c) emotions (d) genbase (e) enron

(f) yeast (g) tmc2007-500 (h) mediamill (i) IMDB-F

Fig. 8: Visualisation of the change (Δ) in class label confidence of PACA between $\mathbf{X}^{pre}$ and X^{pos}. Results are presented on flags through IMDB-F (a-i).



(a) flags (b) CAL500 (c) emotions

(d) genbase (e) enron (f) yeast

(g) tmc2007-500 (h) mediamill (i) IMDB-F

Fig. 9: Visualisation of the change (Δ) in class label confidence of irrelevant and relevant labels predicted by CLIF (left), DELA (middle) and PACA (right) between $\mathbf{X}^{pre}$ and $\mathbf{X}^{pos}$ on each dataset flags through IMDB-F (a-i).

for PACA. On all three models, labels 11 and 12 tend to be heavily reduced in confidence.

7) On tmc2007-500, labels 3-9 and 12-18 tend to increase in confidence, while labels 1, 18 tend to decrease. Notably, perturbations seem to induce decreases on labels 18-21 for PACA.
8) Most labels for mediamill are unaltered, except few between labels 24-36 and 60-72. There are some positive changes to confidence between labels 72-84 on PACA.
9) On IMDB-F, there are almost no negative decrease in confidences for CLIF and PACA. For DELA, a few class labels are decreased in confidence between labels 0 and 12, and label 19.

To better understand the changes in label confidence, Fig.

9 plots the changes (Δ) in class label confidence of irrelevant (red) and relevant (blue) labels predicted by CLIF, DELA, and PACA on all datasets. Adversarial perturbations appear to induce changes that can exploit similar vulnerabilities in decision boundaries across all three models, *i.e.,* the distributions between irrelevant and relevant label confidence changes remain similar across CLIF, DELA, and PACA, although the overall patterns themselves vary across datasets.

1) The patterns across flags, emotions, yeast and tmc2007-500 are mostly similar. On these datasets, the relevant and irrelevant distributions appear to mirror each other. Namely, increasing both false positives and false negatives. On emotions, both distributions spike around 0, although the overall mirror pattern remains.
2) On CAL500, enron and mediamill, the irrelevant label change appears to spike near 0, which suggests that the primary influence to inducing misclassification was to increase false negatives.
3) On genbase, both distributions are heavily mirrored and unimodal for all three models, maximising false positive and false negatives. However, and most importantly, PACA does not appear to induce false positives, relying more on false negatives to induce misclassification. On this specific dataset, it is possible that the prototype-based classification approach in PACA, where label-specific features are generated based on distances between positive and negative probabilistic prototypes, tend to make PACA more resilient to the types of perturbations that affect false positives.
4) On IMDB-F, both false positive and true positives seem to occur most often as both distributions are left tailed. This suggests that the exploited vulnerability in the decision boundary of IMDB-F that induces misclassification also tends to improve the overall true positives, which could also imply that it is easier to classify everything as a positive label.

VI. Conclusions

Deep-learning has unequivocally advanced the field of tabulated multi-label learning, and has thus become state-of-the-art. However, existing adversarial works are either designed for images, white-box-attacks (that requires inside knowledge of the model), or single-objective optimisation, which is a critical shortcoming in research that aims to generate convincing adversarial examples for tabulated multi-label data that can balance classifier robustness, attack success, and concealability. To address these concerns, this paper proposes a highly novel adversarial training method for tabulated multi-label problems, namely *Multi-label black-box attacks via Many-objective Adversarial Perturbations (ML-MAP)*. We primarily propose a novel adversarial training framework for multi-label classification that can generate convincing structured perturbations for tabulated data. Moreover, ML-MAP is designed to learn convincing perturbations using a proxy model to simulate a black-box attack scenario where the true model information is unavailable. This is particularly significant, as ML-MAP can successfully generate perturbations that are capable of inducing misclassification on state-of-the-art deep-learning models that are trained on clean instances. The effectiveness of the perturbations learned by ML-MAP can be attributed to the design of a many-objective optimisation problem that balances the proxy model robustness, attack invisibility, and attack success, which helps generate convincing adversarial examples. In addition to the high attack success rates, and unequivocal success in inducing misclassification across multiple datasets, ML-MAP can also learn perturbations that produce adversarial examples that are statistically unlikely to fall out-of-distribution on almost all datasets, therefore concealing attack. Nonetheless, our analysis also shows that ML-MAP is capable of learning perturbations that can automatically induce potential false positives and false negatives by indirectly learning vulnerabilities in decision boundaries that can increase confidence in irrelevant labels and reduce confidence in relevant labels.

References

[1] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, "Learning deep latent spaces for multi-label classification," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 2838–2844.

[2] J. Bai, S. Kong, and C. Gomes, "Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI'20, 2021.

[3] J.-Y. Hang and M.-L. Zhang, "Collaborative learning of label semantics and deep label-specific features for multi-label classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9860–9871, 2022.

[4] ——, "Dual perspective of label-specific feature learning for multi-label classification," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8375–8386.

[5] Y. Lu, W. Li, H. Li, and X. Jia, "Predicting label distribution from tie-allowed multi-label ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2023.

[6] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, "Cross-modality attention with semantic graph embedding for multi-label classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 709–12 716.

[7] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, "Multi-label classification with label graph superimposing," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 12 265–12 272, 2020.

[8] F. Zhou, S. Huang, and Y. Xing, "Deep semantic dictionary learning for multi-label image classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3572–3580.

[9] J. Yuan, S. Chen, Y. Zhang, Z. Shi, X. Geng, J. Fan, and Y. Rui, "Graph attention transformer network for multi-label image classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 4, pp. 1–16, 2023.

[10] Z. Liu, Q. Cao, Q. Jin, J. Lin, G. Lv, and K. Chen, "Accurate detection of arrhythmias on raw electrocardiogram images: An aggregation attention multi-label model for diagnostic assistance," *Medical Engineering & Physics*, vol. 114, p. 103964, 2023.

[11] G. Wu and J. Zhu, "Multi-label classification: do hamming loss and subset accuracy really conflict with each other?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 3130–3140, 2020.

[12] R. Abdelfattah, Q. Guo, X. Li, X. Wang, and S. Wang, "Cdul: Clip-driven unsupervised learning for multi-label image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1348–1357.

[13] M. Li, D. Wang, X. Liu, Z. Zeng, R. Lu, B. Chen, and M. Zhou, "Patchct: Aligning patch set and label set with conditional transport for multi-label image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 348–15 358.

[14] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic,*

September 23-27, 2013, Proceedings, Part III 13. Springer, 2013, pp. 387–402.

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

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

[17] L. Kong, W. Luo, H. Zhang, Y. Liu, and Y. Shi, "Evolutionary multilabel adversarial examples: An effective black-box attack," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 3, pp. 562–572, 2023.

[18] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-vae: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2016.

[19] S. Wang, S. Chen, T. Chen, S. Nepal, C. Rudolph, and M. Grobler, "Generating semantic adversarial examples via feature manipulation in latent space," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[20] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE, 1996, pp. 312–317.

[21] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, pp. 333–359, 2011.

[22] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The emerging trends of multi-label learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7955–7974, 2021.

[23] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 115–124.

[24] L. Xiao, X. Huang, B. Chen, and L. Jing, "Label-specific document representation for multi-label text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 466–475.

[25] J.-Y. Hang, M.-L. Zhang, Y. Feng, and X. Song, "End-to-end probabilistic label-specific feature learning for multi-label classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, pp. 6847–6855, Jun. 2022.

[26] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2014.

[27] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label specific features for multi-label classification," in *2015 IEEE International Conference on Data Mining*, 2015, pp. 181–190.

[28] ——, "Joint feature selection and classification for multilabel learning," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 876–889, 2017.

[29] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[30] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[31] Q. Song, H. Jin, X. Huang, and X. Hu, "Multi-label adversarial perturbations," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1242–1247.

[32] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[33] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[34] K. Demir, B. Nguyen, B. Xue, and M. Zhang, "A Consistent Lebesgue Measure for Multi-label Learning," 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.00324

[35] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.

[36] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using monte carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems: Proceedings of the 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand, 7th-12th January 2008*. Springer, 2010, pp. 313–326.

[37] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[38] W. Gao and Z.-H. Zhou, "On the consistency of multi-label learning," in *Proceedings of the 24th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, S. M. Kakade and U. von Luxburg, Eds., vol. 19. Budapest, Hungary: PMLR, 09–11 Jun 2011, pp. 341–358.

[39] K. Smith-Miles and X. Geng, "Revisiting facial age estimation with new insights from instance space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2689–2697, 2020.

[40] E. Sarafian, M. Sinay, Y. Louzoun, N. Agmon, and S. Kraus, "Explicit gradient learning for black-box optimization." in *Proceedings of the 37th International Conference on International Conference on Machine Learning*, ser. ICML'20, 2020, pp. 8480–8490.

[41] M. Nomura, S. Watanabe, Y. Akimoto, Y. Ozaki, and M. Onishi, "Warm starting cma-es for hyperparameter optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 9188–9196.

[42] P. S. Laplace, *Théorie analytique des probabilités*. Courcier, 1814.

[43] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[44] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*. Springer, 2011, pp. 145–158.

[45] M. Balasubramanian and E. L. Schwartz, "The isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, pp. 7–7, 2002.

[46] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 2000.

[47] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[48] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance." *Journal of Machine Learning Research*, vol. 11, no. 6, 2010.