國立臺灣大學電機資訊學院資訊工程研究所

博士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

從表徵學習探究具挑戰性視覺分類問題

Representation Learning for Challenging Visual Classification Problems

許雁棋

Yen-Chi Hsu

指導教授: 李明穗 博士、劉庭祿 博士

Advisor: Ming-Sui Lee, Ph.D., Tyng-Luh Liu, Ph.D.

中華民國 112 年 7 月

July, 2023

doi:10.6342/NTU202301574

國立臺灣大學博士學位論文
口試委員會審定書

PhD DISSERTATION ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

從表徵學習探究具挑戰性視覺分類問題

Representation Learning for Challenging Visual Classification Problems

本論文係許雁棋君（學號 D06922021）在國立臺灣大學資訊工程學系完成之博士學位論文，於民國 112 年 6 月 28 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 28 June 2023 have examined a PhD dissertation entitled above presented by HSU,YEN-CHI (student ID: D06922021) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

李明穗 劉庭祿 王鈺強

（指導教授 Advisor）

陳祝嵩 陳煥宗 莊永裕

系主任/所長 Director: 洪士灝

doi:10.6342/NTU202301574

Acknowledgements

I would like to express my heartfelt gratitude and appreciation to everyone who has supported me throughout the completion of this doctoral dissertation.

First and foremost, I would like to sincerely thank my co-advisor, Research Fellow Liu, Tyng-Luh (劉庭祿研究員). Thank you for giving me the opportunity to serve as a research assistant and guiding me through the process of pursuing a Ph.D. Your expertise and guidance have played a crucial role in my doctoral studies, teaching me and leading me to develop independent research capabilities. Your inspiration and encouragement have enabled me to overcome challenges and strive for excellence. I am honored to have been your student and have benefited tremendously from your mentorship.

Additionally, I want to extend my heartfelt thanks to my advisor, Professor Lee, Ming-Sui (李明穗教授). Thank you for granting me the opportunity to pursue a Ph.D. and for sharing your life experiences and academic journey, providing me with valuable support. You have also facilitated collaborations with medical professionals at National Taiwan University, guiding and leading me in conducting research and making breakthroughs from different perspectives.

I would also like to express my gratitude to all the members of the laboratory, particularly Hong, Cheng-Yao (洪晟耀). Thank you for your collaboration and support through-

doi:10.6342/NTU202301574

out the research process. Our discussions and exchanges have enriched my research ideas and provided invaluable feedback and suggestions. Your friendship and teamwork have made me feel warm and inspired on this journey.

Likewise, I want to thank my family and loved ones, especially my parents and my wife. Thank you for your unwavering support and selfless love. Your encouragement and trust have allowed me to pursue my academic goals and become the person I am today.

Finally, I want to thank all the friends and colleagues who have provided assistance and support in my doctoral research. Your friendship and encouragement have given me strength and made me feel less alone on this journey. I appreciate your expertise and valuable advice.

I have encountered many challenges and experienced personal growth throughout the completion of this dissertation. It is a significant milestone in my academic career, and I could not have achieved it without the help of the individuals mentioned above.

Once again, I would like to express my heartfelt gratitude to everyone who has supported and helped me. Your contributions will forever be cherished in my heart.

With sincerest thanks,

Hsu, Yen-Chi (許雁棋)

doi:10.6342/NTU202301574

摘要

本文對具有挑戰性的分類任務的特徵表示進行了全面探索。研究工作聚焦於四個關鍵方面：多實例數據分佈的學習、無標籤數據分佈的學習、現實世界數據分佈的學習以及順序數據分佈的學習。首先在多實例數據的情境下，我們引入了一種新穎的跨注意力池化方法，結合注意力引導，有效地表示給定特定查詢的一組實例。所提出的方法捕捉了關鍵特徵，實現了準確的分類。接著，爲應對無標籤數據分佈的挑戰，本文提出了一種解耦對比學習框架。該框架緩解了對比學習中大批量數據的問題，並討論了各種方法對後續分類任務的影響。然後，在面對現實世界數據分佈帶來的獨特挑戰時，例如細粒度和長尾問題，我們提出了一種自適應批次混淆規範（ABC-Norm）。該方法同時解決了這兩項問題，實現了針對現實世界情境的表徵學習。最後，在處理多個僞造組件和順序問題的深僞影像的表徵問題時，我們將該問題分解爲深僞分類、多標籤定位和僞造順序恢復的任務，並提出了一種多標籤排序機制，結合對比的多實例情境，以恢復順序數據分佈。透過廣泛的實驗，本文爲分類任務的表徵學習做出了重要貢獻，我們討論了最先進的方法，並且在每個方面中的挑戰都提出了新穎的方法並取得突出的研究成果。

關鍵字：表徵學習、多實例學習、自監督學習、現實世界分佈、順序數據分佈。

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

Abstract

This thesis presents a comprehensive exploration of feature representations for challenging classification tasks. The research efforts focus on four key aspects: learning with multi-instance data distributions, learning with unlabeled data distributions, learning with real-world data distributions, and learning with ordering data distributions.

In the context of multi-instance data, we introduce a novel cross-attention pooling approach, incorporating attention guidance, to effectively represent a bag of instances given a specific query. The proposed method captures essential features and enables accurate classification. To address the challenge of unlabeled data distributions, a decoupled contrastive learning framework is proposed. This framework alleviates the issue of large batch sizes in contrastive learning and discusses the implications of various approaches for subsequent classification tasks. Real-world data distributions present unique challenges, such as fine-grained and long-tailed issues. To tackle these complexities, we present an adaptive batch confusion norm (ABC-Norm) that addresses both issues and enables the

doi:10.6342/NTU202301574

learning of robust feature representations tailored to real-world scenarios. Finally, we address the representation of deepfake images, which involve multiple manipulated components and ordering issues. The problem is decomposed into deepfake classification, multi-label localization, and manipulation ordering tasks. A multi-label ranking mechanism, combined with a contrastive multi-instance scenario, is proposed to recover the ordering data distributions.

Through algorithmic design and extensive experimentation, this thesis contributes to the advancement of representation learning for classification tasks. It discusses state-of-the-art methodologies, pinpoints the challenges associated with each aspect, and proposes effective research approaches. The findings of this research provide useful insights into the field of representation learning for tackling challenging classification tasks.

Keywords: representation learning, multi-instance learning, self-supervised, real-world distributions, ordering data distributions.

doi:10.6342/NTU202301574

Contents

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

List of Figures

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

List of Tables

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

Chapter 1 Introduction

Representation learning is a fascinating field of research that revolves around the acquisition of concise and meaningful numerical representations for various types of signal sources. The predominant signal sources in representation learning include video, text, audio, and images. The primary objective of this thesis is to harness these learned representations specifically for image-based tasks, including information retrieval and classification. An exemplary illustration of this concept can be observed in the popular scenario of searching for images on Google. When a user enters text keywords, Google employs representation learning techniques to retrieve and present a set of images that are most relevant to the provided words.

In the field of computer vision, representation learning is commonly accomplished by training deep learning models to transform raw input into numerical vectors, also known as embeddings. When dealing with image data, these numerical vectors are typically multidimensional to capture and preserve the underlying information of the objects within the images.

An effective representation model offers numerous advantages. Firstly, it provides optimal initial weights for other related tasks, such as object detection and semantic segmentation. By leveraging the knowledge encoded within the learned representations, these

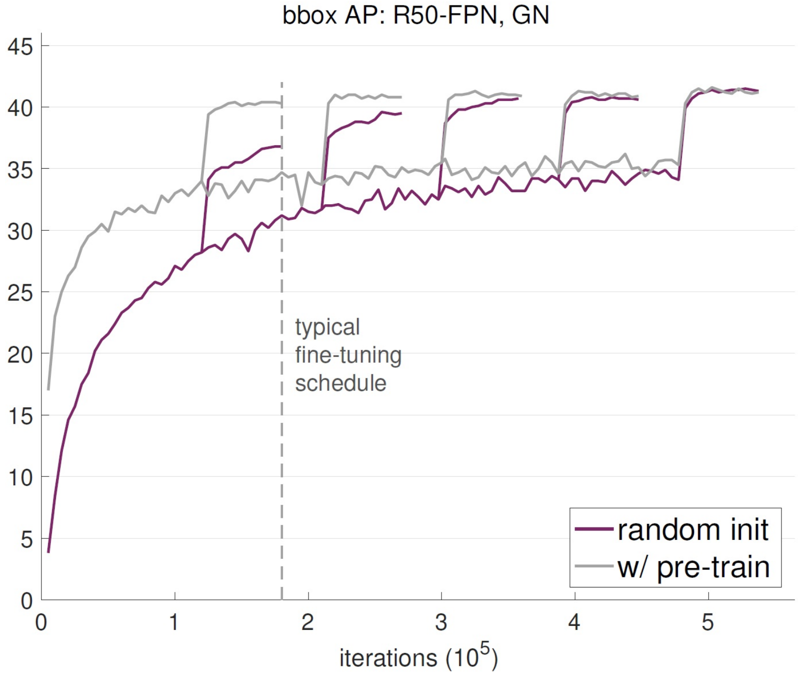


Figure 1.1: The model trained from random initialization needs more iterations to converge in the object detection task.

subsequent tasks can benefit from a solid starting point, facilitating more accurate and efficient results. Secondly, a well-designed representation model can significantly expedite the training process. By utilizing a pre-trained model with well-established embeddings, the learning process can converge more rapidly, saving valuable time and computational resources. Figure 1.1 [52] visually demonstrates the advantages of employing a high-quality pre-trained model, which supplies an excellent initial representation for downstream tasks.

In this thesis, we present an extensive landscape of in-depth research efforts focused on addressing a series for exploring feature representations for challenging classification tasks. The primary objective is to investigate feature representations from different perspectives, encompassing the following aspects:

doi:10.6342/NTU202301574

Learning with Multi-Instance Data Distributions: This aspect explores the representation of a collection of instances, commonly referred to as a "bag," when given a specific query. The goal is to develop effective techniques that capture essential features encompassing the entire bag, enabling accurate classification.

Learning with Unlabeled Data Distributions: This aspect lies in representing images that lack any form of annotation or labeling. The challenge is to devise methodologies that can extract meaningful representations from these unlabeled images, facilitating subsequent classification tasks.

Learning with Real-World Data Distributions: This aspect delves into representations derived from real-world data distributions, which often encompass fine-grained and long-tailed issues. The focus is on developing robust feature representations that can effectively handle the challenges posed by such complex distributions, ultimately leading to improved classification performance.

Learning with Ordering Data Distributions: This aspect specifically addresses the representation of deepfake images, which typically involve multiple manipulated components with ordering issues. The objective is to devise representation learning techniques that can effectively capture and encode the intricate manipulations within these images, enabling accurate classification and recovering the order of manipulations.

We provide a comprehensive overview of the aforementioned aspects, highlighting the challenges associated with each and discussing state-of-the-art methodologies proposed in the literature. The presented research landscape contributes to the advancement of classification tasks by enabling the development of more powerful and robust feature representations.

Chapter 2 Learning with Multi-Instance Data Distributions

2.1 Introduction

Supervised learning techniques that rely on deep neural networks have made significant progress in active research fields of artificial intelligence such as classification [53, 112], the mainstream of computer vision applications. In solving an image classification problem, each training sample often comprises a raw image and the corresponding class/category label. However, such a classification setting may not be sufficient to satisfactorily account for real-life applications nowadays. With the rapid advances of machine learning research, it becomes feasible to simultaneously explore all the useful information of either an image or a batch of images. In other words, image classification is no longer restricted to the problem where an image is labeled as a single category. Among the variants of classification frameworks, *e.g.*, as illustrated in Figure 2.1, we aim to address the multi-instance multi-label learning (MIML) in [150] from a novel viewpoint of learning through queries.

doi:10.6342/NTU202301574

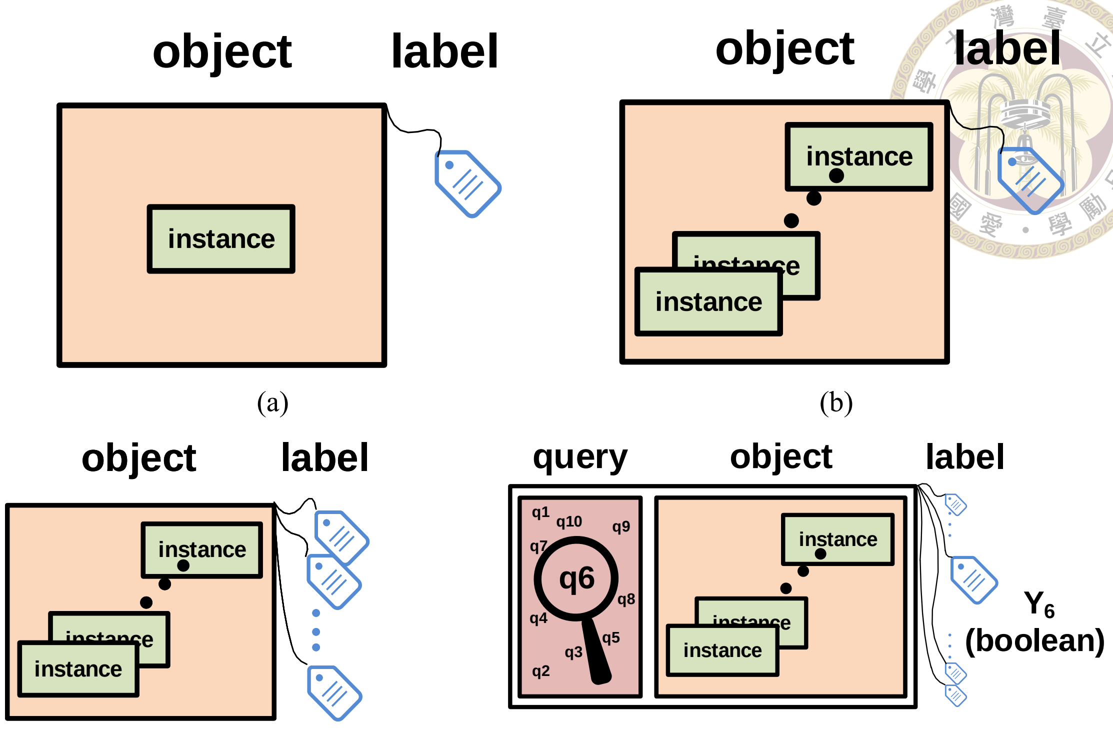
The MIML problem is characterized by that an object or a bag consists of several instances with multiple class labels. While MIMLSVM [149] is proposed to deal with the problem, deep MIML in [42] is shown to be more effective than other traditional methods. Notably, existing supervised learning approaches for MIML are provided with the full binary label vector associated with each training bag, and thus have access to the presence of any class label in a bag. Such a learning setting requires extensive manual efforts in annotating the vast amount of training bags. In our method, a query-driven multiple instance learning (qMIL) framework is proposed to tackle MIML without specifying the full binary label vector. In fact, the qMIL formulation requires only a binary label for each bag along with the corresponding label query. The proposed method thus has two main advantages. First, it is flexible to introduce new classes into the model without the need to modify the labeling information in the existing training data and the classification layer. Second, the query mechanism enables qMIL to inherently and additionally perform zero-shot classification in a crude way.

2.2 Related Work

For the ease of discussion, we divide the literature survey of relevant techniques into three groups, namely, *multi-instance learning*, *attention mechanism*, and *zero-shot learning*.

Multi-instance Learning The MIL paradigm deals with those learning problems for which labels only exist for sets of data points. A set of data points is typically termed as a bag and each data point is considered as an instance. Following [31], a bag is said to be positive with respect to a certain binary label if at least one instance within the

doi:10.6342/NTU202301574



(a) (b) (c) (d)

Figure 2.1: Variants of supervised-learning tasks: (a) Classification (b) Multi-instance learning (MIL) (c) Multi-instance multi-label learning (MIML) (d) Query-driven multi-instance learning (qMIL).

bag is positive. The strategy of [22] maps each bag into a feature space defined by the instances in the training bags via an instance similarity measure and ℓ_1-norm SVM is applied to select important features as well as construct classifiers simultaneously. In [79], the authors construct nearest-neighbor graphs among instances and uncover positive instances within positively-labeled groups. The MIL formulation in [95] is designed to learn a semantic segmentation model based on weak image-level labels. More recently, [128] employs neural networks that aim at solving the MIL problems in an end-to-end manner. An attention-based neural network model is proposed in [59] to detect positive instances automatically. In [29], a recurrent neural network model called MI-RNN is developed to find out the *signature*, which is linked to those positive instances in a bag. Among the aforementioned classical MIL problems, each bag has only one corresponding label. However, in many practical applications, a complex bag (such as an image), which

doi:10.6342/NTU202301574

contains various instances like pixels, may have more than one relevant label. The MIML framework of [149] is established to tackle the complicated scene classification. Over the past few years, assorted algorithms, ranging from traditional, *e.g.*, SVM [6, 91] and k-nearest neighbor (KNN) [142], to popular like deep neural network learning [42], have been proposed to address the MIML problem.

Attention Mechanism The attention mechanism has a significant impact on designing deep learning architecture to solve challenging applications in artificial intelligence, including image captioning, *e.g.*, [134, 137], visual question answering, *e.g.*, [85], and machine translation, *e.g.*, [86]. For solving the MIL or MIML problems, as the individual instance labels of training data are not given, the attention distribution is often learned implicitly via optimizing the bag-level objective function.

Zero-shot Learning A critical limitation of deep learning is that it often takes a massive amount of samples to train a satisfactory model, and the classifier, such as trained by cats and dogs, can only classify cats and dogs. This means that the classifier is not able to be directly applied to recognize other species. On the contrary, zero-shot learning (ZSL) refers to the learning of classifying samples of unseen categories. It implies that the training classes and the zero-shot testing classes are different. For example, the ZSL algorithm proposed in [71] guides the model to classify unseen categories, empowering machines the capacity for reasoning and true intelligence.

Our Approach To establish the proposed qMIL, we first need to generate a training dataset of bags. Specifically, for each query about a certain class label, a bag of instances from randomly-selected classes are generated. If there exists at least one instance from

doi:10.6342/NTU202301574

the query class, the underlying bag is said to be positive and its binary label is set to 1. Otherwise, it is a negative bag with label 0. Notice that only the examples from the classes of interest can be included in a bag. Our setting is different from that in [29] where a positive bag is composed of one or a few positive instances and several negative instances, which are usually noise, *i.e.*, not from any of the underlying classes of interest. In qMIL, each training sample/bag is annotated with a binary label, rather than a binary label vector over all classes as in the MIML setting. However, the proposed method still satisfactorily solves the MIML problem in that a proper bag representation for classification can be obtained by qMIL via more effectively estimating the query-adapted attention distribution over instances within a bag. We summarize the main advantages of the proposed qMIL over other existing techniques below.

1. The qMIL formulation is flexible. When new data of additional classes are included, all binary labelings of the existing training data remain the same, whereas annotating with a full label vector as in the conventional MIML needs to modify all the labeling information.

2. The qMIL network architecture is general. When additional new classes are introduced, the network architecture remains the same. It can be readily fine-tuned to classify the new classes by generating the queries of new classes and the corresponding training bags. However, with the MIML architecture, one would need to expand the classification layer to account for the new classes.

3. The qMIL framework enables zero-shot classification. When data of unseen classes are added in the testing bags, we perform iterative queries to first remove most positive instances of seen classes from a given testing bag, and then compute a

more reliable attention distribution for each query of an unseen class to decide if any positive instance of an unseen class is present or not.

2.3 Approach

The qMIL framework is developed to learn a neural network model that adapts to the underlying query and dynamically yields a proper bag representation for classification. To comprehend the main ideas, we focus on describing: 1) how to generate the training data; 2) how to establish a generalized compatibility measure to facilitate the query-visual co-embedding; 3) how to employ label-dependent regularization to yield the desirable attention distribution over bag instances; and 4) how to use attention pooling to obtain the query-adapted bag representation for classification. Finally, we detail a handy procedure resulted from qMIL to carry out zero-shot classification via iterative queries.

2.3.1 The qMIL Problem

In the classical supervised learning such as multi-class classification, the aim is to train a model that predicts a target label $y \in \{1, \dots, C\}$ for a given test sample $\mathbf{x} \in \mathbb{R}^D$, where C represents the number of classes. However, in the formulation of qMIL, each example is represented as a bag of instances, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{K_X}\}$, where K_X is the number of instances and could vary over bags with a pre-specified upper bound K. Notice that neither dependency nor ordering relationships are considered in generating the instances for each bag.

To incorporate the query mechanism into qMIL, we have a set of C queries, $Q = \{q_1, \dots, q_C\}$, where the query q_c inquires the existence of class label c in a bag, and is

doi:10.6342/NTU202301574

encoded with the corresponding class name/word. The proposed qMIL implicitly solves a more challenging MIML problem than the conventional one. The critical distinction is that each bag X in the training data of qMIL comes with only a single binary ground truth Y indicating the existence of at least one instance of a particular class in X, while the original MIML setting requires a full C-dimensional binary vector describing the presence of all the class labels in X. When $C = 1$, this is exactly the form of training data used for solving a binary MIL problem. For $C > 1$, we use a triplet (X, Y, q) to indicate that the bag label Y depends on the query $q \in Q$ and is defined by

$$Y = \begin{cases} 0, & \text{iff } \sum_{k=1}^{K_X} \mathbb{I}(q \equiv y_k) = 0, \\ 1, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $y_k \in \{1, \ldots, C\}$ is the class label of the instance $\mathbf{x}_k$ in X. The notation $\mathbb{I}(q \equiv y_k)$ is an indicator function for signaling whether the query q concerns the label y_k. We emphasize that the instance-level labels y_k are not available in learning the qMIL model. They are included in (2.1) solely for providing an analytic form in defining the bag label Y with respect to the query q.

With (2.1), it is insightful to describe how the training data of qMIL are generated. Suppose we intend to work with a query subset, $Q' \subseteq Q$, and N training bags. Thus, for each query $q \in Q'$, we generate $N/|Q'|$ bags, which can be divided into two equal-numbered positive and negative subsets, denoted as $\{(X_i^+, Y_i = 1, q)\} \cup \{(X_i^-, Y_i = 0, q)\}$. The total number of instances in each bag is randomly decided with an upper bound K, and only instances with a class label in $\{1, \ldots, C\}$ are considered. These $|Q'|$ query-dependent collections of bags form the final training dataset $\mathcal{S}$ of N bags. It indicates that the training procedure considers equal number of positive and negative training bags for

doi:10.6342/NTU202301574

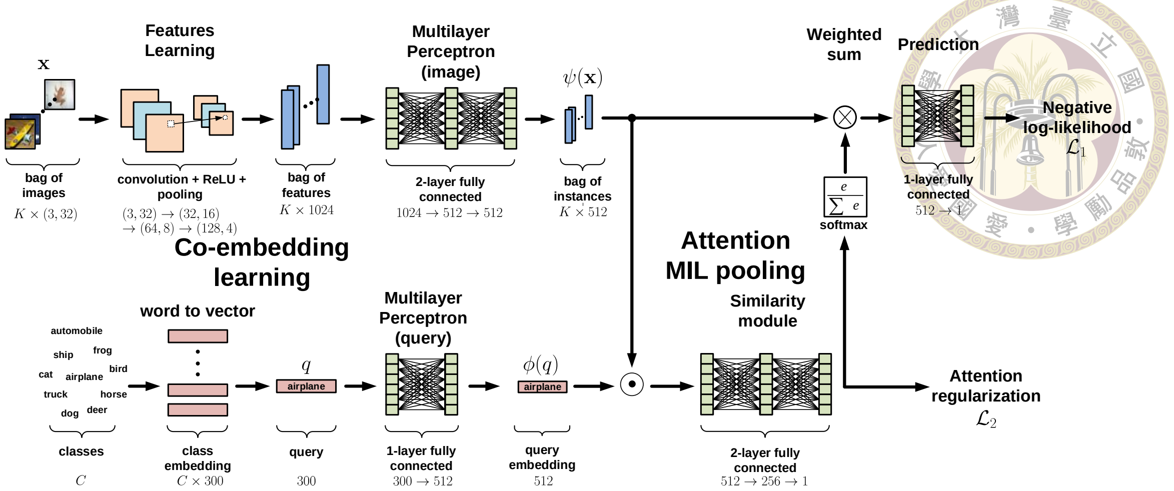


Figure 2.2: The proposed qMIL neural network architecture.

each $q \in Q'$, which enables focusing on learning to solve the classification task without imposing any presumed distribution on the data. In the experiments, we demonstrate that the inference performance of qMIL does not significantly vary with respect to the ratio between the numbers of positive and negative bags.

2.3.2 Query-adapted Attention Pooling

Although the number of instances in a qMIL bag could vary, we hereafter assume that all bags have K instances. After all, null instances can be introduced when needed. The unified bag size makes the batch training of learning the neural network, as shown in Figure 2.2, more convenient. Now consider an arbitrary training bag (X, Y, q), we use *word2vec* [90] to represent the query q as a 300-D feature vector and pass it through a two-layer MLP to obtain the query embedding $\phi(q) \in \mathbb{R}^d$. On the other hand, the image feature vector of each instance $\mathbf{x}$ is forward propagated through a three-layer MLP to yield its visual embedding which is denoted as $\psi(\mathbf{x}) \in \mathbb{R}^d$. The two mappings can be aligned to achieve query-visual co-embedding. To this end, we construct a network component $\mathcal{A}$ to function as a generalized compatibility measure for better exploring the co-embedding.

doi:10.6342/NTU202301574

Specifically, we have

$$\mathcal{A}(\phi(q), \psi(\mathbf{x})) = \sigma_2(\mathbf{w}^\intercal \sigma_1(V(\psi(\mathbf{x}) \odot \phi(q)))), \tag{2.2}$$

where $\mathbf{w} \in \mathbb{R}^{L\times 1}$ and $V \in \mathbb{R}^{L\times d}$ are network parameters, $\odot$ denotes the element-wise product, and σ_1, σ_2 are activation functions. When $L = d$ and linear activation functions in (2.2) are used, the generalized compatibility measure $\mathcal{A}$ simply reduces to taking inner product between $\psi(\mathbf{x})$ and $\phi(q)$ if both V and $\mathbf{w}$ are fixed as the identity versions.

It follows from (2.2) that we can use the compatibility measure $\mathcal{A}$ to compute the unnormalized attention $\alpha_k = \mathcal{A}(\phi(q), \psi(\mathbf{x}_k))$ for each instance $\mathbf{x}_k \in X$ to a given query q. Then the attention-weighted pooling is utilized to obtain the bag representation $\mathbf{z}$ for X, which adapts to the query q as follows:

$$\mathbf{z} = \sum_{k=1}^{K} \beta_k \, \mathbf{x}_k \quad \text{and} \quad \beta_k = \frac{\exp\{\alpha_k/\tau\}}{\sum_{j=1}^{K} \exp\{\alpha_j/\tau\}}, \tag{2.3}$$

where τ is the temperature parameter and β_k is the normalized attention of instance $\mathbf{x}_k \in X$ to q.

2.3.3 Loss Function and Regularization

For each training triplet $(X, Y, q) \in \mathcal{S}$, we now know how to derive the bag's feature vector $\mathbf{z}$ according to (2.3) and the corresponding unnormalized attention vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^\intercal$. To train the network to perform the (binary) classification task for predicting the bag label with respect to q, we need to define a proper loss function $\mathcal{L}$ to accomplish the qMIL learning. Specifically, we consider a label-dependent attention-regularized loss

doi:10.6342/NTU202301574

function:

$$\mathcal{L}(\mathcal{S}) = \mathcal{L}_1(\mathcal{S}) + \lambda\,\mathcal{L}_2(\mathcal{S}), \tag{2.4}$$

where λ is the weighting parameter, and the two losses for classifying each $(X, Y, q) \in \mathcal{S}$ are

$$\mathcal{L}_1(X) = Y \log p(X) + (1 - Y) \log (1 - p(X)), \tag{2.5}$$

$$\mathcal{L}_2(X) = Y \|\boldsymbol{\alpha}(X)\|_1 + (1 - Y)\{\mathrm{Var}(\boldsymbol{\alpha}(X))\}^{\frac{1}{2}}. \tag{2.6}$$

$\mathcal{L}_1$ in (2.5) is the cross-entropy loss and the attention regularization loss $\mathcal{L}_2$ in (2.6) plays a crucial role in the proposed qMIL formulation. Here we justify the form of the proposed regularization loss in (2.6) for the two possible cases.

- When $Y = 1$, the training bag X has a positive label to q and $\mathcal{L}_2 = \|\boldsymbol{\alpha}\|_1$. The ℓ_1-norm regularization effect is to find a *sparse* distribution of the instance attention. The preference is reasonable in the case where at least one instance is relevant to the query q and the sparse prior aims to distribute most attention to the relevant instances.

- When $Y = 0$, we have $\mathcal{L}_2 = \sqrt{\mathrm{Var}(\boldsymbol{\alpha})} = \|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|_2$. In this case all instances in the training bag X are irrelevant to the query q. The use of ℓ_2-norm thus encourages the attention to uniformly spread over all the instances.

doi:10.6342/NTU202301574

2.3.4 Zero-shot Classification via Queries

Thus far we have described how to leverage with the query mechanism to implicitly solve an MIML problem with a (triplet) training dataset, where each training bag is annotated only with a single binary label. We now explain how to apply a learned qMIL model to tackle the following zero-shot scenario. Suppose that in generating testing bags, we decide to consider instances from both the seen and unseen classes. Then, inquiring an arbitrary testing bag X with a query about an unseen class would result in zero-shot classification. We use an explicit example to depict the scenario. Let `car` be a seen class and `truck` an unseen class. A testing bag X includes at least one instance of `car` and all the other instances are not `truck`. A query about `truck` for X would most likely confuse the qMIL model and yields a positive return for the false existence of a `truck` instance. The confusion is caused by that `car` and `truck` are *similar* in the space induced by *word2vec*. Thus, to tackle the resulting zero-shot classification, we consider a two-stage procedure. In stage one, we iteratively perform queries of all the seen classes to identify *strong* positive instances, and exclude them from further considerations. In stage two, now without the severe distraction from the evident instances of seen classes, qMIL can then estimate a proper attention distribution and thus refine the bag representation for zero-shot classification. Further details are provided in the experimental results.

2.4 Experimental Results

We evaluate our method mainly on the MNIST-based dataset (MNIST-BAGS) [59] and CIFAR10-based dataset (CIFAR10-BAGS). Besides the pilot study on zero-shot classification, there are three groups of experimental results. The first set of experiments concerns a standard MIL problem where we compare qMIL with the deep MIL in [59]. In this setting, the total number of query class is just one. The second set of experiments is then extended to dealing with the MIML problem. As we have pointed out that despite using less-annotated training data, qMIL yields convincing results and shows effectiveness over the compared methods. The third set of experiments deals with a popular real-life application, action recognition. The proposed qMIL is applied to determine whether a given video clip contains a specific action to the query, where we have tested with a subset of Activity Net [41].

Learning with qMIL is advantageous, especially in creating training data. We just need to focus, in turn, on each particular category of interest, and mark whether the bag assumes the label or not. This can reduce human errors when annotating multiple labels and effectively reduce data noise. After all, in practical applications, we most likely care about only the categories we are interested in. Finally, given a novel query about an unseen class, the qMIL model is demonstrated to make reasonable predictions that are significantly better than random guesses.

2.4.1 Data Sampling

Table 2.1: Single query results on MNIST/CIFAR over ten runs of training/testing data sampling.

MNIST					**CIFAR10**				
Query	**GatedAttnDMIL**		**qMIL**		**Query**	**GatedAttnDMIL**		**qMIL**	
	accuracy	attention acc.	accuracy	attention acc.		accuracy	attention acc.	accuracy	attention acc.
0	95.4 ± 3.7	99.6 ± 1.2	**96.9 ± 2.2**	**99.6 ± 1.2**	plane	82.4 ± 1.7	82.7 ± 3.2	**89.9 ± 1.7**	**84.8 ± 1.5**
1	97.0 ± 4.1	99.6 ± 1.2	**98.0 ± 2.4**	**99.8 ± 0.6**	car	89.6 ± 1.8	95.7 ± 12.9	**90.7 ± 1.4**	**95.1 ± 1.4**
2	93.7 ± 3.6	99.6 ± 1.2	**95.7 ± 2.7**	**99.6 ± 1.2**	bird	72.4 ± 2.6	60.0 ± 22.7	**73.6 ± 2.4**	**69.7 ± 9.0**
3	93.2 ± 3.6	99.8 ± 0.6	**96.0 ± 2.3**	**100.0 ± 0.0**	cat	75.4 ± 3.0	54.1 ± 12.8	**76.3 ± 2.9**	**59.7 ± 10.3**
4	94.7 ± 2.5	99.2 ± 0.9	**96.5 ± 1.3**	**99.4 ± 0.9**	deer	71.4 ± 3.1	66.6 ± 5.9	**73.8 ± 2.4**	**67.6 ± 5.6**
5	94.0 ± 5.8	100.0 ± 0.0	**97.0 ± 2.2**	**100.0 ± 0.0**	dog	74.1 ± 2.3	62.2 ± 20.0	**74.3 ± 1.8**	**69.8 ± 6.9**
6	94.7 ± 4.1	99.00 ± 1.3	**97.1 ± 2.4**	**99.2 ± 1.3**	frog	82.2 ± 3.0	87.8 ± 1.9	**82.6 ± 2.4**	**88.4 ± 2.5**
7	94.2 ± 3.1	100.0 ± 0.0	**96.1 ± 1.6**	**100.0 ± 0.0**	horse	82.7 ± 2.9	77.8 ± 19.6	**82.8 ± 1.9**	**82.8 ± 7.9**
8	89.3 ± 6.9	99.20 ± 0.9	**92.1 ± 5.9**	**99.6 ± 0.8**	ship	87.8 ± 2.5	89.1 ± 1.8	**88.4 ± 1.9**	**89.8 ± 1.4**
9	91.3 ± 3.6	98.20 ± 1.9	**92.9 ± 3.1**	**98.2 ± 1.9**	truck	85.5 ± 1.8	90.4 ± 2.6	**85.9 ± 1.6**	**91.6 ± 2.4**

We follow the similar data sampling method in [59] to create the MNIST-BAGS MIL dataset from MNIST [72] and analogously from CIFAR10 [70]. The standard MIL problem with one single query proceeds as follows. In MNIST or in CIFAR10, each of the ten categories will be chosen in turn as the one of interest, and the remaining are treated as background/noise. The instances in each bag are randomly included, and the number of instances is an integer arbitrarily sampled from the normal distribution $\mathcal{N}(10, 2)$. To speed up the training process, after data sampling and when necessary, zero images are generated to ensure that each bag has exactly K image instances. We next turn to the MIML scenario. For each image we now have multiple labels but do not indicate the specific label of each instance. (We have described how we construct such training data in establishing the qMIL problem.) There are two kinds of inference tasks for MIML. One is the classical MIML problem, and the other is ours, which is query-driven. For fair comparisons, we adopt the MIML Scene dataset [149] as the benchmark and report 10-fold cross-validation results. Note that the numbers of positive bags and negative bags to a query in the MIML Scene dataset is unbalanced. The ratio between positive and negative bags is about $3 : 1$. The last experiment is about action recognition. In this case, a video clip can be thought of as a bag, while each frame is an instance.

2.4.2 Training and Inference

In the experiments of MNIST-MIL and CIFAR10-MIL, the hyperparameters can be kept the same. This implies that the proposed attention regularization in (2.6) is general and not data-sensitive. In MNIST, our CNN model conforms to the LeNet architecture [72] which comprises two conv layers for MNIST, and three conv layers for CIFAR10. The learning rate is 10^{-4} at initialization and the optimization method is Adam [68]. The

doi:10.6342/NTU202301574

weight decay is 10^{-5}, while λ in (2.4) is 10^{-4} for all the experiments. We fix τ in (2.3) as 0.5. σ_1 and σ_2 in (2.2) are tanh and linear mapping. For single query, the results are reported with the mean and standard deviation from ten different runs of random data sampling. For multiple queries, five random runs are instead evaluated for the sake of efficiency.

Metrics In both our model and the compared method, the output of the bag-level prediction to the MIL problem is a probability p. Thus to compute the accuracy of the bag-level prediction, the decision threshold is set as $p > 0.5$ with label $Y = 1$ and $p \leq 0.5$ with label $Y = 0$. Consider now an arbitrary bag $X = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$. In both MNIST-MIL and CIFAR10-MIL, we indeed have access to the class label of each instance, *i.e.*, $(y_1, \ldots, y_K)$. The instance-level ground truth can be used to evaluate the accuracy of the predicted instance attention in each bag. We name the resulting quantity as the instance-level accuracy. The attention accuracy is evaluated as follows. Each time we predict the bag label as $Y = 1$ for a triplet (X, Y, q), we check the instance label y_{k^*} of the most *manifest* instance $\mathbf{x}_{k^*}$ where $k^* = \arg\max_k \beta_k$ from (2.3). If $y_{k^*} = 1$, then we have correct instance attention.

2.4.3 Standard MIL

In standard MIL experiments, for each single query to a specific class label we first sample 500 training bags, including 250 positive and 250 negative bags from MNIST. Analogously, another 1000 bags (500 "+" & 500 "-") are also generated for testing. The setting for CIFAR10 is the same. We compare our method with the state-of-the-art deep MIL model, denoted as GatedAttnDMIL [59] and report the results in Table 2.1. The proposed qMIL achieves better performances in both bag-level accuracy and instance-level attention accuracy. In Table 2.2, we report the performance versus different numbers of training bags for the CIFAR10 dataset. The results are on 500 testing bags. To achieve bag-level predictions of high confidence, qMIL needs 5000 training bags (2500 "+" & 2500 "-") for a single query. Our method also achieves better results in both accuracy metrics.

Table 2.2: Single query on CIFAR10. N: total # of training bags. (**acc**: accuracy, **att**: attention)

	N bags	100	500	1000	2000	5000
GatedAttnDMIL [59]	**acc**	55.1 ± 8.6	62.1 ± 6.7	61.2 ± 6.2	70.6 ± 4.3	82.4 ± 1.7
qMIL		**56.3 ± 4.5**	**62.8 ± 3.9**	**63.4 ± 4.1**	**71.8 ± 2.8**	**89.9 ± 1.7**
GatedAttnDMIL [59]	**att acc**	49.2 ± 20.1	58.2 ± 13.4	66.7 ± 8.3	76.8 ± 4.4	82.7 ± 3.2
qMIL		**55.3 ± 11.3**	**61.2 ± 8.1**	**67.2 ± 5.2**	**78.2 ± 2.1**	**84.8 ± 1.5**

Table 2.3: Performance with respect to # of queries on CIFAR10. The notation qMIL$^-$ denotes that the regularization loss $\mathcal{L}_2$ in (2.6) is not used in training. For each query, we sample 5000 training bags.

	# queries	1	3	5	7	10
qMIL$^-$	**acc**	82.4 ± 1.7	81.22 ± 1.8	71.23 ± 3.4	65.66 ± 4.6	78.33 ± 2.3
qMIL		**89.9 ± 1.7**	**81.77 ± 1.4**	**79.45 ± 2.7**	**82.09 ± 2.1**	**86.14 ± 1.3**
qMIL$^-$	**att acc**	82.7 ± 3.2	65.52 ± 9.9	53.21 ± 10.37	45.66 ± 20.3	70.64 ± 5.4
qMIL		**84.8 ± 1.5**	**87.22 ± 1.1**	**83.30 ± 1.3**	**86.01 ± 1.2**	**89.18 ± 1.0**

Table 2.4: 10-fold cross validation on MIML Scene dataset.

	accuracy
deep MIML [42]	89.45 ± 1.22
qMIL	**90.20 ± 0.96**

2.4.4 MIML

In the MIML problem, we have two ways of testing. One is to make the testing data the same form by our labeling scheme on training data, and the other is the standard MIML task that a bag of instances has several labels to be predicted. Table 2.3 shows the performances with respect to the numbers of query classes. When excluding the use of $\mathcal{L}_2$ in (2.6) (shown as qMIL^{-} in Table 2.3), we have trained with many different hyperparameters and report the best results. It can be observed that with the attention regularization term, $\mathcal{L}_2$, learning the model becomes easier and more stable during training. (Further details about the regularization effect with $\mathcal{L}_2$ can be found in the supplementary material.)

We have also tested according to the standard MIML task by evaluating the model with each query for a given bag. Table 2.4 and Figure 2.3 include the results of the MIML task on the MIML Scene dataset and the comparison with the deep MIML [42] which is shown to outperform MIML SVM, MIML KNN, MIML RBF and MIML Boost [150]. We adopt a pre-trained ResNet50 [53] and re-implement the deep MIML by following the details described in the paper. The resulting deep MIML architecture consists of the pre-trained ResNet50, 2D sub-concept layer for multiple instances, and max pooling twice to yield the multi-label prediction. It is trained from scratch and learned end-to-end.

To better capture the effect of attention regularization, we investigate how the attention weights of a bag vary with respect to different queries of a class label. Table 2.5 shows the bag-level prediction of probability p and the attention weight distribution according to each query at testing.

Table 2.5: Given a testing bag (13 instances), the instance attention weights vary w.r.t. different queries.

														p
plane	0.04	0.07	0.02	0.01	0.02	0.01	0.03	0.05	0.01	0.12	**0.55**	0.05	0.01	0.99
car	0.00	0.00	0.01	**0.81**	0.01	0.00	0.01	0.01	0.00	0.11	0.01	0.01	0.00	0.99
bird	0.03	0.07	0.03	0.01	0.02	0.03	0.02	**0.54**	0.03	0.01	0.02	0.03	0.15	0.98
cat	0.07	**0.24**	0.19	0.01	0.05	0.11	0.02	0.04	0.08	0.01	0.02	0.02	0.16	0.96
deer	0.15	0.09	0.08	0.01	0.03	0.04	**0.33**	0.07	0.06	0.03	0.03	0.03	0.05	0.01
dog	0.04	0.09	0.29	0.01	0.01	0.08	0.01	0.02	**0.37**	0.01	0.01	0.01	0.06	0.99
frog	0.07	0.08	0.11	0.11	0.04	**0.14**	0.05	0.06	0.07	0.07	0.03	0.04	0.12	0.01
horse	0.06	0.03	0.05	0.01	0.01	0.02	**0.68**	0.02	0.06	0.02	0.01	0.02	0.02	0.96
ship	0.01	0.02	0.00	0.01	**0.53**	0.01	0.00	0.01	0.00	0.01	0.04	0.36	0.01	0.99
truck	0.05	0.05	0.07	0.13	0.05	0.03	0.08	0.05	0.04	**0.26**	0.08	0.09	0.02	0.01

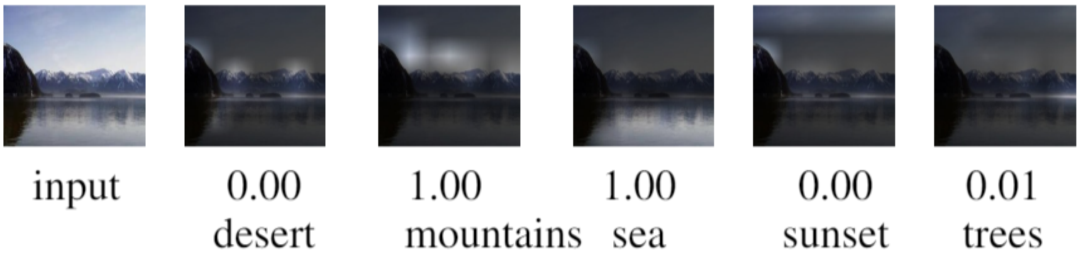


Figure 2.3: From column 2 to column 6: Each includes an attention heatmap and its bag-level probability, while the input image is shown in the first column.

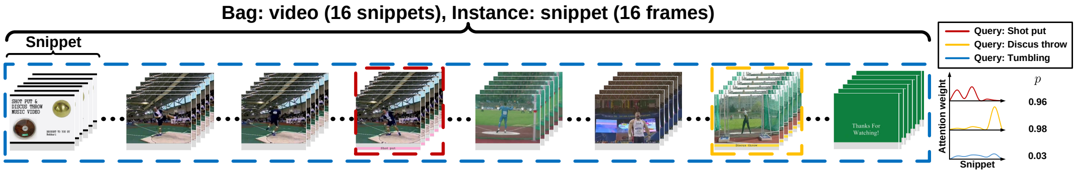


Figure 2.4: qMIL for action recognition. Each video clip comprises 16 snippets. Three different queries are chosen for testing. p is the bag-level probability prediction for supporting a query.

2.4.5 MIML for Video Applications

The proposed qMIL can be readily applied to deal with video-related applications. Particularly, we explore the problem involving the Activity Net [41] and convert the problem into our formulation described in the proposed qMIL. Following [124], each snippet comprises 16 consecutive frames, and a video clip can thus be represented as a sequence of snippets. Under such a setting, a video clip is a bag and each snippet is an instance, while its bag label is defined with respect to the query. In our experiment, we consider those video clips related to the following three action classes, namely, shot put, discus throw, and tumbling. Figure 2.4 shows the result of the proposed qMIL approach to action recognition.

Table 2.6: Zero-shot testing accuracy with seven seen classes and three unseen classes. Test data are sampled from seen+unseen (ten classes) or from unseen (seven classes). IQP denotes the iterative query process.

	horse	ship	truck	total
seen & unseen	58.80	62.20	59.60	60.20
seen & unseen (IQP)	57.80	64.20	63.00	61.67
unseen	66.66	72.00	66.33	68.33

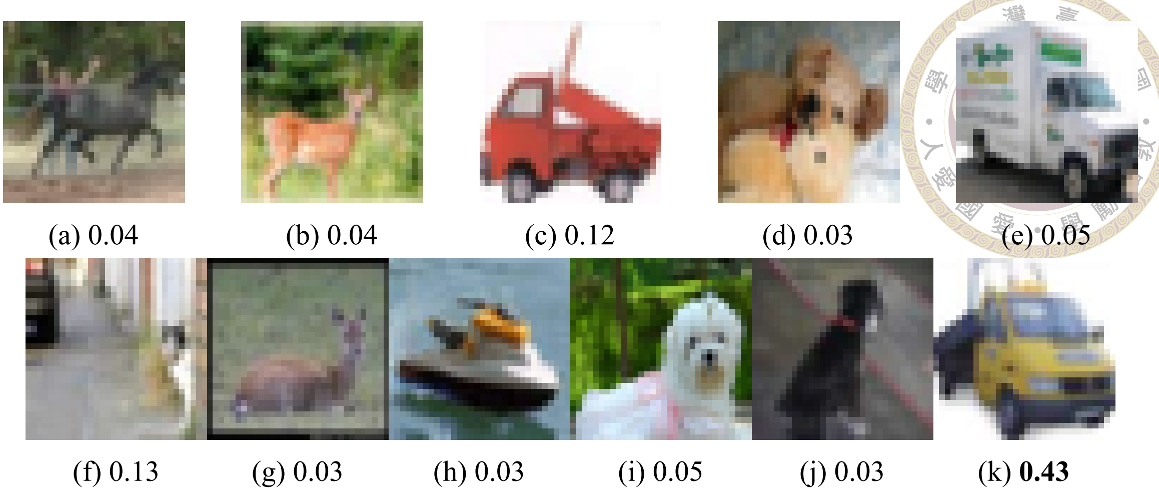
2.4.6 Zero-shot Scenarios

We also test qMIL for zero-shot classification on CIFAR10. Specifically, we train the proposed qMIL with seven seen classes and test on the remaining three unseen classes. Each bag in the training data is randomly composed of instances from the seven seen classes, and the testing data are formed based on two kinds of sampling methods. The fist scenario is that the testing bags are sampled only from the three unseen classes, and the other is sampled from all of the ten classes (seen & unseen). For the latter case, the learned qMIL is carried out with the help of iterative queries as described in **Zero-shot Classification via Queries**. The experimental results of zero-shot classification are shown in Table 2.6 and Figure 2.5. We remark that the zero-shot scenario is essentially different from the conventional formulation. Therefore, it is not appropriate to directly compare it with other specific zero-shot learning techniques, which are cast in a very different way. The application demonstrates that the advantages and flexibility of the proposed qMIL formulation over conventional MIL frameworks.

2.5 Conclusions

From the viewpoint of problem reduction, the proposed qMIL framework indeed can be considered as decomposing MIML into a series of query-driven MIL sub-tasks. The

doi:10.6342/NTU202301574



(a) 0.04 (b) 0.04 (c) 0.12 (d) 0.03 (e) 0.05

(f) 0.13 (g) 0.03 (h) 0.03 (i) 0.05 (j) 0.03 (k) **0.43**

Figure 2.5: The “truck” class is not in the training data. Given the query of unseen “truck”, qMIL with IQP will pay more attention to the “truck” image in a bag and the bag-level probability is $p = 0.96$. The numbers are the attention weights.

reduction yields advantages in two different aspects. First, annotating each training bag requires a single binary label, rather than a binary label vector. It also has the flexibility to expand the training dataset to include data of new classes without the need to modify the labeling information in the existing training bags. Second, the reduced sub-tasks can all be cast as query-driven MIL, and thus can be addressed in a unified neural network architecture. By focusing on solving the reduced MIML problem, we are able to establish a query-visual co-embedding with the label-adapted regularization in (2.6) and represent a given MIL bag with a proper representation for more effective classification. Our future work will focus on improving the qMIL attention mechanism and expanding its application aspect in image/video processing.

Chapter 3 Learning with Unlabeled Data Distributions

3.1 Introduction

As a fundamental task in machine learning, representation learning aims to extract useful information from the raw data for the downstream tasks. It has been regarded as a long-acting goal over the past decades. Recent progress on representation learning has achieved a significant milestone over self-supervised learning (SSL), facilitating feature learning with its competence in exploiting massive raw data without any annotated supervision. In the early stage of SSL, representation learning has focused on exploiting pretext tasks, which are addressed by generating pseudo-labels to the unlabeled data through different transformations, such as solving jigsaw puzzles [92], colorization [143] and rotation prediction [45]. Though these approaches succeed in computer vision, there is a large gap between these methods and supervised learning. Recently, there has been a significant advancement in using contrastive learning [19, 51, 118, 120, 133] for self-supervised pre-training, which significantly closes the gap between the SSL method and supervised learning. Contrastive SSL methods, e.g., SimCLR [19], in general, try to pull different views of the same instance close and push different instances far apart in the representation space.

doi:10.6342/NTU202301574

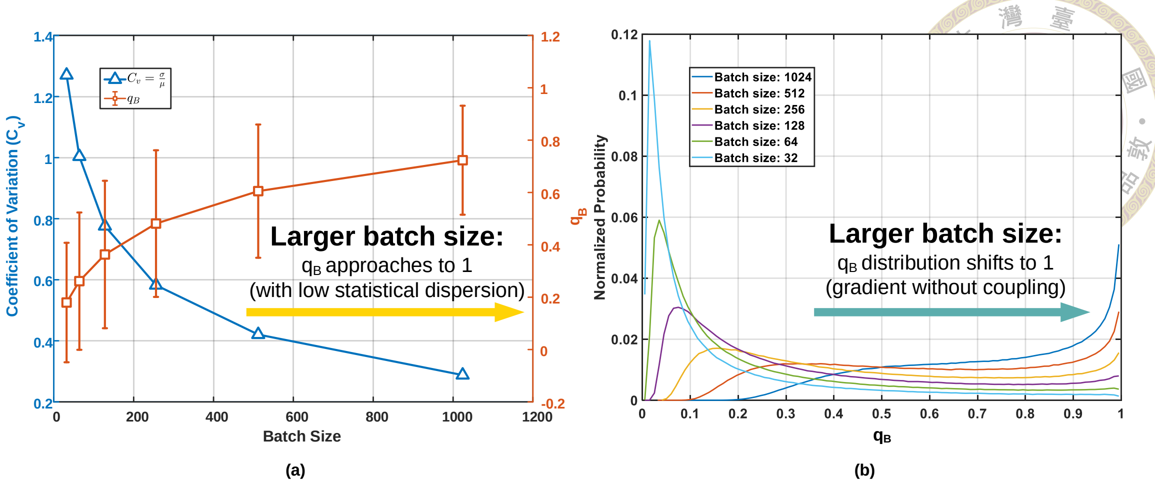


Figure 3.1: An overview of the batch size issue is that general contrastive approaches need large batch sizes to perform better: (a) shows the NPC multiplier q_B in different batch sizes. As the batch size gradually increases, the q_B will approach to 1 with a small coefficient of variation ($C_v = \sigma/\mu$); and (b) illustrates the distribution of q_B with various batch sizes and indicates that the mode value of q_B will shift towards 1 when the batch size increases. Note that the σ and μ are the standard deviation and mean of q_B, respectively. The coefficient of variation, C_v, measures the dispersion of a frequency distribution.

Despite the evident progress of the state-of-the-art contrastive SSL methods, there have been facing several challenges into future development in this direction, including 1) The SOTA models, *e.g.*, [51] may require specific structures such as the momentum encoder and large memory queues, which may complicate the underlying representation learning. 2) The contrastive SSL models, *e.g.*, [19] often depend on large batch size and huge epoch numbers to achieve competitive performance, posing a computational challenge for academia to explore this direction. 3) They tend to be sensitive to hyperparameters and optimizers, introducing additional difficulty reproducing the results on various benchmarks.

Through the analysis of the widely adopted InfoNCE loss in contrastive learning, we identified a negative-positive-coupling (NPC) multiplier q_B in the gradient as shown in Proposition 1. The NPC multiplier modulates the gradient of each sample, and it reduces the learning efficiency due to easy SSL classification tasks: 1) when a positive sample is very close to the anchor; 2) when negative samples are far away from the anchor; and 3) when there is only a small number of negative samples (i.e., a small batch size). A less-informative (nearby) positive view would reduce the gradient from a batch of informative negative samples or vice versa. Such a coupling exacerbates when smaller batch sizes are used.

Meanwhile, we also investigate the relationship between q_B and batch size through the baseline, SimCLR. As can be seen in Figure 3.1, the distribution of q_B has a strong positive correlation with the batch size. Figure 3.1(a) shows that when batch size gradually increases, q_B not only approaches 1 but also reduces the coefficient of variation C_v. The distribution with larger C_v has low statistical dispersion and vice versa. Figure 3.1(b) indicates that the mode value of q_B will also shift from 0 to 1 when the batch size becomes larger. Hence, it is reasonable to fix the value of q_B, alleviating the influence of batch size.

By removing the coupling term from the Info-NCE loss, we reach a new formulation, the *decoupled contrastive learning* (DCL). The new objective function significantly improves the training efficiency with less sensitivity to sub-optimal hyper-parameters requires neither large batches, momentum encoding, or large epochs to achieve competitive performance on various benchmarks. The main contributions of the proposed DCL can be characterized as follows:

1) We provide both theoretical analysis and empirical evidence to show the NPC effect in the InfoNCE-based contrastive learning;

2) We introduce DCL objective, which casts off the NPC coupling phenomenon, significantly improves the training efficiency, and it is less sensitive to sub-optimal hyper-parameters;

3) Extensive experiments are provided to show the effectiveness of the proposed method

doi:10.6342/NTU202301574

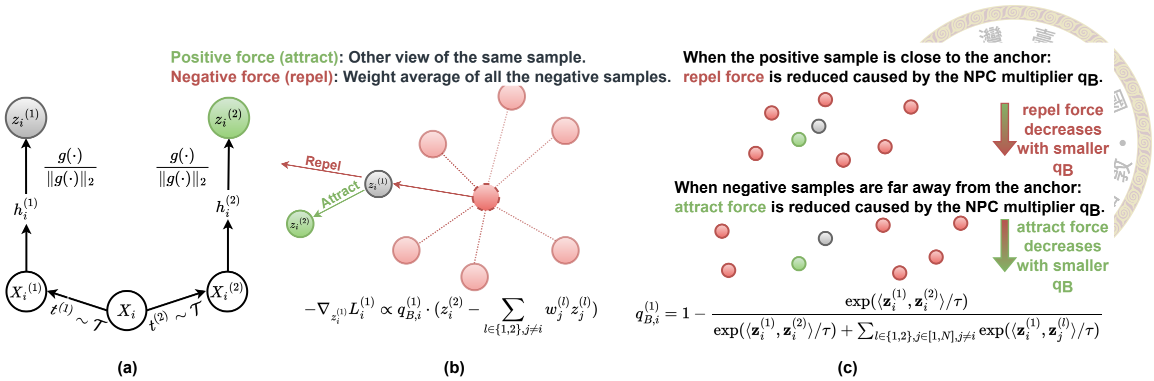

$z_i^{(1)}$
$z_i^{(2)}$
$\frac{g(\cdot)}{\|g(\cdot)\|_2}$
$\frac{g(\cdot)}{\|g(\cdot)\|_2}$
$h_i^{(1)}$
$h_i^{(2)}$
$X_i^{(1)}$
X_i
$X_i^{(2)}$
$t^{(1)} \sim \mathcal{T}$
$t^{(2)} \sim \mathcal{T}$
(a)
Positive force (attract): Other view of the same sample.
Negative force (repel): Weight average of all the negative samples.
Repel
Attract
$z_i^{(1)}$
$z_i^{(2)}$
$-\nabla_{z_i^{(1)}} L_i^{(1)} \propto q_{B,i}^{(1)} \cdot (z_i^{(2)} - \sum_{l\in\{1,2\},j\neq i} w_j^{(l)} z_j^{(l)})$
(b)
When the positive sample is close to the anchor:
repel force is reduced caused by the NPC multiplier qB.
repel force decreases with smaller qB
When negative samples are far away from the anchor:
attract force is reduced caused by the NPC multiplier qB.
attract force decreases with smaller qB
$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle/\tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle/\tau) + \sum_{l\in\{1,2\},j\in[1,N],j\neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)} \rangle/\tau)}$
(c)

Figure 3.2: Contrastive learning and negative-positive coupling (NPC). (a) In SimCLR, each sample $\mathbf{x}_i$ has two augmented views $\{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$. They are encoded by the same encoder f and further projected to $\{\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\}$ by a normalized MLP. (b) According to Equation 3.4. For the view $\mathbf{x}_i^{(1)}$, the cross-entropy loss $L_i^{(1)}$ leads to a positive force $\mathbf{z}_i^{(2)}$, which comes from the other view $\mathbf{x}_i^{(2)}$ of $\mathbf{x}$ and a negative force, which is a weighted average of all the negative samples, i.e. $\{\mathbf{z}_j^{(l)} | l \in \{1,2\}, j \neq i\}$. However, the gradient $-\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)}$ is proportional to the NPC multiplier. (c) We show two cases when the NPC term affects learning efficiency. The positive sample is close to the anchor and less informative on the top. However, the gradient from the negative samples is also reduced. On the bottom, when the negative samples are far away and less informative, the learning rate from the positive sample is mistakenly reduced. In general, the NPC multiplier from the InfoNCE loss makes the SSL task simpler to solve, leading to reduced learning efficiency.

that DCL achieves competitive performance **without** large batch sizes, large training epochs, momentum encoding, or additional tricks such as stop-gradient and multi-cropping, etc. This leads to a plug-and-play improvement to the widely adopted InfoNCE-based contrastive learning;

4) We show that DCL can be easily combined with the SOTA contrastive methods, e.g. NNCLR [39], to achieve further improvements.

3.2 Related Work

Contrastive Learning. Contrastive learning (CL) constructs positive and negative sample pairs to extract information from the data itself. In CL, each anchor image in a batch has only one positive sample to construct a positive sample pair [19, 49, 51]. CPC [120] predicts the future output of sequential data by using current output as prior knowledge,

doi:10.6342/NTU202301574

which can improve the feature representing the ability of the model. Instance discrimination [133] proposes a non-parametric cross-entropy loss to optimize the model at the instance level. Inv. spread [136] makes use of data augmentation invariant and the spread-out property of instance to learn features. MoCo [51] proposes a dictionary to maintain a negative sample set, thus increasing the number of negative sample pairs. Different from the aforementioned self-supervised CL approaches, [66] proposes a supervised CL that considers all the same categories as positive pairs to increase the utility of images.

Collapsing Issue on the Number of Negatives. In CL, the objective is to maximize the mutual information between the positive pairs. However, to avoid the "*collapsing output*", vast quantities of negative samples are needed so that the learning objectives obtain the maximum similarity and have the minimum similarity with negative samples. For instance, in SimCLR [19], training requires many negative samples, leading to a large batch size (i.e., 4096). Furthermore, to optimize such a huge batch, a specially designed optimizer LARS [138] is used. Similarly, MoCo [51] needs a vast queue (i.e., 65536) to achieve competitive performance. BYOL [46] does not collapse output without using any negative samples by considering all the images are positive and to maximize the similarity of "projection" and "prediction " features. On the other hand, SimSiam [21] leverages the Siamese network to introduce inductive biases for modeling invariance. With the small batch size (i.e., 256), SimSiam is a rival to BYOL (i.e., 4096). Unlike both approaches that achieved their success through empirical studies, we tackle from a theoretical perspective, proving that an intertwined multiplier q_B of positive and negative is the main issue to contrastive learning.

Batch Size Sensitivity on InfoNCE. Several works of literature focus on batch size sensitivity concerning the InfoNCE objective function. [119] proposes an objective based on

doi:10.6342/NTU202301574

relative predictive coding that maintains the balance between training stability and batch size sensitivity. [54] follows the [4] and extends the idea between the local and global features. [93] proposes a Wasserstein distance to prevent the encoder from learning any other differences between unpaired samples. [62] and [101] learn better representation by sampling hard negatives, particularly for small batches. Other recent works [40, 151] aim to mitigate the issue of small batch size in InfoNCE loss. Although the basic principle of recent works and DCL is derived from InfoNCE objective function, we provide a novel perspective to support the decoupling between positive and negative terms in InfoNCE loss is essential. Simply removing the term from the denominator pre-training to positive pairs can drastically improve the performance and keep the objective function invariant to batch size sensitivity.

3.3 Decouple Negative and Positive Samples in Contrastive Learning

We choose to start from SimCLR because of its conceptual simplicity. Given a batch of N samples (e.g. images), $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, let $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}$ be two augmented views of the sample x_i and B be the set of all of the augmented views in the batch, i.e. $B = \{\mathbf{x}_i^{(k)} | k \in \{1, 2\}, i \in [\![1, N]\!]\}$. As shown by Figure 3.2(a), each of the views $\mathbf{x}_i^{(k)}$ is sent into the same encoder network f and the output $\mathbf{h}_i^{(k)} = f(\mathbf{x}_i^{(k)})$ is then projected by a normalized MLP projector that $\mathbf{z}_i^{(k)} = g(\mathbf{h}_i^{(k)})/\|g(\mathbf{h}_i^{(k)})\|$. For each augmented view $\mathbf{x}_i^{(k)}$, SimCLR solves a classification problem by using the rest of the views in B as targets, and assigns the only positive label to $\mathbf{x}_i^{(u)}$, where $u \neq k$. So SimCLR creates a cross-entropy loss function $L_i^{(k)}$

doi:10.6342/NTU202301574

for each view $\mathbf{x}_i^{(k)}$, and the overall loss function is $L = \sum_{k\in\{1,2\}, i\in[\![1,N]\!]} L_i^{(k)}$.

$$L_i^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau) + U_{i,k}}, \tag{3.1}$$

where

$$U_{i,k} = \sum_{l\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)}\rangle/\tau) \tag{3.2}$$

means the summation of negative terms for the view k of the sample i.

Proposition 1. : *There exists a negative-positive coupling (NPC) multiplier $q_{B,i}^{(1)}$ in the gradient of $L_i^{(1)}$:*

$$\begin{cases} -\nabla_{\mathbf{z}_i^{(1)}} L_i^{(1)} = \\ \frac{q_{B,i}^{(1)}}{\tau}\left(\mathbf{z}_i^{(2)} - \sum_{l\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \frac{\exp\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)}\rangle/\tau}{U_{i,1}} \cdot \mathbf{z}_j^{(l)}\right) \\ -\nabla_{\mathbf{z}_i^{(2)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau} \cdot \mathbf{z}_i^{(1)} \\ -\nabla_{\mathbf{z}_j^{(l)}} L_i^{(1)} = -\frac{q_{B,i}^{(1)}}{\tau} \frac{\exp\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)}\rangle/\tau}{U_{i,1}} \cdot \mathbf{z}_i^{(1)} \end{cases} \tag{3.3}$$

where the NPC multiplier $q_{B,i}^{(1)}$ is:

$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\rangle/\tau) + U_{i,1}} \tag{3.4}$$

and $U_{i,1} = \sum_{l\in\{1,2\}, j\in[\![1,N]\!], j\neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(l)}\rangle/\tau)$. Due to the symmetry, a similar NPC multiplier $q_{B,i}^{(k)}$ exists in the gradient of $L_i^{(k)}, k \in \{1,2\}, i \in [\![1,N]\!]$.

As we can see, all of the partial gradients in Equation 3.3 are modified by the common NPC multiplier $q_{B,i}^{(k)}$ in Equation 3.4. Equation 3.4 makes intuitive sense: when the SSL classification task is easy, the gradient would be reduced by the NPC term. However, the positive samples and negative samples are strongly coupled. When the negative samples are far away and less informative (easy negatives), the gradient from an informative, positive sample would be reduced by the NPC multiplier $q_{B,i}^{(1)}$. On the other hand, when the positive sample is close (easy positive) and less informative, the gradient from a batch of informative negative samples would also be reduced by the NPC multiplier. When the batch size is smaller, the SSL classification problem can be significantly simpler to solve. As a result, the learning efficiency can be significantly reduced with a small batch size setting.

Figure 3.1(b) shows the NPC multiplier q_B distribution shift w.r.t. different batch sizes for a pre-trained SimCLR baseline model. While all of the shown distributions have prominent fluctuation, the smaller batch size makes q_B cluster towards 0, while the larger batch size pushes the distribution towards $\delta(1)$. Figure 3.1(a) shows the averaged NPC multiplier $\langle q_B \rangle$ changes w.r.t. the batch size and the relative fluctuation. The small batch sizes introduce significant NPC fluctuation. Based on this observation, we propose to remove the NPC multipliers from the gradients, which corresponds to the case $q_{B,N\to\infty}$. This leads to the decoupled contrastive learning formulation. [127] also proposes an alignment & uniformity loss which does not have the NPC. However, a similar analysis introduces negative-negative coupling from different positive samples. In other words, [127] considers all the negative samples in the batch together, which may cause the gradient to be dominated by a specific negative pair. In Appendix 5, we provide a thorough discussion and demonstrate the advantage of DCL loss against [127].

Proposition 2. the DCL Loss: *Removing the positive pair from the denominator of Equation 3.1 leads to a decoupled contrastive learning loss. If we remove the NPC multiplier $q_{B,i}^{(k)}$ from Equation 3.3, we reach a decoupled contrastive learning loss $L_{DC} = \sum_{k\in\{1,2\},i\in[\![1,N]\!]} L_{DC,i}^{(k)}$, where $L_{DC,i}^{(k)}$ is:*

$$L_{DC,i}^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle/\tau)}{\cancel{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle/\tau)} + U_{i,k}} \tag{3.5}$$

$$= -\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle/\tau + \log U_{i,k} \tag{3.6}$$

The proofs of Proposition 1 and 2 are given in Appendix. Further, we can generalize the loss function L_{DC} to L_{DCW} by introducing a weighting function for the positive pairs i.e. $L_{DCW} = \sum_{k\in\{1,2\}, i\in[\![1,N]\!]} L^{(i,k)}_{DCW,i}$.

$$L^{(k)}_{DCW,i} = -w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \log U_{i,k} \tag{3.7}$$

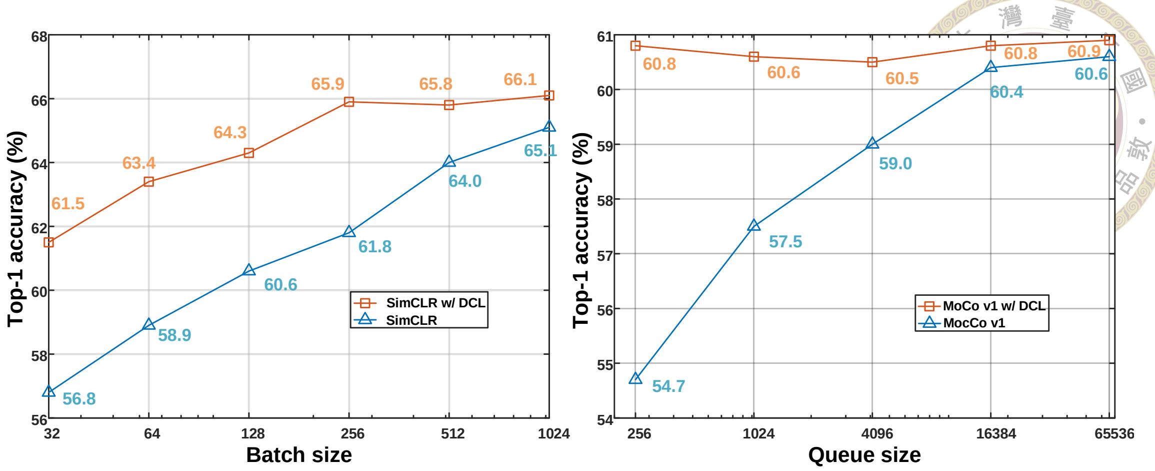
where we can intuitively choose w to be a negative von Mises-Fisher weighting function that $w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}) = 2 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \sigma)}{\mathrm{E}_i\left[\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \sigma)\right]}$ and $\mathrm{E}\,[w] = 1$. L_{DC} is a special case of L_{DCW} and we can see that $\lim_{\sigma\to\infty} L_{DCW} = L_{DC}$. The intuition behind $w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})$ is that there is more learning signal when a positive pair of samples are far from each other, and $E\left[w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}) \langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle\right] \approx E\left[\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle\right]$. Other similar weight functions also provide similar results. In general, we find such a weighting function, which gives a larger weight to the hard positives tend to increase the representation quality.

3.4 Experiments

This section empirically evaluates the proposed decoupled contrastive learning (DCL) and compares it to general contrastive learning methods. We summarize the experiments and analysis as the following: (1) the proposed work significantly outperforms the general InfoNCE-based contrastive learning on both large-scale and small-scale vision benchmarks; (2) we show that the enhanced version of DCL, DCLW, could further improve the representation quality; and (3) we further analyze DCL with ablation studies on ImageNet-1K, hyperparameters, and few learning epochs, which shows fast convergence of the proposed DCL. Note that all the experiments are conducted with 8 Nvidia V100 GPUs on a single machine.

3.4.1 Implementation Details

ImageNet. For a fair comparison on ImageNet data, we implement the proposed decoupled structure, DCL, by following SimCLR [19] with ResNet-50 [53] as the encoder backbone and use cosine annealing schedule with SGD optimizer. We set the temperature τ to 0.1 and the latent vector dimension to 128. Following the OpenSelfSup benchmark [140],



(a)

(b)

Figure 3.3: Comparisons on ImageNet-1K with/without DCL under different numbers of (a): batch sizes for SimCLR and (b): queues for MoCo. Without DCL, the top-1 accuracy significantly drops when batch size (SimCLR) or queues (MoCo) becomes very small. Note that the temperature τ is 0.1 for SimCLR and 0.07 for MoCo in the comparison.

we evaluate the pre-trained models by training a linear classifier with frozen learned embedding on ImageNet data. We further consider evaluating DCL on ImageNet-100, a selected subset of 100 classes of ImageNet-1K. Note that all models on ImageNet are trained for 200 epochs.

CIFAR and STL10. For CIFAR10, CIFAR100, and STL10, ResNet-18 [53] is used as the encoder architecture. Following the small-scale benchmark [130], we set the temperature τ to 0.07. All models are trained for 200 epochs with SGD optimizer, a base $lr = 0.03 * batchsize/256$, and evaluated by k nearest neighbor (kNN) classifier. Note that on STL10, we include both the $train$ and $unlabeled$ set for model pre-training. We further use ResNet-50 as a stronger backbone by following the implementation [100], using the same backbone and hyperparameters.

3.4.2 Experiments and Analysis

DCL on ImageNet. This section illustrates the effect of DCL against InfoNCE-based approaches under different batch sizes and queues. The initial setup is to have 1024 batch size (SimCLR) and 65536 queues (MoCo [51]) and gradually reduce the batch size (SimCLR) and queue (MoCo) to show the corresponding top-1 accuracy by linear evaluation.

Table 3.1: Comparisons with/without DCL under different batch sizes from 32 to 512. Results show the effectiveness of DCL on five widely used benchmarks. The performance of DCL keeps steadier than the SimCLR baseline while the batch size is varied.

Batch Size	32	64	128	256	512
Dataset	ImageNet-1K (kNN / Linear)				
Baseline (ResNet-50)	40.2/56.8	42.9/58.9	45.1/60.6	46.3/61.8	49.4/64.0
w/ DCL (ResNet-50)	**43.7/61.5**	**46.3/63.4**	**48.5/64.3**	**49.8/65.9**	**50.1/65.8**
Dataset	ImageNet-100 (kNN / Linear)				
Baseline (ResNet-50)	67.8/74.2	71.9/77.6	73.2/79.3	74.6/80.7	75.4/81.3
w/ DCL (ResNet-50)	**74.9/80.8**	**76.3/82.0**	**76.5/81.9**	**76.9/83.1**	**76.8/82.8**
Dataset	CIFAR-10 (kNN / Linear)				
Baseline (ResNet-18)	78.9/79.8	80.4/81.3	81.1/82.8	81.4/83.0	81.3/83.3
w/ DCL (ResNet-18)	**83.7/85.1**	**84.4/85.9**	**84.4/85.7**	**84.2/85.3**	**83.5/84.7**
Dataset	CIFAR-100 (kNN / Linear)				
Baseline (ResNet-18)	49.4/51.3	50.3/53.8	51.8/55.3	52.0/56.3	52.4/56.8
w/ DCL (ResNet-18)	**51.1/55.4**	**54.3/58.3**	**54.6/58.9**	**54.9/58.5**	**55.0/58.4**
Dataset	STL-10 (kNN / Linear)				
Baseline (ResNet-18)	74.1/76.2	77.6/77.8	79.3/80.0	80.7/81.3	81.3/81.5
w/ DCL (ResNet-18)	**82.0/85.2**	**82.8/86.3**	**81.8/86.1**	**81.2/85.7**	**81.0/85.6**

Figure 3.3 indicates that without DCL, the top-1 accuracy drastically drops when batch size (SimCLR) or queue (MoCo) becomes very small. While with DCL, the performance keeps steadier than baselines (SimCLR: -4.1% vs. -8.3%, MoCo: -0.4% vs. -5.9%).

Specifically, Figure 3.3 further shows that in SimCLR, the performance with DCL improves from 61.8% to 65.9% under 256 batch size; MoCo with DCL improves from 54.7% to 60.8% under 256 queues. The comparison fully demonstrates the necessity of DCL, especially when the number of negatives is small. Although batch size increases to 1024, DCL (66.1%) still improves over the SimCLR baseline (65.1%).

We further observe the same phenomenon on ImageNet-100 data. Table 3.1 shows that, with DCL, the top-1 linear performance only drops 2.3% compared to the InfoNCE baseline (SimCLR) of 7.1% when the batch size is varied.

In summary, it is worth noting that, while the batch size is small, the strength of $q_{B,i}$, which is used to push the negative samples away from the positive sample, is also relatively

Table 3.2: Comparisons between SimCLR baseline, DCL, and DCLW. The linear and kNN top-1 (%) results indicate that DCL improves baseline performance, and DCLW further provides an extra boost. Note that results are under batch size 256 and epoch 200. All models are both trained and evaluated with the same experimental settings. The backbones are ResNet-18 and ResNet-50 for CIFAR and ImageNet, respectively.

Dataset	CIFAR10 (kNN)	CIFAR100 (kNN)	ImageNet-100 (linear)	ImageNet-1K (linear)
SimCLR	81.4	52.0	80.7	61.8
DCL	84.2 (**+2.8**)	54.9 (**+2.9**)	83.1 (**+2.4**)	65.9 (**+4.1**)
DCLW	**84.8** (+3.4)	**55.2** (+3.2)	**84.2** (+3.5)	**66.9** (+5.1)

Table 3.3: Improve the DCL model performance on ImageNet-1K with tuned hyperparameters: temperature and learning rate, and stronger image augmentation. Note that models are trained with 256 batch size and 200 epochs.

ImageNet-1K (256 Batch size; 200 epoch)	Linear Top-1 Accuracy (%)
DCL	65.9
+ optimal (τ, l_r) = (0.2, 0.07)	67.8 (**+1.9**)
+ asymmetric augmentation [46]	68.2 (**+0.4**)

weak. This phenomenon tends to reduce the efficiency of learning representation. While taking advantage of DCL alleviates the performance gap between small and large batch sizes. Hence, through the analysis, we find out DCL can simply tackle the batch size issue in contrastive learning. With this considerable advantage given by DCL, general SSL approaches can be implemented with fewer computational resources or lower standard platforms. Compared to InfoNCE, DCL is more applicable across all large-scale SSL applications.

DCL on CIFAR and STL10. For STL10, CIFAR10, and CIFAR100, we implement DCL with ResNet-18 as encoder backbone. In Table 3.1, it is observed that DCL also demonstrates its strong effectiveness on small-scale benchmarks. In the evaluation (kNN / Linear) summary, DCL outperforms its baseline by 4.8% / 5.3% (CIFAR10) and 1.7% / 4.4% (CIFAR100) under a small batch size 32. The accuracy (kNN / Linear) of the SimCLR baseline on STL10 is also improved significantly by 7.9% / 9.0%.

Decoupled Objective with Re-Weighting DCLW. We only replace L_{DC} with L_{DCW} with no possible advantage from additional tricks. Both DCL and the baselines apply the same training instruction of the OpenSelfSup benchmark for fairness. Note that we empirically choose $\sigma = 0.5$ in the experiments. Results in Table 3.2 indicates that, DCLW achieves extra 5.1% (ImageNet-1K), 3.5% (ImageNet-100) gains compared to the baseline. For CIFAR data, an extra 3.4% (CIFAR10) 3.2% is gained from the addition of DCLW. It is worth noting that, trained with 200 epochs, DCLW reaches 66.9% with batch size 256, surpassing the SimCLR baseline: 66.2% with batch size 8192.

3.4.3 Ablations

We perform extensive ablations on the hyperparameters of DCL on both ImageNet data and other small-scale data, i.e., CIFAR and STL10. By seeking better configurations empirically, we see that DCL gives consistent gains over the standard InfoNCE baselines (SimCLR and MoCo-v2). In other ablations, we see that DCL achieves more gains over both SimCLR and MoCo-v2, i.e., InfoNCE-based baselines, also when training for 100 epochs only.

DCL Ablations on ImageNet. In Table 3.3, we have slightly improved the DCL model performance on ImageNet-1K: 1) tuned hyperparameters, temperature τ and learning rate ; 2) asymmetric image augmentation (e.g., BYOL). To obtain a stronger baseline, we conduct an empirical hyperparameter search with batch size 256 and 200 epochs. This improves DCL from 65.9% to 67.8% top-1 accuracy on ImageNet-1K. We further adopt the asymmetric augmentation policy from BYOL and improve DCL from 67.8% to 68.2% top-1 accuracy on ImageNet-1K.

DCL Ablations on CIFAR. Further experiments are conducted based on the ResNet-50 backbone and large learning epochs (i.e., 500 epochs). The DCL model with kNN eval, batch size 32, and 500 epochs of training could reach 86.1% compared to 82.2%. For the following experiments in Table 3.4, we show DCL ResNet-50 performance on CIFAR10 and CIFAR100. In these comparisons, we vary the batch size to show the effectiveness of DCL.

MoCo-v2 with DCL. We are aware that it is more convincing to compare the proposed DCL against a more compelling version, MoCo-v2. Comparisons on both ImageNet-1K

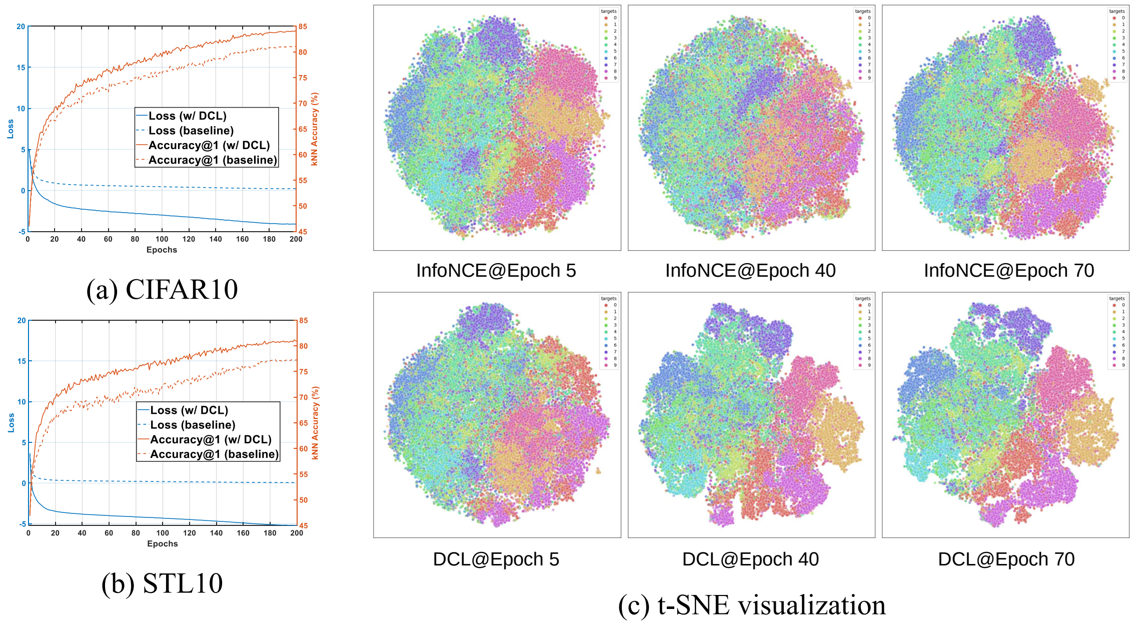
Table 3.4: The comparisons with/without DCL under various batch sizes from 32 to 512 on ResNet-50.

Architecture@epoch	ResNet-50@500 epoch									
Dataset	CIFAR10 (kNN)					CIFAR100 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	82.2	85.9	88.5	88.9	89.1	49.8	55.3	59.9	60.6	61.1
SimCLR w/ DCL	**86.1**	**88.3**	**89.9**	**90.1**	**90.3**	**54.3**	**58.4**	**61.6**	**62.0**	**62.2**

Table 3.5: Linear top-1 accuracy (%) comparison with MoCo-V2 on ImageNet-1K and ImageNet-100.

Queue Size	32	64	128	256	8192	64	256	65536
Dataset	ImageNet-100 (Linear)					ImageNet-1K (Linear)		
MoCo-v2 Baseline (ResNet-50)	73.7	76.4	78.7	78.7	79.8	63.9	67.1	67.5
MoCo-v2 w/DCL (ResNet-50)	**76.2**	**78.3**	**79.6**	**79.6**	**80.5**	**65.8**	**67.6**	**67.7**

and ImageNet-100 in Table 3.5 indicate that DCL becomes significantly more effective than MoCo-v2 when the queue size gets smaller.



(a) CIFAR10

(b) STL10

InfoNCE@Epoch 5

InfoNCE@Epoch 40

InfoNCE@Epoch 70

DCL@Epoch 5

DCL@Epoch 40

DCL@Epoch 70

(c) t-SNE visualization

Figure 3.4: Comparisons between DCL and InfoNCE-based baseline (SimCLR) on (a) CIFAR10 and (b) STL10 data. DCL speeds up the model convergence during the SSL pre-training and provides better performance than the baseline on CIFAR and STL10 data. (c) t-SNE visualization of CIFAR-10 with 32 batch size. DCL shows a stronger separation force between the features than SimCLR.

Few Learning Epochs. DCL can alleviate the shortcoming of the traditional contrastive learning framework, which needs a large batch size long learning epochs to achieve higher

Table 3.6: ImageNet-1K top-1 accuracy (%) on SimCLR and MoCo-v2 with/without DCL under few training epochs. We further list results under 200 epochs for clear comparison. With DCL, the performance of SimCLR trained under 100 epochs nearly reaches its performance under 200 epochs. The MoCo-v2 with DCL also reaches higher accuracy than the baseline under 100 epochs.

	SimCLR	SimCLR w/ DCL	MoCo-v2	MoCo-v2 w/ DCL
100 Epoch	57.5	64.6	63.6	64.4
200 Epoch	61.8	65.9	67.5	67.7

performance. The previous state-of-the-art, SimCLR, heavily relies on large quantities of learning epochs to obtain high top-1 accuracy. (e.g., 69.3% with up to 1000 epochs). DCL aims to achieve higher learning efficiency with few learning epochs. We demonstrate the effectiveness of DCL in InfoNCE-based frameworks SimCLR and MoCo-v2 [20]. We choose the batch size of 256 (queue of 65536) as the baseline and train the model with only 100 epochs. We make sure other parameter settings are the same for a fair comparison. Table 3.6 shows the result on ImageNet-1K using linear evaluation. With DCL, SimCLR can achieve 64.6% top-1 accuracy with only 100 epochs compared to SimCLR baseline: 57.5%; MoCo-v2 with DCL reaches 64.4% compared to MoCo-v2 baseline: 63.6% with 100 epochs pre-training.

We further demonstrate that, with DCL, learning representation becomes faster during the early stage of training compared to the InfoNCE-based learning scheme. The reason is that DCL successfully solves the decoupled issue between positive and negative pairs. Figure 3.4 on (a) CIFAR10 and (b) STL10 shows that DCL improves the speed of convergence and reaches higher performance than the baseline on CIFAR and STL10 data. The t-SNE visualization in Figure 3.4 (c) also supports the proposed theoretical derivation that removing the batch-size dependent impact (i.e., NPC multiplier) should improve representation learning abilities over the InfoNCE-based learning scheme.

Table 3.7: Linear top-1 accuracy (%) comparison of SSL approaches on ImageNet-1K. Given lower computational budget, DCL model are better than recent SOTA approaches. Its effectiveness **does not rely on** large batch size and epochs (SimCLR [19], NNCLR [39]), momentum encoding (BYOL [46], MoCo-v2 [20]), or other tricks such as stop-gradient (SimSiam [21]) and multi-cropping (SwAV [12]).

ResNet-50 w/	SimCLR	BYOL	SwAV	MoCo-v2	SimSiam	Barlow Twins	NNCLR	NNCLR +DCL
Epoch	400			400		300	1000	400
Batch Size	4096			256		256	256 / 512	256 / 512
ImageNet-1K (Linear)	69.8	73.2	70.7	71.0	70.8	70.7	68.7 / 71.7	**71.1 / 72.3**

3.5 Discussion

Comparison with other SOTA SSL Approaches. The primary goal of this work is to provide an efficient and effective improvement to the widely used InfoNCE-based contrastive learning, where we decouple the positive and negative terms to achieve better representation quality. DCL is less sensitive to suboptimal hyperparameters and achieves competitive results with minimal requirements. Its effectiveness does not rely on large batch size and learning epochs, momentum encoding, negative sample queues, or additional tactics (e.g., stop-gradient and multi-cropping). Overall, DCL provides a more robust baseline for the contrastive-based SSL approaches. Though this work aims not to provide a SOTA SSL approach, DCL can be combined with the SOTA contrastive learning methods, such as NNCLR [39], to achieve better performance without large batch size and learning epochs. In Table 3.7, we provide extensive comparisons to SOTA SSL approaches on ImageNet-1K to validate the effectiveness of DCL. In Table 3.8, we further show that DCL achieves competitive results compared to VICReg [2], Barlow Twins [139], SimSiam [21], SwAV [11], and DINO [13] on ImageNet-100 and CIFAR-10.

Generalization of DCL to Different Domains. DCL can be easily adapted to different domains (e.g., speech and language models) to achieve competitive performance. We demonstrate that DCL can be combined with SOTA SSL speech models, e.g., wav2vec 2.0 [1] which uses transformer backbone and requires enormous computation resources. We evaluate wav2vec 2.0 on its downstream tasks and perform better by applying the DCL method. Detailed results and discussion can be found in Appendix. To the best of our knowledge, DCL can be potentially combined with a transformer-based language

Table 3.8: kNN & linear top-1 accuracy (%) comparison of SSL approaches on CIFAR10 and ImageNet-100.

ResNet-18 @ 256 Batch Size	DINO	SwAV	SimSiam	VICReg	Barlow Twins	NNCLR	NNCLR+DCL
CIFAR-10, 1000 Epoch (kNN)	89.5	89.2	90.5	92.1	92.1	91.8	**92.3**
ImageNet-100, 400 Epoch (Linear)	74.9	74.0	74.5	79.2	80.2	79.8	**80.6**

Table 3.9: Results of DCL and SimCLR with large batch size and learning epochs.

ImageNet-1K (ResNet-50)	Batch Size	Epoch	Top-1 Accuracy (%)
SimCLR	256	200	61.8
SimCLR	256	400	64.8
SimCLR	1024	400	67.3
SimCLR w/ DCL	256	200	67.8 (**+6.0**)
SimCLR w/ DCL	256	400	69.5 (**+4.7**)
SimCLR w/ DCL	1024	400	69.9 (**+2.6**)

model, CLIP [98], which uses a very large batch size of 32768. With DCL, CLIP shall maintain its complexity and achieve huge learning efficiency when the batch size becomes smaller. Note that it has been implemented by [125].

DCL Convergence for Large Batch Sizes. The performance of DCL appears to have less gain compared to InfoNCE-based baseline when the batch size is large. According to Figure 3.1 and the theoretical analysis, the reason is that the NPC multiplier $q_B \to 0$ when the batch size is large (e.g., 1024). As shown in the analysis, InfoNCE loss converges to the DCL loss as the batch size approaches infinity. With 400 training epochs, the ImageNet-1K top-1 accuracy slightly increases from 69.5% to 69.9% when the batch size increases from 256 to 1024. Please refer to Table 3.9.

3.6 Conclusion

We identify the negative-positive-coupling (NPC) effect in the widely used InfoNCE loss, making the SSL task significantly easier to solve with smaller batch size. By removing the NPC effect, we reach a new objective function, *decoupled contrastive learning* (DCL). The proposed DCL loss function requires minimal modification to the SimCLR baseline and provides efficient, reliable, and nontrivial performance improvement on various benchmarks. Given the conceptual simplicity of DCL and that it requires neither momentum encoding, large batch size, or long epochs to reach competitive performance. Notably, DCL can be combined with the SOTA contrastive learning method, NNCLR, to achieve 72.3% ImageNet-1K top-1 accuracy with 512 batch size in 400 epochs. We wish that DCL can serve as a strong baseline for the contrastive-based SSL methods. Further, an important lesson from the DCL loss is that a more efficient SSL task shall maintain its complexity when the batch size becomes smaller.

Chapter 4 Learning with Real-World Data Distributions

4.1 Introduction

The performance of an image classification model critically depends on the underlying data distribution, both during the training and the testing stages. For the majority of real-world applications, their underlying data distributions can substantially deviate from those of conventional benchmark collections established solely for research evaluations. Indeed, the distribution of real-world data is often not regular, and for many practical applications, it tends to be more or less fine-grained and even complicated with long-tailed imbalance. To account for such discrepancies in data distribution, recent datasets, *e.g.*, iNaturalist 2018 [121], have been proposed to bridge the gap so that their resulting classification techniques can be widely applied. Figure 4.1 illustrates two notable and challenging aspects of iNaturalist. First, it exhibits a long-tailed distribution, characterized by extremely imbalanced ratios between head and tail categories. In particular, the almost three orders of magnitude difference in the number of training instances embodied in the long-tailed distribution imposes a difficult task in learning proper representations of tail classes. Second, the object categories in this dataset are also fine-grained, while inter-class similarity and intra-class variations are subtly intertwined. Performing classifications over iNaturalist 2018 is essentially a daunting task, no matter what a specific group (`many`, `medium` or `few`) of fine-grained object classes is under consideration. Motivated by these challenges, we aim to simultaneously address both the fine-grained and long-

doi:10.6342/NTU202301574

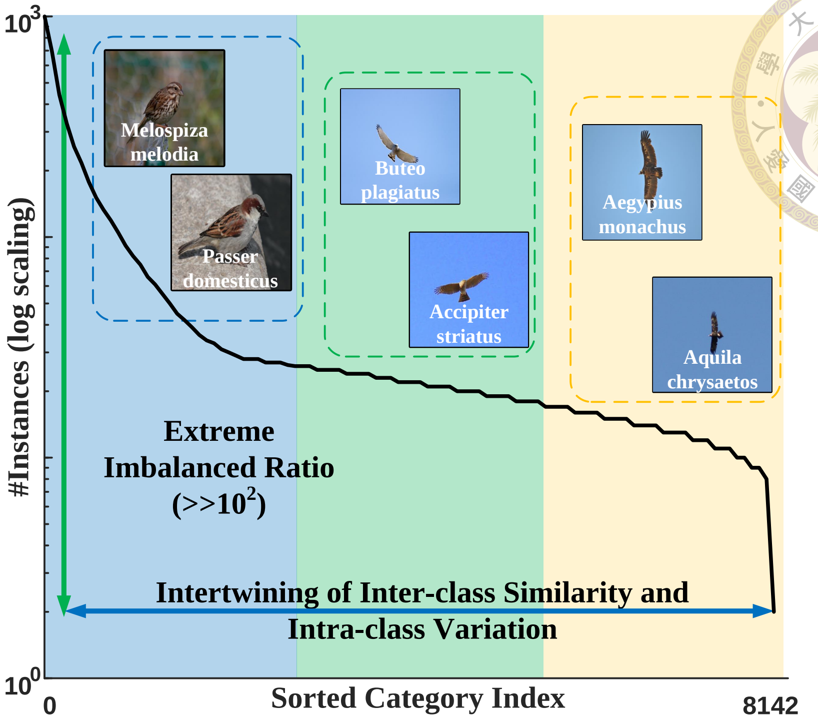


Figure 4.1: The distribution of real-world data can include various subtleties such as *fine-grained* and *long-tailed* complications. In terms of algorithm design, these two aspects of challenges can be exemplified by iNaturalist 2018 [121]. As illustrated, the extremely imbalanced numbers of instances among its object classes could derange learning proper features of tail classes for effective classification. Meanwhile, model overfitting could become a major concern in that it is hard to disentangle the fine-grained ambiguities due to the inter-class similarities and intra-class variations in the underlying object categories.

tailed issues in designing classification techniques for practically dealing with real-world data.

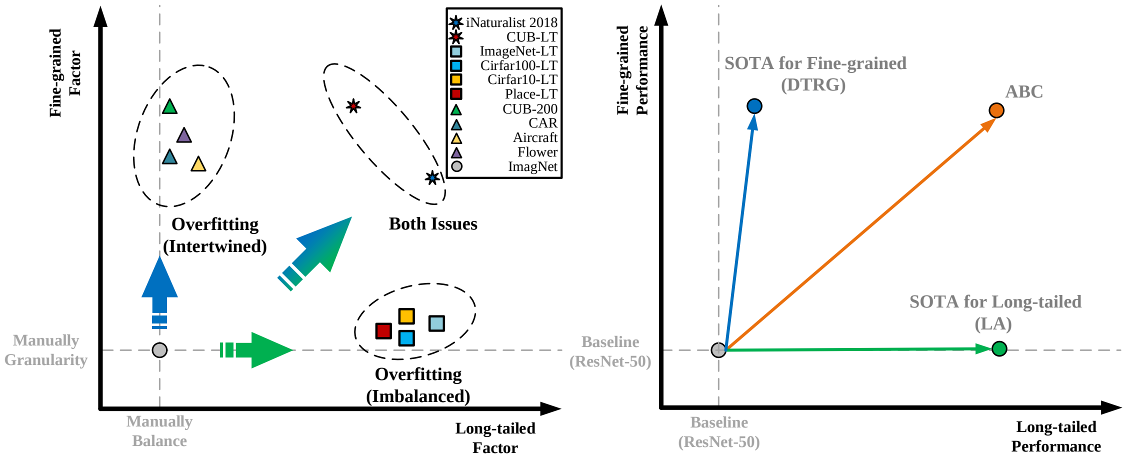


Figure 4.2: Left: Different datasets exhibit varying degrees of long-tailed and fine-grained characteristics. Right: Mainstream techniques focus on solving one aspect of the two characteristics, where DTRG [81] (blue dot) and LA [89] (green dot) are respectively current SOTA techniques for tackling the fine-grained and long-tailed classification tasks.

doi:10.6342/NTU202301574

Fine-grained visual classification (FGVC) is an active and challenging problem in computer vision. Such a recognition task differs from the classical problem of large-scale visual classification (LSVC) by focusing on differentiating *similar* sub-categories of the same meta-category. While the inter-class similarity among the object categories is pervasive, the intra-class variations further impose ambiguities in learning a unified and discriminative representation for the FGVC task. On the other hand, considering the issue of long-tailed distribution brings in another aspect of difficulty in developing practical classification techniques. The significantly large numbers of samples from head categories tend to dominate the training procedure. Even with sophisticated learning strategies, the resulting classification model often ends up performing poorly for the tail categories, compared with the expected result on the head counterparts. In fact, the performance curve somewhat resembles the shape of a long-tailed distribution.

We note from existing literature of object classification research that there are only a few attempts to simultaneously solve the two aforementioned challenging issues. Relevant developments mainly focus on tackling either of the two tasks. In FGVC, most of the recent research efforts have converged to learning pivotal local/part details related to distinguishing fine-grained categories *e.g.*, [43, 135, 146]. Moreover, to improve the classification performance further, a number of these efforts require the fusion of several sophisticated computer vision techniques, such as in [36, 44]. In resolving the long-tailed difficulty, previous approaches have drawn on balanced data sampling to rectify their model training [56, 63, 132]. For example, the recent technique of [63] first learns the representation and then refines the classifier by balanced sampling. All these different research attempts involve varying degrees of fine-grained and long-tailed factors. As shown in Figure 4.2 (Left), we take the maximum imbalanced ratio and the normalized feature cosine similarity between object categories as the respective criterion to measure the fine-grained and long-tailed factors and characterize the two aspects of difficulties among popular datasets adopted in object recognition research. Moreover, Figure 4.2 (Right) indicates that a purely fine-grained state-of-the-art (SOTA) approach does not necessarily perform well for the long-tailed case, and vice versa, while our approach provides a unified solution to tackling the two challenging issues of image classification.

doi:10.6342/NTU202301574

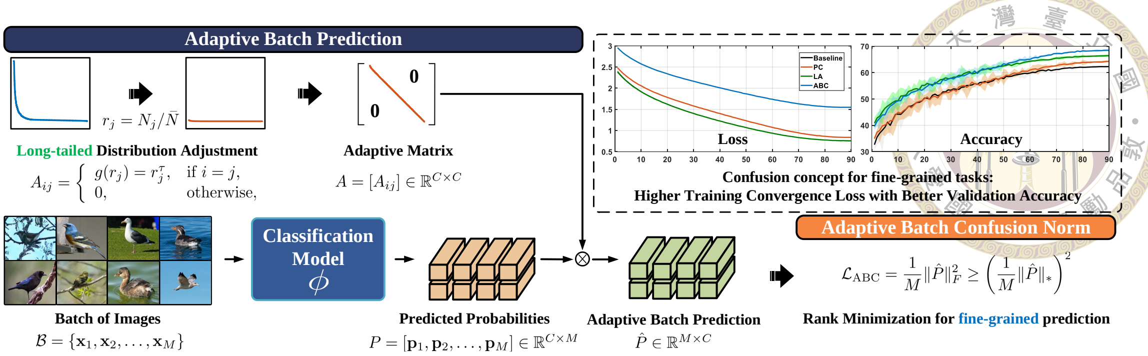

Adaptive Batch Prediction
$r_j = N_j / \bar{N}$
0
0
Long-tailed Distribution Adjustment
$A_{ij} = \begin{cases} g(r_j) = r_j^{\tau}, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$
Adaptive Matrix
$A = [A_{ij}] \in \mathbb{R}^{C \times C}$
Baseline
PC
LA
ABC
Loss
Accuracy
0.5
1
1.5
2
2.5
3
30
40
50
60
70
0
10
20
30
40
50
60
70
80
90
Confusion concept for fine-grained tasks:
Higher Training Convergence Loss with Better Validation Accuracy
Adaptive Batch Confusion Norm
Classification Model
ϕ
Batch of Images
$\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$
Predicted Probabilities
$P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M] \in \mathbb{R}^{C \times M}$
$\otimes$
Adaptive Batch Prediction
$\hat{P} \in \mathbb{R}^{M \times C}$
$\mathcal{L}_{\mathrm{ABC}} = \frac{1}{M} \|\hat{P}\|_F^2 \geq \left(\frac{1}{M} \|\hat{P}\|_* \right)^2$
Rank Minimization for fine-grained prediction

Figure 4.3: Overview of *adaptive batch confusion norm* (ABC-Norm). The adaptive batch prediction $\hat{P}$ can be obtained by class-wise modulating the predicted probabilities P with respect to the adaptive matrix A that encodes the underlying data distribution. Our formulation then adds slight classification confusions to yield an adversarial regularization effect in model training. Despite that ABC-Norm converges to a higher training loss than other techniques, it indeed achieves better validation accuracy.

In this work, we focus on establishing a fundamental approach based on exploring the characteristics of the real-world data distributions rather than relying on various data augmentation schemes and sophisticated DNN-based engineering tricks. From the two plots in Figure 4.3, we observe that when the objective function during training converges very close to zero, the results in testing are often not the best. To avoid being trapped with over-optimizing the underlying model, previous approaches have adopted regularization techniques to resolve this matter. Take, for example, the inclusion of *margin* in the triplet loss [105]. The design principle of triplet loss is to separate positive and negative samples by at least a default margin, say m, which turns out to play a pivotal role in boosting the learning efficacy. Different from typical regularization techniques, it implicitly raises the learning difficulty of the objective function, instead of limiting the model capacity.

The concept of incorporating extra difficulty into training has also been proposed in dealing with the FGVC problem. Pairwise Confusion (PC) [37] and Maximum Entropy (MaxEnt) [38] are two such approaches, closely related to our proposed method. PC argues that slightly confusing the model in training can prevent overfitting problems. MaxEnt observes that the data diversity of FGVC is usually smaller than that of a large-scale classification dataset, *e.g.*, ImageNet. It thus presumes that the entropy of the model's prediction should tend to be higher than that of typical classification scenarios. Both PC and MaxEnt add a confusion-like loss to improve the FGVC performances of their resulting models. Still, there are currently no relevant arguments in addressing fine-grained and

doi:10.6342/NTU202301574

long-tailed issues simultaneously.

We are thus motivated to develop a new classification technique, termed *adaptive batch confusion norm* (ABC-Norm), to regularize its corresponding *adaptive batch prediction* (ABP) matrix to better account for real-world data distributions. ABC-Norm can be used to deal with both fine-grained and long-tailed factors and to construct an adversarial loss for enhancing the training efficacy. Optimizing with respect to the ABC-Norm drives the learning process to (class-wise) adaptively add confusions to achieve better classification results. We also provide a mathematical derivation to justify the concept and the ideas it represents. Figure 4.3 illustrates an overview of ABC-Norm. We characterize the advantages of our method as follows.

- The computation of ABC-Norm regularization is efficient and does not incur significant increase in training time.
- Unlike related techniques, *e.g.*, [63, 129] that decouple representation learning from classification or learn multiple distribution-aware experts, our regularization-based method leads to an end-to-end trainable implementation.
- Without relying on complicated model design or sophisticated data augmentations such as in, *e.g.*, [36, 81, 117], ABC-norm not only provides a unified solution to resolving fine-grained and long-tailed issues but also improves the baselines to achieve competitive classification results.

4.2 Related Work

In addressing conventional computer vision tasks, the underlying distribution of training data is often relatively balanced. The numbers of samples across various object categories do not differ substantially, and in addition, the diversity among the categories is typically high. However, the data distributions for real-world applications are far more complicated; it could even contain fine-grained and long-tailed complexities at the same time. Recent related work for image classification tends to emphasize either aspect of

doi:10.6342/NTU202301574

the two difficulties, but not both. Taking such development into account, we divide the literature survey of relevant techniques into two groups, namely, *fine-grained visual classification* and *long-tailed visual recognition*.

4.2.1 Fine-grained visual classification

The Fine-Grained Visual Classification (FGVC) problem is notably characterized by two intriguing properties, significant inter-class similarity and intra-class variations, which cause learning an effective FGVC classifier a challenging task. Driven by impressive research progress, the setting of FGVC has gradually evolved from strong labels to weak labels.

Early work In the initial efforts for tackling FGVC, the developed methods mostly assume that the training datasets are made with comprehensive annotations, such as the part location labels in CUB-200 [122]. Along this line, Berg *et al*. [5] explore the labeled part locations to eliminate highly similar object categories for improving the classifier. Huang *et al*. [58] introduce an approach established based on a two-stream classification network to capture both object-level and part-level information explicitly. However, due to the rapid research advances in visual classification, the most recent FGVC approaches are designed to complete the model learning based on the category labels solely. Hence, without accessing the part location labels, how to learn the discriminative parts automatically becomes the next research direction.

Discriminative parts Existing FGVC approaches usually draw on data augmentations and specific attention mechanisms to effectively learn the discriminative parts. Yang *et al*. [135] propose a self-supervision mechanism to localize informative regions without the need of bounding-box and part annotations. Wang *et al*. [131] present a filter bank within a CNN framework to learn high-quality discriminative patches. Zheng *et al*. [146] introduce a trilinear attention sampling network for fine-grained image recognition, which can learn rich feature representations from hundreds of part proposals. Chen *et al*. [24]

doi:10.6342/NTU202301574

propose a *destruction and construction learning* (DCL) framework for fine-grained image recognition. DCL partitions each training image into several local regions and then shuffles them by a *region confusion mechanism* (RCM). It implicitly excludes the global object structure information and forces the model to predict the category label based on local information. Moreover, construction learning can model the semantic correlation among parts of the object. In other words, the ability to identify the object category from local details is expected to be enhanced through shape destruction. Du *et al*. [36] apply a progressive training strategy to address the fine-grained classification task. They formulate a framework named *progressive multi-granularity* (PMG) training with two key components. One is a training strategy that progressively fuses multi-granularity features, and the other is a puzzle generator to form images containing information of different granularity levels. Chang *et al*.[14] propose a *mutual-channel loss* (MCLoss) that drives the model to learn channel diversity and emphasize different discriminative regions. In summary, the above techniques are established based on employing richer augmentations and specialized attention mechanisms. In the case of the top-performing PMG, each iteration requires four different phases of augmentation combined with four classifiers. Although the results are state-of-the-art, PMG requires more training time and extensive model parameters.

Auxiliary task variants Several related approaches include an additional branch to explore auxiliary information. Shu *et al*. [111] propose a self-training framework for FGVC with insufficient data annotation by considering an additional auxiliary task path to generate pseudo labels. They leverage the Grad-CAM technique [106] to generate salient regions for seeking discriminative parts, which can be further extended to yield multiple attention maps for improving the quality of the representation. Chang *et al*. [15] introduce a novel FGVC problem setting by generalizing it from single-label to multiple-label predictions on a predefined label hierarchy. A user study is also provided to show that a multi-granular label hierarchy is more expressive and probably preferred. Their proposed solution shows that the inherent coarse-fine hierarchical relationship can improve FGVC performance.

doi:10.6342/NTU202301574

Regularization effects The regularization-related formulations for dealing with intra-class variations and inter-class similarity in FGVC generally have two main implications. First, it can be applied to alleviate the overfitting problem in learning an FGVC model. Dubey *et al*. [37] propose to divide each batch into two groups and train the model with a loss function including *pairwise confusion* (PC). The design reasons that bringing the class-wise probabilities closer could prevent the learned FGVC model from overfitting. Second, the regularization tactic implicitly maximizes the prediction entropy. MaxEnt [38] assumes that the data diversity of FGVC is intuitively smaller than the large-scale dataset, ImageNet. So the prediction entropy for the FGVC task is reasonable to become more prominent than usual. In other words, regularization approaches escalate the training difficulty on the total loss, which complicates the training convergence and forces the model to search for an ideal local minimum. In [81], Liu *et al*. introduce *dynamic target relation graphs* (DTRG) to address the fine-grained classification problem with a self-supervised regularization. DTRG evaluates every training sample to calculate the class center online. And then, DTRG aims to reduce the intra-class distance between each training feature and its corresponding class center, while keeping the class centers to be away from each other. It can be observed that the regularization principle of DTRG is quite different from the entropy-based confusion view entailed in PC and MaxEnt. In addition, the training process of DTRG is more intricate and also requires substantial augmentation techniques to strengthen the outcome of model learning.

4.2.2 Long-tailed visual recognition

Distribution re-balancing Existing techniques for long-tailed visual recognition that consider distribution re-balancing can be divided into two groups: *re-sampling* and *re-weighting*. As described in [16, 35, 50, 87], re-sampling involves adjusting the sampling frequencies of different categories based on their sample count via under-sampling for head categories and over-sampling for tail categories. The approach of class-balanced sampling [108] weights each image based on the number of samples in its category. In [48], the dynamic-sampling mechanism, termed as *repeat factor sampling* (RFS) by Gupta

doi:10.6342/NTU202301574

et al. [48], also aims to balance the number of instances across categories. While the goal of re-sampling is to reduce the overfitting of head data, the tactic may not always be a reliable solution. It could cause over-sampling of small amounts of tail data, resulting in insufficient sample diversity and under-sampling of large amounts of head data, leading to insufficient learning.

Loss re-weighting The strategy of re-weighting has been widely utilized in the loss calculation of a classification task. Unlike re-sampling, re-weighting offers greater flexibility and ease of computation, making it a popular choice for resolving the challenge of long-tailed distribution in more complex tasks such as object detection and instance segmentation. When an image contains multiple objects that need to be detected or segmented, it is often more manageable to reweigh the loss at the image level rather than sample by category. Re-weighting implementations range from reverse weighting based on category distribution to more advanced methods such as Hard Example Mining [110], Focal loss [78], and Label-Distribution-Aware Margin (LDAM) loss [8], which adjust the weight according to the classification credibility without the need for category knowledge. Owing to its ease of implementation, re-weighting has been shown to yield competitive results in complex tasks [27, 61, 113].

Model training strategies Another viewpoint for solving the long-tailed visual recognition problem is that the re-balancing technique should be applied only to the classifier, and the distribution of image features during representation learning should remain unchanged. This two-stage training strategy, in which the classifier is trained with re-balanced data and the representation is learned with the original data, is considered an effective solution for handling the long-tailed distribution. Kang *et al*. [64] divide the training of a long-tailed classification model into two steps, first directly learning a representation model from traditional classification with raw data and then connecting a separate classifier via class-balanced sampling learning. Zhou *et al*. [147] realize the two-step learning with a two-branch model where both branches share parameters and are dynamically weighted, one branch learning from raw data and the other from re-sampled data. Li *et al*. [75] adopt

doi:10.6342/NTU202301574

a two-stage learning approach and introduce a balanced group softmax module into the classification head. Meanwhile, Hu *et al*. [55] tackle the long-tailed distribution scenario through incremental learning from the head to the tail. Wang *et al*. [126] add a separate classifier to calibrate prediction logits, while Tang *et al*. [115] compute the moving average vector of a feature in the traditional training framework, excluding it from the gradient calculation. Menon *et al*. [89] revisit the classic idea of logit adjustment based on statistical information, encouraging a large relative margin between the logits of rare and dominant labels. Tian *et al*. [117] address long-tailed object recognition with the VL-LTR model, which jointly trains the image and text encoders by considering co-embedding between class-wise linguistic and visual information. Wang *et al*. [123] establish a quantitative measure, defining an overlap coefficient between von Mises-Fisher distributions, to evaluate representation quality for long-tailed learning.

In summary, the majority of the aforementioned methods for long-tailed learning emphasize exploring the aspect of data distribution. Such approaches, as we have just described, can be broadly categorized into three groups: distribution re-sampling, loss re-weighting, and model training strategies. In this work, we introduce a novel approach to addressing fine-grained and long-tailed issues at the same time. By infusing pivotal statistical characteristics of the data distribution into an adaptive matrix, the proposed regularization learning with an adversarial loss is shown to be a promising solution.

4.3 Our Method

Consider now learning a classification model Φ, as illustrated in Figure 4.3, with respect to a dataset $\mathcal{D}$ of C object categories, where each sample $\mathbf{x} \in \mathcal{D}$ is specified with a one-hot class label vector $\mathbf{y}$. For an arbitrary training batch $\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ from $\mathcal{D}$ and $M \leq C$, forward propagation via Φ yields M predicted (softmax) probabilities, denoted as

$$P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M] \in \mathbb{R}^{C \times M}, \tag{4.1}$$

doi:10.6342/NTU202301574

where $\Phi(\mathbf{x}_i) = \mathbf{p}_i \in \mathbb{R}^C$ is the predicted probability distribution. Let $\mathbf{p}_{i,j}$ be the probability of $\mathbf{x}_i$ being class j. We have $\sum_{j=1}^{C} \mathbf{p}_{i,j} = 1$. The batch prediction matrix P in (4.1) is central to our approach—its rank property is closely related to how our approach resolves the fine-grained issue.

The data distribution over the C object classes in $\mathcal{D}$ reflects the long-tailed characteristic. Let N_j be the sample size of class j and $\bar{N}$ be the averaged sample size over the C classes. We express the ratio of N_j to $\bar{N}$ as $r_j = N_j/\bar{N}$ and consider a unit-coefficient power function of r_j, namely $g(r_j) = r_j^{\tau}$, to model the underlying long-tailed distribution. Note that the real-valued power τ is a hyper-parameter of our method, and its value is to be adjusted according to the extent of long-tailed distribution. Specifically, to encode the class-wise imbalance, we define an adaptive matrix $A = [A_{ij}] \in \mathbb{R}^{C \times C}$ by

$$A_{ij} = \begin{cases} g(r_j) = r_j^{\tau}, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases} \tag{4.2}$$

where (as we will explain later) the value of τ, along with r_j, reflects the degree of long-tailed attribute and can be adaptively set to account for different application scenarios.

4.3.1 Adaptive Batch Confusion Norm

We aim to introduce a regularization based framework to simultaneously address the fine-grained and long-tailed issues of object classification. Based on (4.1) and (4.2), we construct an *Adaptive Batch Prediction* (ABP) matrix $\hat{P} \in \mathbb{R}^{M \times C}$ by

$$\hat{P} = P^{\intercal} A\,, \tag{4.3}$$

where the adjusted softmax prediction of each sample in $\mathcal{B}$ now forms a row vector of $\hat{P}$. Observe from (4.2) that how the adaptive matrix A modifies the prediction outputs depends on the exponent τ and the *imbalanced* factor r_j of each class j in the training data $\mathcal{D}$. When $r_j \to 1$ or the exponent $\tau \to 0$, A would approach the identity matrix I. In other words, the ABP matrix in (4.3) will be reduced to $P^{\intercal}$ when the distribution of training data is class-wise balanced, or τ is set to 0. Both cases exclude the long-tailed consideration of $\hat{P}$.

The main idea of our approach is to establish a unified regularization mechanism from the ABP matrix $\hat{P}$ so that the model training process can effectively improve its inference performance on our targeted classification scenarios. To this end, we propose the *Adaptive Batch Confusion Norm* (ABC-Norm) to assess the corresponding loss, expressed as $\mathcal{L}_{ABC}$, which realizes the desired regularization effects for addressing the fine-grained and long-tailed issues. Specifically, we define the loss term for the ABC-Norm regularization as

$$\mathcal{L}_{ABC} = \frac{1}{M} \| \hat{P} \|_F^2 \,, \tag{4.4}$$

where $\| \cdot \|_F$ denotes the Frobenius norm and M is the batch size as in (4.1). Unlike other existing techniques that are often developed by integrating several sophisticated classification modules to tackle the fine-grained or long-tailed difficulties, our formulation learns the proposed model by directly optimizing the following objective function:

$$\mathcal{L}_{\text{total}} = (1 - \lambda)\, \mathcal{L}_{\text{CE}} + \lambda\, \mathcal{L}_{\text{ABC}}, \tag{4.5}$$

where $\lambda \in [0, 1]$ is a weight parameter,

$$\mathcal{L}_{\text{CE}} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} \mathbf{y}_{i,j} \log \mathbf{p}_{i,j} \tag{4.6}$$

is the conventional cross-entropy loss, and

$$\begin{aligned} \mathcal{L}_{\text{ABC}} = \frac{1}{M} || \hat{P} ||_F^2 &= \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} \hat{\mathbf{p}}_{i,j}^2 \\ &= \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} A_{j,j} \mathbf{p}_{i,j}^2 . \end{aligned} \tag{4.7}$$

Empirically, we set $\tau = 0.5$ and consider $\lambda \in \{0.1, 0.3, 0.5\}$ for various datasets to achieve the best performance accounting for different fine-grained and long-tailed characteristics of real-world distributions. Since the regularization term $\mathcal{L}_{\text{ABC}}$ in (4.5) is the only factor

doi:10.6342/NTU202301574

that distinguishes our method from a vanilla classification scheme, the performance gains reported in our experiments are evidently owing to the ABC-Norm efficacy.

4.3.2 ABC-Norm: Justifications and Properties

The rank of the ABP matrix $\hat{P}$ in (4.3) plays a pivotal role in our formulation, and is closely related to the ABC-Norm regularization. Assume for the moment that minimizing $\mathcal{L}_{\text{ABC}}$ can lead to rank minimization of $\hat{P}$. It would then reduce the variability among the M softmax predictions of P from a batch $\mathcal{B}$, and infuse slight classification *confusions* into the training procedure. Whereas correct predictions would always be penalized with the confusion loss as in (4.5), the training would be driven to further improve the model by reducing the cross-entropy loss as much as possible, and consequently better solve the fine-grained classification problem. Such an adversarial regularization idea is analogous to enhancing the model learning by introducing an extra margin to increase the difficulty of a correct prediction.

It is known that the rank-related minimization problems are often NP-hard. We follow [99] and consider convex relaxation so that the underlying rank minimization of $\hat{P}$ can be reduced to the minimization of its nuclear norm,

$$\|\hat{P}\|_* = \sum_{i=1}^{M} \sigma_i(\hat{P})\,, \tag{4.8}$$

where $\sigma_i(\cdot)$ yields the ith singular value of the corresponding matrix. However, training a deep neural network with an objective function that involves solving singular values of a non-trivial matrix is not practically feasible. It is also the main reason that we do not establish the ABC-Norm regularization based on the nuclear norm. We instead consider minimizing its upper bound as in (4.4). In this way, rank minimization of $\hat{P}$ can be efficiently achieved by employing $\mathcal{L}_{ABC}$. To complete the mathematical derivation of our method, we are left to justify the following upper-bound property.

Property 1. *If the batch size M is set as less or equal to the number of classes C, then*

$$\mathcal{L}_{\text{ABC}} = \frac{1}{M}\|\hat{P}\|_F^2 \geq \left(\frac{1}{M}\|\hat{P}\|_*\right)^2 . \tag{4.9}$$

It follows that minimizing the nuclear norm of $\hat{P}$ can be achieved by including $\mathcal{L}_{\text{ABC}}$ in the total loss. That is, rank minimization is implicitly carried out during the model training of the classifier Φ. To verify the upper-bound property stated in (4.9), we have, from the matrix norm definitions and Cauchy-Schwarz inequality,

$$\begin{aligned}
\mathcal{L}_{\text{ABC}} &= \frac{1}{M}\|\hat{P}\|_F^2 = \frac{1}{M}\sum\nolimits_{i=1}^{M} \sigma_i^2(\hat{P}) \\
&= \frac{1}{M^2}\left(\sum\nolimits_{i=1}^{M} \sigma_i^2(\hat{P})\right)\left(\sum\nolimits_{i=1}^{M} 1^2\right) \\
&\geq \frac{1}{M^2}\left(\sum\nolimits_{i=1}^{M} \sigma_i(\hat{P}) \cdot 1\right)^2 \\
&= \left(\frac{1}{M}\sum\nolimits_{i=1}^{M} \sigma_i(\hat{P})\right)^2 \\
&= \left(\frac{1}{M}||\hat{P}||_*\right)^2 .
\end{aligned}$$

We now turn our attention to explaining how the adaptive matrix $A \in \mathbb{R}^{C\times C}$ in $\hat{P} = P^\intercal A$ is used in dealing with the long-tailed issue. Notice that A is a diagonal matrix whose jth diagonal entry $A_{jj} = r_j^\tau = (N_j/\bar{N})^\tau$ adjusts the predicted probability $\mathbf{p}_{i,j}$ for the jth class. When learning with a long-tailed training dataset $\mathcal{D}$, its head classes are those that include more training samples and thus have $r_j \gg 1$. Hence the adaptive effects on these head classes are to enforce more confusions/difficulties in classify their abundant samples. On the contrary, tail classes are characterized with $r_j \ll 1$ and the adaptive matrix A is used to instead lessen their confusion regularization so that learning with these scarce data can be guided by the cross-entropy loss.

doi:10.6342/NTU202301574

4.3.3 ABC-Norm vs. Relevant Regularization

To our knowledge, there are no existing regularization techniques that are developed to simultaneously tackle both the fine-grained and long-tailed classifications. The two most relevant approaches, but focusing on only the fine-grained aspect, are Pairwise Confusion (PC) [37] and Maximum Entropy (MaxEnt) [38]. We describe their design principles and relevance to the ABC-Norm regularization below.

PC Regularization This is the first work [37] that brings in the "confusion" concept to solve the fine-grained classification task. The purpose of confusion energy is twofold. Besides preventing the model training form overfitting, it implicitly increases the learning difficulty to aim for performance gains in testing. PC randomly divides each batch into two equal-size sub-batches. While computing the individual cross-entropy losses for each sample of the whole batch, it evaluates the pairwise confusion loss, denoted as $\mathcal{L}_{\text{PC}}$, by sampling from the two parts. Specifically, we have

$$\mathcal{L}_{\text{PC}} = \frac{2}{M} \sum_{i=1}^{M/2} \mathbb{I}(\mathbf{y}_i = \mathbf{y}_{i+M/2}) ||\mathbf{p}_i - \mathbf{p}_{i+M/2}||_2 \,, \tag{4.10}$$

where $\mathbb{I}(\cdot)$ is the indicator function to signal whether two paired training samples are of the same category.

MaxEnt Regularization The maximum entropy criterion is proposed in [38] to more effectively address the fine-grained classification problem. As the inter-class variations between fine-grained classes could be subtly minimal, MaxEnt regularization assumes no prior distributions other than the uniform one should be imposed on the softmax predictions. Analogous to PC, the maximum entropy regularization also increases the learning difficulty and therefore drives the optimization process to work harder in tackling the challenging classification scenario. The corresponding loss for MaxEnt regularization is defined as follows:

$$\mathcal{L}_{\text{MaxEnt}} = \frac{-1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} \mathbf{p}_{i,j} \log \mathbf{p}_{i,j} \,. \tag{4.11}$$

Comparing the three regularization schemes, ABC-Norm, PC and MaxEnt, their most distinction is that our formulation tackles not only the fine-grained but also the long-tailed difficulty. Furthermore, by setting the adaptive matrix A in (4.3) to the identity matrix I, we can look further into how their design improves the performance on fine-grained classification. The three techniques resemble each other by imposing adversarial difficulty in the model training to enhance the classification efficacy. For PC versus ABC-Norm, both are established based on the concept of confusion, while ABC-Norm has the advantage of exploring the adversarial measure from an entire batch at the same time, rather than the pairwise mechanism as in PC. For MaxEnt versus ABC-Norm, while the softmax prediction of each sample in the batch $\mathcal{B}$ being a uniform distribution is a minimum for both regularization losses, $\mathcal{L}_{\text{ABC}}$ is more general in accommodating other minima. In our experiments, we replace $\mathcal{L}_{\text{ABC}}$ in the total loss in (4.5) with $\mathcal{L}_{\text{PC}}$ and $\mathcal{L}_{\text{MaxEnt}}$, respectively to thoroughly compare their performances on various datasets and settings.

4.4 Experiments

4.4.1 Datasets

We conduct experiments on the six datasets listed in Table 4.1. In particular, our main objective is to evaluate the efficacy of the proposed ABC-Norm approach to the real-world classification challenges over the two datasets, CUB-LT [104] and iNaturalist2018 [121]. To further analyze its performance, we evaluate ABC-Norm on three fine-grained datasets (CUB, CAR, AIR) and a long-tailed dataset (ImageNet-LT), respectively. The results of our experiments demonstrate that ABC-Norm can effectively and efficiently tackle the challenging classification tasks posed by these benchmark datasets.

Real-world We begin by evaluating the ABC-Norm regularization on the two real-world datasets, CUB-LT [104] and iNaturalist2018 [121], each of which includes both fine-grained and long-tailed distribution characteristics. iNaturalist2018 is a large-scale col-

Table 4.1: Dataset splits in our experiments.

Dataset	# Train	# Val/Test	# Category
iNaturalist2018	437,513	24,426	8,142
CUB-LT	2,945	2,348	200
CUB	5,994	5,794	200
CAR	8,144	8,041	196
AIR	6,667	3,333	100
ImageNet-LT	115,846	20,000	1,000

lection. Owing to its challenging nature, as demonstrated in recent literature [9, 63], the performance on this dataset could serve as an objective measure for the effectiveness of each particular method.

FGVC We then compare solely the fine-grained classification results from four different regularization approaches, *adaptive batch confusion Norm* (ABC-Norm), *pairwise confusion* (PC) [37], *maximum entropy* (MaxEnt) [38], and *dynamic target relation graphs* (DTRG) [81] on the three popular fine-grained visual classification datasets, namely, CUB-200-2011 [122], Stanford Cars [69], and FGVC-Aircraft [88]. The size ratio between training and testing sets is about $1:1$ for CUB-200-2011 and Stanford Cars, and about $2:1$ for FGVC-Aircraft. The class distribution of the three datasets is nearly balanced, which can be used to measure the proposed method's performance only in the fine-grained scenario. Notice that the adaptive matrix A will be reduced to an identity matrix I in dealing with the balanced data distribution.

Long-tailed Finally, we carry out experiments on the long-tailed dataset ImageNet-LT [83], which can be considered to have a low fine-grained factor. The study aims to confirm the capability of ABC-Norm to tackle long-tailed learning over purely imbalanced datasets. In line with the definition in [63], we divide the categories into three groups: `Many`, `Medium`, and `Few`, representing the categories with instance numbers in ranges $(100, +\infty)$, $(20, 100]$, and $(0, 20]$, respectively.

4.4.2 Implementation Details

We now describe the implementation details of the experiments on the real-world, fine-grained, and long-tailed datasets. All results are obtained from end-to-end training, and the numerical outcomes represent the mean of three runs. We implement our method using the PyTorch framework [94] on a platform with four Nvidia V100 GPUs. The source code will be made available for public use.

Real-world These results pertain to the CUB-LT and iNaturalist2018 datasets. To ensure a fair comparison, our training settings mostly conform to those outlined in [63, 123]. The backbone network is ResNet-50 with an input size of 224×224 and 90 training epochs, optimized using SGD. The batch size is set to 16 for CUB-LT and 128 for iNaturalist2018. The initial learning rate is set to $0.004 \times M$, where M denotes the batch size, and is decreased by a cosine annealing schedule. The regularization weight λ is set to 0.5, and the value of τ for the adaptive matrix A is set to 0.5.

FGVC We evaluate the performance of the ABC-Norm on popular classification architectures, including ResNet series [53] and DenseNet-161 [57], in the fine-grained visual classification task. The training setup for the different regularization terms remains consistent. We adopt the data augmentation strategy from [24], using an input size of 448×448 and randomly applying horizontal flipping. The initial learning rate, the weighting factor λ, and τ are set to 0.008, 0.3, and 0.5, respectively. The training batch size is 16 when the GPU memory allows, and the adopted optimization algorithm is Momentum SGD with cosine annealing [84] for the learning rate decay. Taking account of the smaller scale of FGVC datasets compared to iNaturalist2018, we train the model for 200 epochs to assess the outcomes of different regularization methods.

Long-tailed visual recognition We further evaluate the proposed ABC-Norm on an imbalanced dataset, ImageNet-LT. The implementation details follow the training process described in [63]. We report results for both ResNeXt-50 and ResNeXt-152, and observe

doi:10.6342/NTU202301574

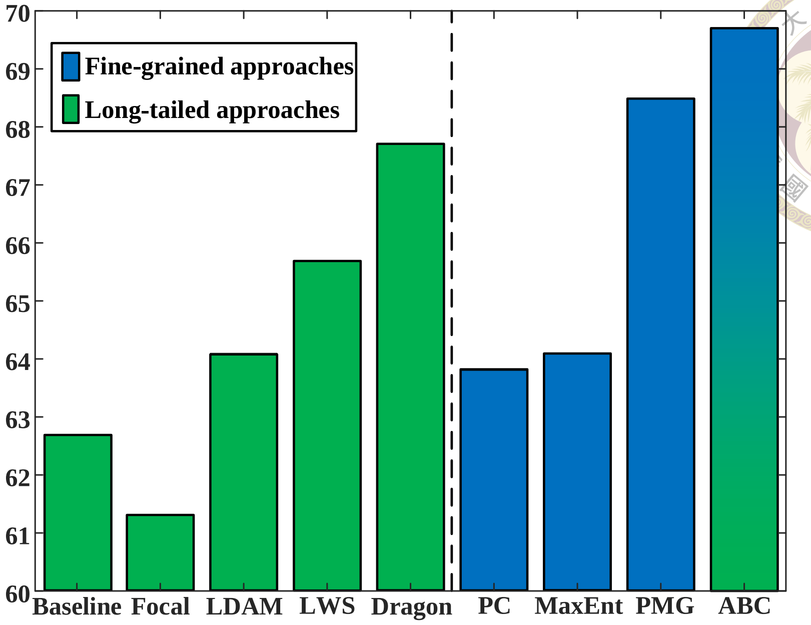


Figure 4.4: Compare the proposed ABC-Norm with other long-tailed and fine-grained approaches on CUB-LT.

consistent behavior between shallow and deep models. Given the substantial imbalance present in the ImageNet-LT with a low fine-grained factor, we set the hyper-parameters λ and τ to 0.1 and 0.5, respectively.

4.4.3 Real-world Data

Before we delve into the real-world data, let us quickly look at a small-scale one, CUB-LT, which contains both fine-grained and long-tailed factors. It is an appropriate dataset for investigating the performances among the respective approaches for fine-grained [36, 37] and for long-tailed [63, 89, 104]. As shown in Figure 4.4, PC and MaxEnt, which are proposed to account for the fine-grained factor, only show slight improvements for resolving the long-tailed issue. PMG provides a strong performance, but requires more advanced data augmentations and larger model sizes. Meanwhile, LDAM, LWS, vMF, and Dragon demonstrate that addressing the long-tailed issue can also improve performance on real-world data distributions. However, the proposed ABC-Norm significantly outperforms these approaches by explicitly tackling both the fine-grained and long-tailed challenges.

Table 4.2 shows the experimental results on the large-scale and real-world distribu-

doi:10.6342/NTU202301574

Table 4.2: Compare the ABC-Norm with other primary approaches on iNaturalist2018. The backbone model used in this experiment is vanilla ResNet-50 baseline without using additional parameters and advanced augmentation schemes.

Method	Many	Medium	Few	Total
	90 epochs			
Baseline	**72.2**	63.0	57.2	61.7
Focal [78]	-	-	-	61.1
Re-weighted	-	-	-	64.9
cRT	-	-	-	65.2
PC† [37]	70.9	64.6	59.6	62.1
MaxEnt† [38]	69.8	65.1	59.4	61.9
LDAM [9]	-	-	-	64.6
LDAM w/ DRW [9]	-	-	-	68.0
LWS [63]	65.0	66.3	65.5	65.9
LA [89]	-	-	-	66.4
ABC-Norm	66.6	68.0	68.2	68.4
ABC-Norm‡	66.5	**73.4**	**69.2**	**70.8**
	200 epochs			
Baseline	**75.7**	66.9	61.7	65.8
cRT	73.2	68.8	66.1	68.2
PC† [37]	67.8	64.2	60.2	62.8
MaxEnt† [38]	70.8	65.3	59.1	62.1
DTRG [81]	-	-	-	65.5
DTRG w/ DRW [81]	-	-	-	69.5
LWS [63]	71.0	69.8	68.8	69.5
vMF [123]	72.8	71.7	70.0	71.0
ABC-Norm‡	68.1	**73.2**	**70.4**	**71.4**

† Re-implement with the same setting as ours.

‡ Follow the data augmentation scheme in [123].

tion dataset, iNaturalist2018. The adaptive matrix A enables the ABC-Norm to emphasize the head categories but scale down the regularization effect on the tail categories. Note that our models are trained not only with the most common way of data sampling, *i.e.*, *instance-balanced sampling*, but also in an end-to-end manner. In contrast, LWS [63] learns the model in two stages, which requires the use of *class-balanced sampling*. Notwithstanding that LA [89] has the same starting point as ours, which also proposes an approach that does not require any extra parameters, strong augmentation schemes, and data sampling strategies, the ABC-Norm regularization does yield a better performance. The main advantage of ABC-Norm over LA on this real-world dataset is that the proposed ABC-Norm

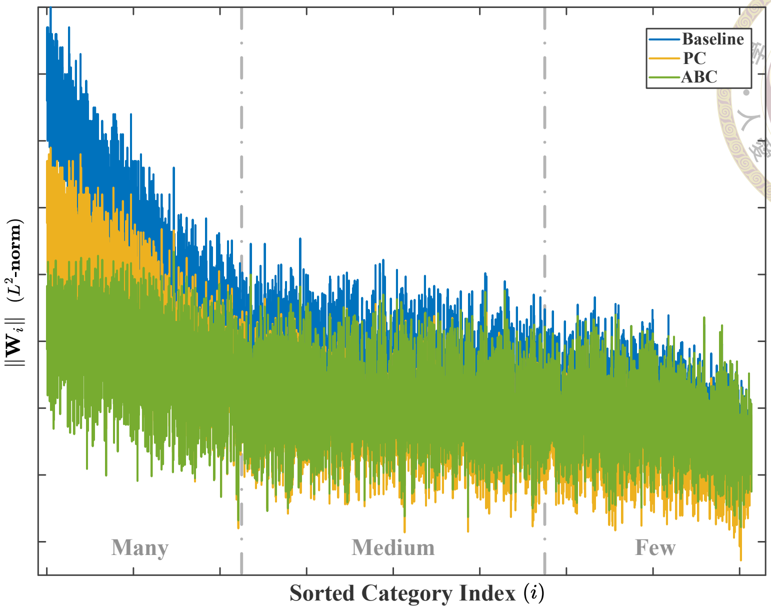


Figure 4.5: The distribution of L^2-norm weight magnitude $\|\mathbf{w}_i\|$ for baseline, PC and ABC-Norm, where $\mathbf{w}_i$ is the classifier weight vector of category i.

provides a unified solution to addressing both long-tailed and fine-grained factors.

Recall that PC [37], MaxEnt [37] and DTRG [81] are introduced to validate that proper regularization is useful for dealing with the fine-grained problem. We, however, observe that the three techniques only yield slight improvements on the real-world dataset. In fact, to properly tackle the long-tailed difficulty, DTRG [81] has adopted the DRW [8] schedule. Compared with DTRG, ABC-Norm achieves better performance without relying on additional schemes such as Mixup and DRW. (We have also reported in Table 4.2 the result of ABC-Norm using the data augmentation scheme from [123].) Overall, our ABC-Norm method provides a general and flexible approach to solving real-world classification tasks.

4.4.4 Model Analysis

We begin by evaluating the effect of regularization on the magnitude of the classifier weight $\mathbf{w}_i$ for each category i, as depicted in Figure 4.5. While the L^2-norm magnitude distribution $\|\mathbf{w}_i\|$ of the baseline method exhibits a long-tailed pattern, the proposed ABC-Norm instead produces a smoother magnitude distribution for the head categories,

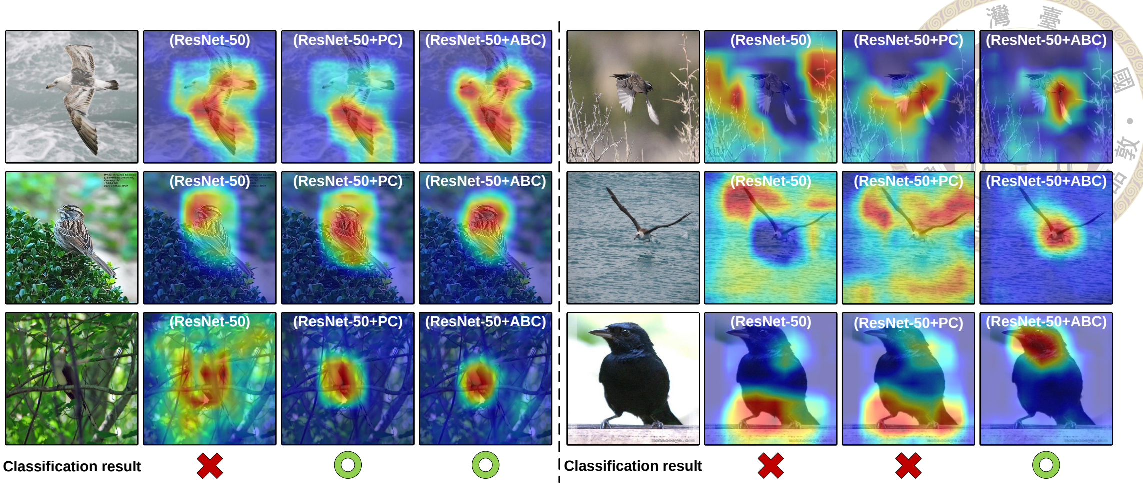


Figure 4.6: Grad-CAM heatmap visualization for six testing images. In each example, the resulting heatmap is specified by the corresponding model (ResNet-50, ResNet-50+PC, ResNet-50+ABC). Results of correct classification are marked with “ ✖ ” and otherwise, “ ◎ ”.

Table 4.3: Head-to-head comparisons among four different regularization approaches, ABC-Norm, PC, MaxEnt and DTRG, on the standard FGVC datasets CUB-200-2011 (CUB), Stanford Cars (CAR), and FGVC-Aircraft (AIR).

Model	ResNet-50			ResNeXt-50			ResNeXt-101			DenseNet-161		
	CUB	CAR	AIR	CUB	CAR	AIR	CUB	CAR	AIR	CUB	CAR	AIR
Baseline	85.5	92.7	90.3	86.3	93.1	90.9	87.3	93.5	91.6	87.5	93.4	92.7
PC [37]	87.0	92.4	90.1	87.5	93.2	91.2	88.2	93.7	92.4	88.2	93.6	92.9
MaxEnt [38]	87.2	91.9	90.3	87.6	92.8	91.3	88.2	93.4	92.5	88.3	93.3	93.0
DTRG$^-$ [81]	**88.3**	**94.8**	93.0	-	-	-	-	-	-	89.0	**94.8**	**94.0**
ABC-Norm	87.8	94.3	**93.2**	**88.1**	**94.4**	**93.3**	**88.6**	**94.5**	**93.5**	**89.2**	**94.8**	93.5

The notation $^-$ indicates the results by DTRG without using Mixup, as reported in the original paper [81].

reducing their dominance. In comparison, PC also lessens the dominance of head categories, but the distribution remains largely unchanged, indicating the persistence of the long-tailed issue.

Next, we conduct an ablation study on the iNaturalist2018 dataset to assess the impact of different batch sizes on the various regularization approaches, including ABC-Norm, PC, and MaxEnt. Figure 4.7 shows that the performance variations among different batch sizes are similar across all regularization methods as well as the baseline. This suggests that the influence of batch size stems from the use of “batch normalization” and the correlation between the performance of ABC-Norm and batch size is weak. Hence, choosing a specific batch size for ABC-Norm is generally not an issue of concern.

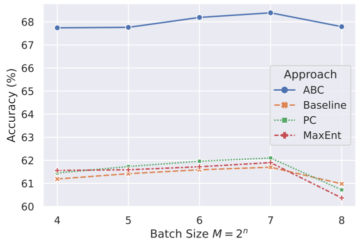


Figure 4.7: Accuracy versus batch size on iNaturalist2018 for the three regularization methods: ABC-Norm, PC and MaxEnt.

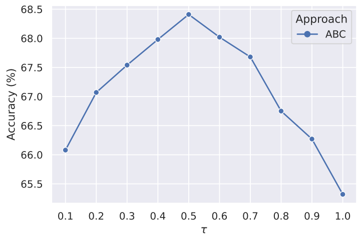


Figure 4.8: An ablation study about various τ values for the adaptive matrix A on iNaturalist2018. The sweet point for τ on the real-world dataset, iNaturalist2018, locates at 0.5.

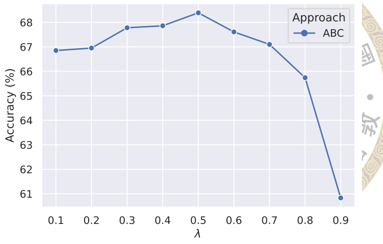


Figure 4.9: An ablation study of the regularization weight λ on iNaturalist2018. We observe that the suitable value for λ locates in the range $[0.1, 0.6]$.

The long-tailed issue often requires the selection of an appropriate value of hyperparameter τ to incorporate the statistical information embodied in the training data. Figure 4.8 shows the results of varying the τ value, where the resulting curve exhibits a downward parabolic trend from $\tau = 0.1$ to $\tau = 1.0$, with the best performance achieved at $\tau = 0.5$.

We also investigate the optimal regularization weight λ between cross-entropy loss and ABC-Norm in (4.5). Figure 4.9 displays the probing result of such search. The classification performance gradually improves as λ increases, reaching a sweet spot at $\lambda = 0.5$. Beyond this point, further increasing the λ value leads to a decrease in performance, suggesting that the suitable range of λ is $[0.1, 0.5]$.

We conclude our analysis by providing a qualitative comparison of the baseline, PC, and ABC-Norm methods using the class activation mapping (Grad-CAM) [106] on the CUB dataset. As shown in Figure 4.6, the results reveal that PC and ABC-Norm correctly predict more samples than the baseline. The redder an area is, the more significant the model's prediction is, while the bluer the area indicates the opposite. For instance, PC and ABC-Norm focus on the appropriate regions to identify the object rather than the background. Moreover, as in the right panel of Figure 4.6, ABC-Norm can correctly classify even the challenging samples that the PC and baseline fail to recognize. This is because that ABC-Norm further exploits the inter-class similarity information to ensure the resulting classifier to focus on the most discriminative parts.

Table 4.4: Accuracy versus batch size on the CUB dataset.

Batch Size	2	4	8	16	32
Baseline	80.9	84.5	85.4	85.5	85.5
PC [37]	81.1	85.8	86.7	87.0	86.9
MaxEnt [38]	**81.4**	85.9	86.9	87.2	87.0
ABC-Norm	81.2	**86.1**	**87.5**	**87.8**	**87.7**

4.4.5 More on Fine-grained

To investigate the compatibility of the proposed ABC-Norm on the FGVC datasets, we conduct experiments with different backbones against PC [37], MaxEnt [38] and DTRG [81], respectively. The backbones are chosen from shallow to deep, including ResNet-50, ResNeXt-50, ResNeXt101 and DenseNet-161. We re-implement PC and MaxEnt with the same training condition. Table 4.3 shows the head-to-head comparison; the experimental results imply that ABC-Norm outperforms both PC and MaxEnt. Compared to DTRG, although the performances are about even, the training process of ABC-Norm is simple and essentially the same as the baseline case. Moreover, we also conduct an ablation study to gauge the batch-size influence. Table 4.4 shows that the batch-size influence is still similar to that in Figure 4.7. It suggests that we only need to pay more attention to the hyper-parameters λ and τ. Among the confusion-based techniques, ABC-Norm is not only more effective in fine-grained classification than PC and MaxEnt but also valid when dealing with the long-tailed issue.

4.4.6 More on Long-tailed

Since ImageNet-LT has a low fine-grained factor but poses strong long-tailed difficulty, we perform experiments on it to confirm the effectiveness of ABC-Norm for purely long-tailed learning. Following the training formulation in [63], we decompose the training process into two stages, representation learning and classifier learning. In stage one for representation learning, the data sampling strategy is instance-balanced which can also be called a baseline. Next, the sampling strategy turns class-balanced to fine-tune the classi-

Table 4.5: Following [63], the experiment on ImageNet-LT is carried out with ResNeXt-50 and ResNeXt-152, respectively.

Method	ResNeXt-50 / ResNeXt-152			
	Many	Median	Few	Total
Baseline	**65.9/69.1**	37.5/41.4	7.7/10.4	44.4/47.8
with LWS	60.2/63.5	47.2/50.4	30.3/34.2	49.9/53.3
PC	63.9/66.9	35.5/37.3	8.8/10.2	42.8/45.1
PC + LWS	57.3/59.5	46.4/48.7	29.8/32.6	48.4/50.6
MaxEnt	63.4/66.1	35.9/37.8	8.6/10.1	42.5/44.8
MaxEnt + LWS	59.3/61.9	46.1/47.8	29.5/30.8	48.9/50.9
ABC-Norm	65.5/68.7	43.1/45.5	10.9/12.3	47.5/49.9
ABC-Norm + LWS	60.7/63.6	**49.7/51.8**	**33.1/35.5**	**51.7/54.2**

fier at stage two. Table 4.5 first shows the results based on ResNeXt-50. At stage one, with end-to-end training, the baseline, PC, and MaxEnt are prone to overfit the head categories since these methods do not take account of the imbalanced distribution of the training set. On the contrary, the results show that the ABC-Norm significantly improves and alleviates the domination problem of head categories. Furthermore, through stage two, fine-tuning the classifier with LWS can improve the representation model learned from ABC-Norm. In conclusion, tackling the long-tailed distribution with our method can learn a better representation model than PC and MaxEnt.

To further verify the robustness of the proposed ABC-Norm regularization, we re-evaluate the experiment with the same setting but using a deeper backbone, ResNeXt-152. The experimental results are also presented in Table 4.5. We see that no matter how deep or shallow the model is, ABC-Norm still achieves consistent improvements, which again confirms the compatibility of the ABC-Norm approach.

4.4.7 Additional Results

Finally, to demonstrate that it is no coincidence that ABC-Norm improves via sufficient training, we also explore the experiment on both a deep network and longer training epochs. Following the training procedure from previous work [63, 89], we apply the ABC-

doi:10.6342/NTU202301574

Table 4.6: iNaturalist2018 classification by different methods with ResNet-152, based on 90 and 200 training epochs.

Setting	90 epochs	200 epochs
Baseline	65.0	69.0
Re-weighted	68.1	69.9
PC† [37]	66.9	69.3
MaxEnt† [38]	66.6	69.2
LWS [63]	69.1	72.1
LA [89]	68.9	69.9
ABC-Norm	71.7	72.6
ABC-Norm‡	**73.8**	**74.0**

† Re-implement with the same setting as ours.
‡ Follow the data augmentation scheme in [123].

Norm to the ResNet-152 backbone and report the experimental results trained with 90 and 200 epochs on the real-world dataset, iNaturalist2018. This additional experiment is designed to justify the robustness of our method and address the concern that the ABC-Norm is effective only for a specific setting. The experimental results are shown in Table 4.6. For the same data augmentation and the complete model, the results are consistent with those of Table 4.2. Meanwhile, the other two regularization-based approaches, PC and MaxEnt, still do not perform well in this experiment, which includes both fine-grained and long-tailed difficulties in the underlying real-world dataset. With all our extensive experimental results, we demonstrate that the proposed approach, ABC-Norm, is generic and not specific.

4.5 Conclusions

We introduce Adaptive Batch Confusion Norm (ABC-Norm), a general regularization technique to tackle the challenging problem of fine-grained and long-tailed image classification. Our method is simple in design, consisting of only the cross-entropy and the ABC-Norm regularization terms. During training, the ABC-Norm regularization adaptively generates confusion for each object category and activates an adversarial-like learning mechanism, leading to improved learning efficiency and more discriminative features within regions of interest. Through experiments, we show that ABC-Norm outperforms other relevant (adversarial) regularization-based approaches, such as PC and MaxEnt, and effectively reduces overfitting in training. In future work, we plan to generalize this regularization concept to transformer-based networks and enhance its effectiveness with attention mechanisms.

doi:10.6342/NTU202301574

doi:10.6342/NTU202301574

Chapter 5 Learning with Ordering Data Distributions

5.1 Introduction

With the rapid growth of face-swapping techniques, deep forgery has become a concerned issue on social media. An effective solution to address the matter is to utilize neural network-based approaches to decide the authenticity of given images. The task of deepfake classification is usually formulated as a binary classification problem. Recent research efforts on deepfake classification have delivered saturated performances [7, 18, 109, 144, 145]. Nevertheless, owing to the impressive development of generative networks, *e.g.*, StyleGAN and diffusion models [65, 67, 102], deep forgery is no longer limited to face-to-face interchange. In particular, Shao *et al*. [107] propose a sequential facial manipulation dataset, Seq-DeepFake, in which the fake facial images are manipulated with the requested sequential constraints from the source (*e.g*., components and attributes) by StyleMapGAN [67]. Take, for example, in Figure 5.1, the annotation of "Eyebrow-Hair-Lip" indicates that the resulting facial image has been successively manipulated with the eyebrow, hair, and lip in the specified order. The sequential manipulation can be treated as a multi-label "localization" problem to decide not only which facial components have been manipulated but also what the manipulation order is. The latter task further complicates the localization scenario into a multi-label ranking problem, which poses significant challenges and opens a new frontier for deepfake-related research.

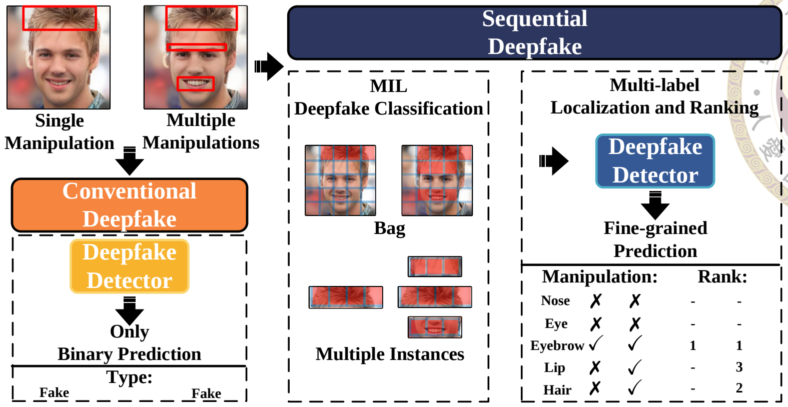


Figure 5.1: Besides the conventional deepfake setting as binary classification that simply dichotomies the images into genuine/fake, this work focuses on the subtle scenario that forged images through deepfake mechanisms may be locally manipulated by one or more than one facial component/attribute. We introduce a multi-label ranking approach to tackling the "fine-grained" deepfake task (*i.e.*, to localize the modified facial components and to identify the order of manipulations), and also develop a contrastive multi-instance learning (MIL) framework to solve the binary classification.

Detecting the sequential facial manipulations is more challenging than conventional deepfake classification. It causes that most of the existing deepfake solutions are no longer applicable. For example, the success of Face X-ray [73] is based on the observation that a fake facial image must have an essential blending operator to smooth the face boundary to make the forged image more natural during the face-swapping process. The particular method then focuses on learning how to capture the blending region from the paired source and target images. However, the tactic apparently does not work well on the sequential facial manipulation dataset, SeqDeepFake [107]. The inefficiency is caused by two main factors. First, the paired source and target information of each manipulated image in SeqDeepFake is not available. Second, the resulting classifier from adversarial learning is often highly related to the generator. Therefore, it is hard to generalize the method to distinguish the sequentially manipulated images without completely updating the generative model in [73]. Such inappropriateness to work on component-wise deepfake indeed applies to the majority of related methods, *e.g.*, [17, 18, 34, 109, 145]. After all, they are developed to solve a binary classification problem, rather than dealing with the sequential deepfake manipulation.

Aiming to establish a unified approach to deepfake detection, we decompose the underlying problem into three subtasks, including *deepfake classification*, *deepfake local-*

doi:10.6342/NTU202301574

ization, and *manipulation order*. In resolving the first subtask, we propose contrastive *multiple instance learning* (MIL) that treats an image as a bag and the spatial features as instances to tackle deepfake classification via minimizing a contrastive MIL loss. We then establish a multi-label ranking formulation to address the other two subtasks. Concerning the ability to identify which facial components have been forged, we loosely term the process as deepfake localization. In addition, it is reasonable to incorporate ranking reasoning into the stage so that the ranked list of multi-label probabilities can reflect the sequential modification order. With such, training the network model can be done via multi-task learning, and results in an effective deepfake detection model capable of accomplishing the three aforementioned tasks. We characterize our main contributions as follows.

- We decompose the general deepfake problem into three parts, *deepfake classification*, *deepfake localization*, and *manipulation order* which lead to a systematic view of solving the deepfake problem comprehensively.
- We propose a contrastive multi-instance learning formulation for binary deepfake classification. The synergy between the two learning paradigms improves the model learning effectively, and more importantly, it gives rise to a well-established concept of how to define the probability of a given image being deepfake.
- We develop a multi-label ranking approach to coupling multi-label predictions with ranking reasoning. In inference, the sequential order of deepfake manipulations can be readily obtained from the rank order of the output multi-label probabilities.
- We establish a unified approach to deepfake classification and localization, and achieve state-of-the-art performances on popular benchmark datasets.

5.2 Related work

Deepfake detection. Owing to the active development of face manipulation technology and the upsurge of people's awareness about multimedia security, more research efforts have been paid to develop all sorts of deepfake detection methods in recent years.

Deepfake detection can be categorized into two types of approaches based on the underlying data format: *image-based* [7, 18, 34, 73, 80, 109, 144, 145, 152] and *video-based* [26, 47, 74, 148]. For image-based deepfake detection, Zhu *et al.* [152] propose a two-stream architecture to enrich the face feature for detection. One is a conventional network, and the other is a 3D decomposition framework that aims to find more clues and details on the face image. Chen *et al.* [18] fuse the RGB and frequency features with a cross-attention module to learn an artifact mask decoder from the fake images. The decoder uses the source and target information from the manipulated image to generate the mask as a ground-truth label. Cao *et al.* [7] regard the detection problem as anomaly detection and utilize an encoder-decoder framework for real-fake representation learning. Liu *et al.* [80] determine the forgery image from the phase spectrum variation between the original and up-sampled images. Zhao *et al.* [144] introduce multiple attention modules to capture different discriminative locations and insert a texture enhancement block in the backbone for extracting the high-frequency features. Several other methods attempt to capture the artifacts generated by swapping faces from two images. For example, Li *et al.* [73] propose a face X-ray to find the blended region from the forgery image. Moreover, Zhao *et al.* [145] exploit the fact that forgery faces are manipulated from two different sources and propose an inconsistent image generator for supporting the classifier learning the consistency mask. Based on a similar entry point, Dong *et al.* [34] utilize the self-attention mechanism to form an identity consistency transformer for detecting a forgery image. To extend the above concepts, Shiohara *et al.* [109] introduce a self-blended framework that can learn the blended clues from the proposed augmentation technique.

For video-based deepfake detection, Cozzolino *et al.* [26] use a three-dimensional morphable model to generate the deepfake video and learn a temporal network to embed the sequence features for the video classifier. Zhou *et al.* [148] present a two-plus-one joint detection model for tackling both manipulated visual and auditory modalities.

More recently, Shao *et al.* [107] generalize the image-based deepfake detection from a binary classification problem to a multi-label classification problem. Specifically, the image is now manipulated from sequential components/attributes, dramatically increasing

doi:10.6342/NTU202301574

the detection challenge.

Multiple instance learning. The *multiple instance learning* (MIL) [32] paradigm defines a "bag" as *positive* if it contains at least one positive instance. In other words, all instances in a negative bag are assumed to be negative. A previous approach by Chen *et al*. [23] transforms each sample bag into a high-dimensional feature space and adopts the Support Vector Machine (SVM) to determine essential features and construct the classifier simultaneously. Ilse *et al*. [60] introduce MIL attention pooling that leverages neural networks to parameterize the distribution of instances in a bag to detect predefined positive instances. In medical imaging, several approaches regard MIL-related tasks on histopathology datasets as weakly supervised learning. Zhang *et al*. [141] introduce the pseudo-bag concept to enrich the sample bags to address the insufficiency of whole slide images. Furthermore, Thandiackal *et al*. [116] present ZoomMIL that utilizes a multi-level zooming mechanism to fuse multiple magnifications and reduce the computation cost.

Ranking mechanism. A ranking scheme is designed to find the optimal sorting function that can rank the sequential input. While early efforts [3, 77] propose the bitonic sorting network to solve the rank issue, techniques of current trend rely on the neural network to achieve the differential ranking operation. Petersen *et al*.first present Differentiable Sorting Networks [96] and take it as an extension by enforcing monotonicity and limiting the bound of approximation error. They subsequently introduce a differential top-k network [97] to address the multi-class problem via the ranking mechanism.

5.3 Method

We consider a general formulation of deepfake detection that the underlying photorealistic manipulations can be applied to either the whole facial region or some of the predefined facial components. For the sake of discussion, we categorize the former task as *deepfake classification* and the latter as *deepfake localization*, where in this scenario we

doi:10.6342/NTU202301574

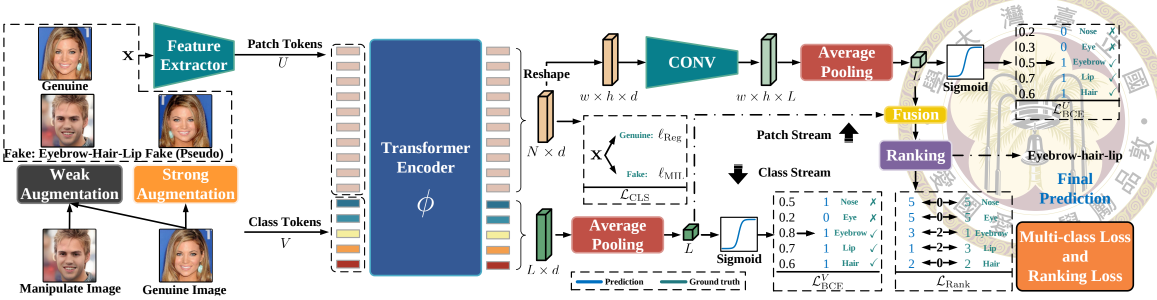


Figure 5.2: The model architecture of our method. There are two types of input tokens: *patch tokens* extracted from CNN+FPN and *learnable class tokens*. The stage of model training is driven by three loss terms: $\mathcal{L}_{\text{CLS}}$, $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{Rank}}$ to achieve contrastive multiple instance learning, multi-label localization and ranking, respectively. In the inference stage, the sequential order of deepfake manipulations can be readily obtained from the rank order of the output multi-label probabilities.

also need to recover the sequential order of the component-wise deepfake manipulations as described in [107].

Problem formulation. Suppose there are totally L facial components to which photorealistic manipulations can be applied. Since the exact order of modifying the facial components does matter, we cast the task of deepfake localization as a *multi-label ranking* problem [30]. Consider now a deepfake dataset $\mathcal{D} = \{(\mathbf{x}, Y)\}$, where $\mathbf{x}$ is an image and $Y = \{l_i\}_{i=1}^{k}$ with $k \leq L$ is an ordered subset of $\{1, 2, \ldots, L\}$, indicating that the ith ($i \leq k$) deepfake modification has been performed on the l_ith facial component. When Y is an empty set, it implies that $\mathbf{x}$ is a genuine facial image. It is convenient to generate from Y two L-dimensional vectors $\mathbf{y} = (y_i)$ and $\mathbf{r} = (r_i)$ by

$$y_i = \begin{cases} 1, & \text{if } i = l_j \in Y; \\ 0, & \text{otherwise}, \end{cases} \tag{5.1}$$

and

$$r_i = \begin{cases} j, & \text{if } i = l_j \in Y; \\ L, & \text{otherwise}, \end{cases} \tag{5.2}$$

where $\mathbf{y}$ is the standard multi-label binary vector and $\mathbf{r}$ is the corresponding rank vector. We realize the above definitions with a hands-on example. Assume that totally five facial components can be modified, *i.e.*, $L = 5$, and a deepfake image has been created by first manipulating facial component 4 and then facial component 2. Our definitions imply that

$Y = \{4, 2\}$, $\mathbf{y} = (0, 1, 0, 1, 0)$ and $\mathbf{r} = (5, 2, 5, 1, 5)$.

To train a deepfake detection model with the training data $\mathcal{D}$, we consider a CNN-transformer network, as illustrated in Figure 5.2. For each training sample $(\mathbf{x}, Y)$, the CNN+FPN module transforms $\mathbf{x}$ into feature maps of size $\mathbb{R}^{w \times h \times d}$, which can be reshaped and row-wise ℓ_2-normalized into a token vector $T \in \mathbb{R}^{N \times d}$ and $N = w \times h$ is the spatial size.

We then form two vectors of tokens, including the patch tokens $U \in \mathbb{R}^{N \times d}$ and the learnable class tokens, $V \in \mathbb{R}^{L \times d}$. The two sets of tokens are passed through the transformer encoder ϕ, which carries out self attention to correlate their features by

$$U \xrightarrow{\phi} \widetilde{U} \in \mathbb{R}^{N \times d}, \quad V \xrightarrow{\phi} \widetilde{V} \in \mathbb{R}^{L \times d}. \tag{5.3}$$

We compute the similarity values of each patch token to all other tokens by

$$S = \max(\widetilde{U}\widetilde{U}^{\top}, 0) \in \mathbb{R}^{N \times N}, \tag{5.4}$$

where S is rectified into a nonnegative matrix such that all of its elements are in $[0, 1]$. Since the similarity matrix is symmetric and we concern only the correlations of each token to all other tokens, it is sufficient to focus on the upper triangular part of S, excluding those in the diagonal. We arrange thees entries of interest in an ascending order of similarity value and denote them by

$$\mathbf{u} = (u_1, u_2, \ldots, u_n), \tag{5.5}$$

where $n = N(N-1)/2$ corresponds to the size of upper triangle of S.

MIL deepfake classification. With the sorted list $\mathbf{u}$ of similarity responses between patch tokens, we can consider the task of deepfake detection from the multiple instance learning (MIL) viewpoint. That is, we consider a face image $\mathbf{x}$ as a bag and the positive label 1 indicates that $\mathbf{x}$ is indeed fabricated as a deepfake one. In terms of the elements in $\mathbf{u}$, if $\mathbf{x}$ is a deepfake image, we expect to uncover that there exists at least one u_i (starting from the front end of $\mathbf{u}$) with a small value close to 0. On the other hand, a negative bag (*i.e.*, $\mathbf{x}$ is not a deepfake image) implies all u_i are close to 1. To incorporate the above observations into the model learning process, we introduce a *contrastive* formulation to realize the MIL concept for deepfake detection. Assume that a deepfake image $\mathbf{x}$ results in the k smallest similarity responses in the front end of the sorted list $\mathbf{u}$. We propose to compute its probability of being deepfake by contrasting the average responses from the positive and negative distributions:

$$P(\mathbf{x}; k) = \frac{\exp(u^{+}(k)/\tau)}{\exp(u^{+}(k)/\tau) + \exp(u^{-}(k)/\tau)} \tag{5.6}$$

where τ is the temperature parameter,

$$u^{+}(k) = \frac{1}{k}\Big(a - \sum\nolimits_{i=1}^{k} u_i\Big), \tag{5.7}$$

$$u^{-}(k) = \frac{1}{n-k}\Big(a - \sum\nolimits_{i=k+1}^{n} u_i\Big), \tag{5.8}$$

and a is a scalar that is set to 1 in our implementation. The contrastive ratio in (5.6) can be used to predict the probability of a given image $\mathbf{x}$ being a deepfake one by

$$P(\mathbf{x}) = \max_{1 \le k \le n} P(\mathbf{x}; k), \tag{5.9}$$

where the reason for seeking a maximum is supported by the existence of at least one positive/fake instance. We thus define the contrastive MIL loss for each $(\mathbf{x}, Y) \in \mathcal{D}$ as

$$\ell_{\text{MIL}}(\mathbf{x}) = -J(Y)\log P(\mathbf{x}) - (1 - J(Y))\log(1 - P(\mathbf{x})) \tag{5.10}$$

where $J(Y) = 1$ if a sample $(\mathbf{x}, Y)$ is a deepfake image, and 0, otherwise. In addition, for an authentic image $\mathbf{x}$, it is reasonable to expect that all the similarity responses u_i should be close to 1. The useful observation motivates the inclusion of the following regularization loss:

$$\ell_{\text{Reg}}(\mathbf{x}) = \sum_{i=1}^{n} \|1 - u_i\|_2, \tag{5.11}$$

doi:10.6342/NTU202301574

to ensure proper similarity responses for a real $\mathbf{x}$. We can then express the loss function for deepfake classification as

$$\mathcal{L}_{\text{CLS}} = \sum_{(\mathbf{x},Y)\in\mathcal{D}} \ell_{\text{MIL}}(\mathbf{x}) + (1 - J(Y))\, \ell_{\text{Reg}}(\mathbf{x}). \tag{5.12}$$

Multi-label localization and ranking. The contrastive MIL formulation leads to a new loss term specified in (5.12) for learning deepfake classification. To extend our method for deepfake localization, we consider multi-label ranking to uncover which facial components have been modified as well as the underlying order of manipulations. The Transformer encoder ϕ generates, for each sample $(\mathbf{x}, Y)$, two sets of features from the patch tokens, $U \in \mathbb{R}^{N\times d}$ and the class tokens, $V \in \mathbb{R}^{L\times d}$ as in (5.3). Our network model applies convolutions to U and then carries out average pooling to obtain the patch-token logits $\mathbf{f}^U = (f_i^U) \in \mathbb{R}^L$. In a similar way, we have the class-token logits $\mathbf{f}^V = (f_i^V) \in \mathbb{R}^L$. By independently applying a sigmoid function σ to each logit, we obtain two sets of multi-label predictions as

$$P_i^{\mathcal{X}}(\mathbf{x}) = \sigma(f_i^{\mathcal{X}}) \in [0,1], \quad i = 1,\ldots,L, \tag{5.13}$$

where $\mathcal{X}$ can be replaced by U or V to respectively imply that the predictions are based on the features from patch tokens or class tokens. Recall that the ground-truth label vector Y yields the corresponding multi-label binary vector $\mathbf{y} = (y_i)$ and the rank vector $\mathbf{r} = (r_i)$, which are both L-dimensional. With the multi-label predictions given by (5.13), we define the multi-label BCE loss as

$$\mathcal{L}_{\text{BCE}}^{\mathcal{X}} = \sum_{(\mathbf{x},Y)\in\mathcal{D}} \mathbf{1} \cdot \ell^{\mathcal{X}}(\mathbf{x}), \tag{5.14}$$

where "·" denotes inner product, **1** is all-ones vector, and the ith element of $\ell^{\mathcal{X}}(\mathbf{x}) \in \mathbb{R}^L$ is defined by

$$\ell_i^{\mathcal{X}}(\mathbf{x}) = -y_i \log P_i^{\mathcal{X}}(\mathbf{x}) - (1 - y_i)\log(1 - P_i^{\mathcal{X}}(\mathbf{x})). \tag{5.15}$$

It is worth mentioning that both the multi-label predictions P^U and P^V from (5.13) are computed only during the training stage. Including the two loss terms $\mathcal{L}_{\text{BCE}}^U$ and $\mathcal{L}_{\text{BCE}}^V$

doi:10.6342/NTU202301574

helps regulate the model training and more critically align the class-wise logits from the patch-token and class-token streams.

We are now ready to solve the multi-label ranking problem. To begin with, we average the patch-token and the class-token logits to obtain $\mathbf{f} = (f_i) = (\mathbf{f}^U + \mathbf{f}^V)/2$. The fusion between the two streams gives rise to multi-label predictions $\{P_i(\mathbf{x})\}_{i=1}^L$, analogous to those from (5.13). The main idea behind our formulation is as follows: by constructing a rank-aware loss term, the learned network model is expected to output multi-label predictions $\{P_i(\mathbf{x})\}_{i=1}^L$ that respect the rank order $\mathbf{r} = (r_i)$, implied by the given sample $(\mathbf{x}, Y) \in \mathcal{D}$. In other words, if $i, j \in Y$ and $r_i < r_j$ (*i.e.*, facial component i is modified before facial component j is manipulated), the network is trained to make multi-label predictions with $P_i(\mathbf{x}) > P_j(\mathbf{x})$. To this end, we design the following loss term for tackling multi-label ranking,

$$\mathcal{L}_{\text{Rank}} = \sum_{(\mathbf{x},Y)\in\mathcal{D}} \mathbf{w}(\mathbf{r}, \{P_i(\mathbf{x})\}) \cdot \ell(\mathbf{x}), \tag{5.16}$$

where $\ell(\mathbf{x}) \in \mathbb{R}^L$ is analogously defined as in (5.15) but with multi-label predictions $\{P_i(\mathbf{x})\}_{i=1}^L$ based on the fused logits $\mathbf{f}$. To complete the explanation of (5.16), it remains to elaborate how the rank-aware weight vector $\mathbf{w}(\mathbf{r}, \{P_i(\mathbf{x})\})$ is designed. As our aim to preserve the rank order $\mathbf{r}$ in the multi-label predictions $\{P_i(\mathbf{x})\}_{i=1}^L$, we let $\mathbf{o} = (o_i) \in \mathbb{R}^L$ to encode the rank order (non-increasing order of probability values) among the multi-label predictions. We then define the weight vector $\mathbf{w}(\mathbf{r}, \{P_i(\mathbf{x})\}) = (w_i) \in \mathbb{R}^L$ by

$$w_i = \begin{cases} \alpha, & \text{if } i \notin Y \ \wedge \ r_i > |Y|; \\ \alpha \times |o_i - r_i|, & \text{otherwise,} \end{cases} \tag{5.17}$$

where α is a hyperparameter to our method. We now justify the definition of $\mathbf{w}$. Given a deepfake sample $(\mathbf{x}, Y) \in \mathcal{D}$, there are $|Y| \leq L$ components that have been modified. The first condition in (5.17) indicates that facial component i is genuine and its corresponding prediction $P_i(\mathbf{x})$ is not among the $|Y|$ largest outputs of $\{P_i(\mathbf{x})\}_{i=1}^L$. Such an outcome is preferable, and thus w_i is uniformly set to α. The second condition includes two scenarios. The first is that $i \notin Y$ and $r_i \leq |Y|$. This implies that the network model predicts a high-rank deepfake probability to a genuine facial component, which should be penalized with

doi:10.6342/NTU202301574

$\alpha \times |o_i - L|$. (Note that from (5.2), when $i \notin Y$, we set $r_i = L$.) The second scenario concerns the case that $i \in Y$, *i.e.*, facial component i has been changed. We thus formulate the definition of w_i to enforce reducing the difference between o_i and r_i. We conclude that by adding $\mathcal{L}_{\text{Rank}}$ to our formulation, the learned network model can output multi-label predictions $\{P_i(\mathbf{x})\}_{i=1}^{L}$ to detect which facial components have been manipulated, and also the order of modifications, which is implied by the resulting order of probability magnitudes.

Total loss. To train the proposed network model for simultaneously carrying out deepfake classification and localization, our method considers the following total loss:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CLS}} + \lambda_1 \, \mathcal{L}_{\text{BCE}} + \lambda_2 \, \mathcal{L}_{\text{Rank}}, \tag{5.18}$$

where λ_1 and λ_2 are parameters to weigh the effects of specific loss terms, and $\mathcal{L}_{\text{BCE}} = \mathcal{L}_{\text{BCE}}^{U} + \mathcal{L}_{\text{BCE}}^{V}$. Note that the two sets of multi-label probability predictions $\{P_i^U\}$ and $\{P_i^V\}$ are computed only in the training stage so that $\mathcal{L}_{\text{BCE}}^{U}$ and $\mathcal{L}_{\text{BCE}}^{V}$ can be utilized to achieve effective model training. In inference, the multi-label prediction is provided solely from the $\mathcal{L}_{\text{Rank}}$ head, as shown in Figure 5.2.

Finally, we emphasize that the proposed approach provides a unified solution to the deepfake problem. When dealing with a classical task of binary deepfake classification, it is convenient to exclude the $\mathcal{L}_{\text{Rank}}$ term from the total loss in (5.18) and simply set the number of learnable class tokens to one to achieve binary classification.

Table 5.1: The experimental results with multi-label and ranking scenarios on the Seq-FaceComp and Seq-FaceAttr datasets.

Method	Backbone	Seq-FaceComp Acc.		Seq-FaceAttr Acc.	
		Multi-label (%)	Ranking (%)	Multi-label (%)	Ranking (%)
Baseline	ResNet-34 [53]	78.32	69.66	85.14	66.99
DETR [10]		-	69.87	-	67.93
SeqFakeFormer [107]		-	72.13	-	67.99
Ours		**81.24** ↑ 2.92	**72.76** ↑ 3.10	**86.38** ↑ 1.24	**68.53** ↑ 1.54
Baseline	ResNet-50 [53]	79.54	69.75	88.23	66.66
DETR [10]		-	69.75	-	67.62
SeqFakeFormer [107]		-	72.65	-	68.86
Ours		**82.77** ↑ 3.23	**73.34** ↑ 3.59	**90.12** ↑ 1.89	**69.28** ↑ 2.62
Ours	Swin [82]	**83.01** ↑ 3.47	**73.52** ↑ 3.87	**90.43** ↑ 2.20	**69.75** ↑ 3.09

5.4 Experimental results

We report comparisons with other techniques and experimental results on the sequential deepfake dataset [107], the multi-label scenario and traditional binary deepfake classification. A comprehensive ablation study is also provided to validate the effects of the key elements in our method.

Sequential deepfake datasets. The Seq-DeepFake dataset, introduced in a recent study [107], comprises two collections of sequential image manipulations, namely *sequential facial components manipulation* and *sequential facial attributes manipulation*. To simplify the notation, we refer the two as Seq-FaceComp and Seq-FaceAttr, respectively. In Seq-FaceComp, facial components are transplanted from a source image to a target image, resulting in manipulated images that exhibit distinct face components and orders. This dataset contains 35,166 facial images, including both manipulated and genuine images, annotated with 28 different manipulation sequences. The proportion of manipulation sequence lengths ranges from 1 to 5, with percentages of approximately 20.48%, 20.06%, 18.62%, 20.88%, and 19.96%, respectively. In contrast, Seq-FaceAttr directly modifies specific attributes on the target face without relying on source images. It consists of $49,920$ facial images, each of which is annotated with one of the 26 manipulation sequence types. And the distribution of each sequence length is balanced. Notably, both Seq-FaceComp and Seq-FaceAttr have a maximum sequence length of 5, denoted as $L = 5$ in the proposed formulation described in Section 2.3. Moreover, we can also evaluate the multi-label scenario on Seq-FaceComp and Seq-FaceAttr without the ordering factor.

Binary deepfake datasets. Deepfake detection has several benchmark datasets available, including FaceForensics++ (FF++) [103], Celeb-DF (CDF) [76], WildDeepfake (WDF) [153], DeepFakeDetection (DFD) [103], and DeepFake Detection Challenge [33] (DFDC). These datasets have been extensively employed to investigate the binary classification problem. FF++ is the most commonly used dataset, comprising four manipulation techniques with 1,000 videos for each. CDF uses an improved deepfake algorithm that

doi:10.6342/NTU202301574

generates 5,639 fake videos and 590 genuine videos. WDF is a real-world dataset with 3,509 fake and 3,805 genuine face sequences. DFD offers 1,000 deepfake videos. DFDC is a large-scale dataset that contains 2,500 genuine and 2,500 fake videos in the public test set. Our approach is readily applicable to the binary deepfake classification when $\mathcal{L}_{\text{Rank}}$ is excluded from (5.18) to form the total training loss, and the number of learnable class tokens is reduced to one.

Table 5.2: The experimental results with intra-testing and cross-testing. The model for cross-testing is only trained on the FF++ dataset.

Method	Backbone	Intra-testing AUC		Cross-testing (Train on FF++ only) AUC			
		FF++ (%)	CDF (%)	CDF (%)	WDF (%)	DFDC (%)	DFD (%)
Baseline	Xception [25]	96.30	99.73	61.80	62.72	48.98	87.86
Baseline	EfficientNet-B4 [114]	99.70	99.81	64.29	63.83	-	-
Multi-Att [144]	EfficientNet-B4 [114]	99.29	99.94	67.44	59.74	-	-
SPSL [80]	Xception [25]	96.91	-	76.88	-	66.16	-
RECCE [7]	Xception [25]	99.32	99.94	68.71	64.31	69.06	-
Face X-Ray [73]	Xception [25]	99.17	-	80.58	-	**80.92**	95.40
LRL [18]	Xception [25]	99.46	-	78.26	-	76.53	89.24
SBIs [109]	EfficientNet-B4 [114]	99.64	93.74	**93.18**	-	72.42	97.56
SBIs* [109]	Swin [82]	99.72	95.68	89.12	70.56	71.08	97.34
Ours	Swin [82]	**99.82** ↑ 3.52	**99.98** ↑ 0.25	91.56 ↑ 29.76	**73.41** ↑ 10.69	75.17 ↑ 26.19	**97.88** ↑ 10.02

Implementation details. To ensure a fair comparison with SeqFakeFormer [107] on the problem of sequential facial manipulations, we implement our method by adopting ResNet-34 and ResNet-50 [53] as the CNN backbone for generating features. We then transform the spatial features into a sequential form, represented as tokens, and concatenate them with L learnable class tokens to form the input to a 1-layer transformer. The model is trained using the described loss function in (5.18), with hyper-parameters λ_1, λ_2, τ, and α set to 1, 1, 0.2, and 1, respectively. The model is trained for 200 epochs using a cosine annealing schedule, with the initial learning rate set to 0.00025 and decayed to 0. Moreover, we also provide the experimental results with Swin [82], a transformer architecture, as the feature extractor backbone for the subsequent experiments. To train a model based on Swin, we follow the training settings described in [82], but increase the number of training epochs to 400.

For the conventional deepfake classification task, we begin by extracting facial images from the videos using RetinaFace [28] and resizing them to 384×384. The training process is similar to that of the sequential facial manipulation scenario with Swin as the backbone, except that we set $L = 1$ and $\lambda_2 = 0$. These minor adjustments demonstrate

doi:10.6342/NTU202301574

the flexibility and versatility of our proposed approach in Section 2.3 for various tasks.

It is imperative to consider that the computation scale of n is roughly $\mathcal{O}(N^4)$, where N denotes the size of the feature map. Therefore, a brute-force search of k from 1 to n in (5.9) and (5.10) can have a significant impact on training speed and efficiency. To overcome this limitation, we adopt a strategy where we uniformly sample the k values from 1 to n with 100 points, rather than conducting an exhaustive search. This approach improves the training efficiency and also allows us to achieve our multi-instance learning goal.

Data augmentation. Inspired by the concept of SBIs [109] for enhancing model robustness and generalization, we propose two augmentation techniques. The first technique, referred to as strong augmentation, involves creating a pseudo fake image that simulates a fake label from a genuine face. To achieve this, we extract the facial landmarks from the real image and perturb them to generate a similar but counterfeit face. The second technique, referred to as weak augmentation, is a widely-used approach in image classification training such as horizontal flip, random crop, color jitter, and Gaussian blur. To incorporate strong and weak augmentation into model training, we first apply the strong augmentation method to produce a pseudo fake image from the original genuine image. Then, we randomly select a fake image from the dataset and combine it with the genuine and pseudo fake images to ensure their comparable quantities.

5.4.1 Comparison

Sequential deepfake manipulation. In the sequential facial manipulation problem, the primary objective is to compare our method with SeqFakeFormer [107]. The SeqFakeFormer model employs a combination of CNN and transformer models, along with an auto-regressive mechanism to address the sequential challenge. In contrast, we introduce a ranking mechanism for the multi-label scenario instead of an auto-regressive mechanism, which results in our model being more efficient during training and inference. To evaluate the performance of the proposed approach, we adopt the fixed accuracy (Fixed-Acc) [107]

doi:10.6342/NTU202301574

metric on Seq-FaceComp and Seq-FaceAttr datasets. The Fixed-Acc metric computes the accuracy between the prediction and annotation sequences while considering their rank-wise dependencies. The performance comparison of our approach with baseline, DETR, and SeqFakeFormer is presented in Table 5.1. The proposed approach is practical and effective for addressing sequential deepfake issues and has demonstrated better performance than the aforementioned methods.

Multi-label deepfake scenario. In real-world applications, identifying the forged components/attributes of a manipulated facial image can often be more crucial than determining the ordering of the manipulations. To address this, we simplify the sequential facial manipulation problem into a multi-label scenario. The performance of the multi-label classification approach can be considered as an upper bound for the sequential problem. Therefore, in Table 5.1, we present experimental results for both the baseline and ours in the multi-label column. Without the ordering issue, the performance improvement can be attributed to the proposed data augmentation strategy and the contrastive MIL loss $\mathcal{L}_{\text{CLS}}$.

Binary deepfake classification. In binary deepfake classification, we evaluate the proposed approach using both intra-testing and cross-testing scenarios. Recent research on deepfake classification can be classified into two groups. The first group focuses solely on classification [10, 80, 144], utilizing only genuine and fake annotations. The second group has no training limitations [34, 73, 109, 145], and many researchers have incorporated an adversarial learning mechanism to enrich the fake samples, thereby strengthening the classifier. In our work, we combine the advantages of both groups, utilizing the proposed data augmentation strategy with an end-to-end training approach.

Intra-testing. The process of intra-testing involves training and testing a model on the same dataset. Table 5.2 with “Intra-testing” column displays that most approaches have successfully tackled the deepfake classification problem, with even baseline models such as Xception and EfficientNet-B4 exhibiting high performance. While the proposed method achieves the best performance, the improvement is marginal. As previously discussed in the introduction, the standard classification scenario is approaching saturation. Hence, the

doi:10.6342/NTU202301574

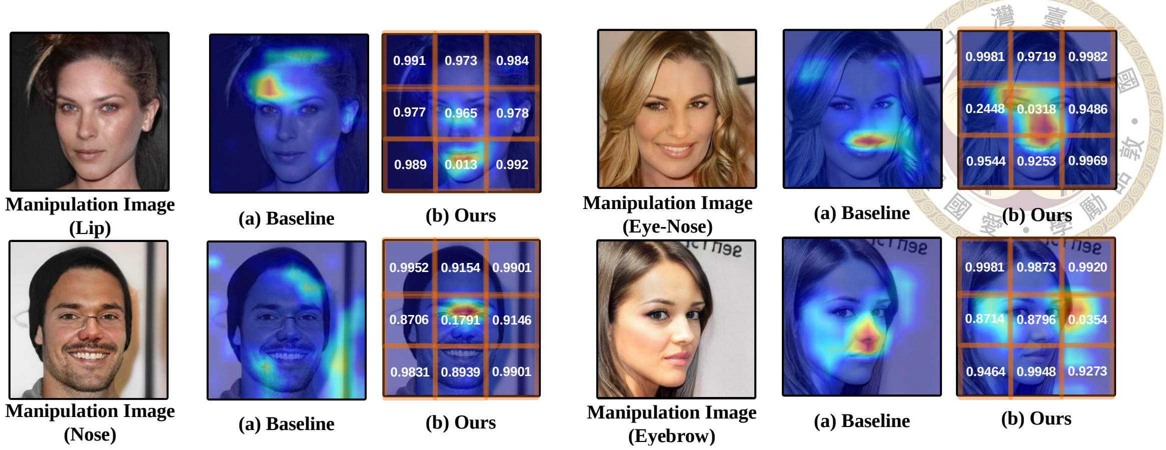


Figure 5.3: Qualitative results by Grad-CAM [106] between baseline and the proposed approach. Four test images from the Seq-FaceComp with "Lip", "Nose", "Eye-Nose" and "Eyebrow" manipulation. (a) Although the heatmap from the baseline has noticed the accurate region sporadically, it still has a gap to improve. (b) The heatmap from the proposed approach has successfully focused on the manipulation region.

main challenge of deepfake classification has transferred to the cross-testing scenario.

Cross-testing. In the cross-testing scenario, the model is exclusively trained on the FF++ dataset, as is typical in the deepfake setting. It is then assessed on the test sets of Celeb-DF, WDF, DFDC, and DFD. The experimental findings are reported in the "Cross-testing" column of Table 5.2. The proposed approach demonstrates promising results. In the interest of equitable comparison, we have also implemented the results of SBIs with Swin backbone, identified as SBIs*. Our results show significant improvement, particularly on the DFDC dataset, due to the augmentation strategy and $\mathcal{L}_{\text{CLS}}$. Consequently, exploiting the fine-grained information between each patch token is highly significant.

5.4.2 Analysis and discussion

In this section, we present an ablation study of the proposed approach in Table 5.3. It is worth noting that, unlike the baseline presented in Table 5.1, the ResNet-50 model with $\mathcal{L}^{U}_{\text{BCE}}$ in Table 5.3 generates only multi-label predictions. These predictions are subsequently sorted based on the probability of each component for evaluation. Without the proposed $\mathcal{L}_{\text{Rank}}$, although the multi-label performance achieved by ResNet-50 in Table 5.1 is commendable, the Fixed-Acc metric declines significantly due to the incorrect order of the generated sequence. Hence, the proposed $\mathcal{L}_{\text{Rank}}$ is crucial in transforming multi-label

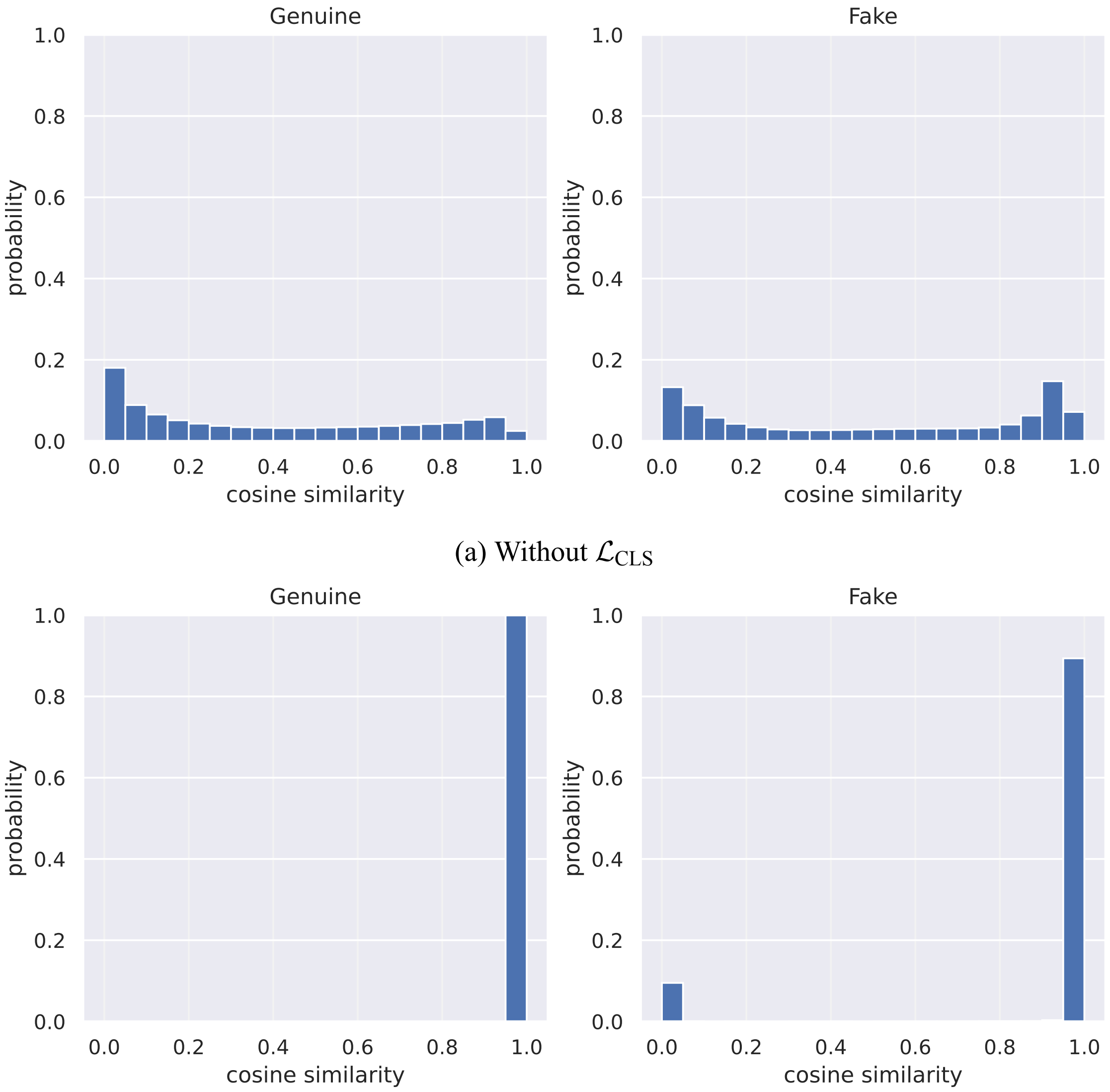
Table 5.3: An ablation study of the proposed losses on the Seq-FaceComp with multi-label ranking setting (ResNet-50).

Model	$\mathcal{L}^{U}_{\text{BCE}}$	$\mathcal{L}^{V}_{\text{BCE}}$	$\mathcal{L}^{U}_{\text{CLS}}$	$\mathcal{L}^{U}_{\text{Rank}}$	Seq-FaceComp Acc. Ranking (%)
I	✓				51.22
II	✓		✓		53.14
III	✓			✓	71.12
IV	✓		✓	✓	72.18
V		✓	✓	✓	71.64
VI	✓	✓	✓	✓	**73.34**

predictions into sequence predictions. Additionally, the proposed contrastive MIL loss, $\mathcal{L}_{\text{CLS}}$, significantly improves the performance of the model.

We proceed to present qualitative results using Grad-CAM [106] on Seq-FaceComp, as shown in Figure 5.3. The heatmaps are generated by backpropagating the "Eyebrow" and "Eye-Nose" logits. As a result of the contrastive MIL and ranking mechanism, Figure 5.3 (b) displays a more focused and accurate heatmap than the baseline. In addition, we provide the mean self-similarity values in (5.4) for each area to highlight the effect of $\mathcal{L}_{\text{CLS}}$. As expected, the patch with a lower similarity value compared to others indicates the location of the manipulation region.

Lastly, we evaluate the effectiveness of the proposed contrastive MIL loss, $\mathcal{L}_{\text{CLS}}$. To illustrate, we present in Figure 5.4 a histogram of the average distribution $\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} \mathbf{u}$ on the FF++ test set. Figure 5.4(a) depicts a phenomenon where the classifier can accurately distinguish genuine and fake facial images from a small variation distribution, despite it being challenging for humans to do so. However, with the introduction of the proposed $\mathcal{L}_{\text{CLS}}$, we are able to clearly define and simplify the distribution between genuine and fake. When looking at Figure 5.4 (b) from an alternative perspective, we note that a fake facial image typically results from two genuine facial images. As a result, the affected regions often appear at the facial boundary or composite parts, while the inner and outer face regions remain genuine. Consequently, the proportion of these affected regions is small compared to the full image. This phenomenon is consistent with the hypothesis of the MIL viewpoint that a fake image exists in minimal k points, where $k \ll n$.



(a) Without $\mathcal{L}_{\text{CLS}}$

(b) With $\mathcal{L}_{\text{CLS}}$

Figure 5.4: The histogram of averaged distribution $\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} \mathbf{u}$. (a) The histograms from the baseline are like an "U" shape, no matter whether the images are genuine or fake. (b) With the contrastive MIL loss $\mathcal{L}_{\text{CLS}}$, we regularize the $\mathbf{u}$ close to 1 in genuine images and encourage the k values from $\mathbf{u}$ to approaching 0 in fake images.

5.5 Conclusion

We present a unified approach for simultaneously addressing sequential deepfake manipulation and binary deepfake classification. To achieve this, we systematically decompose the general deepfake problem into three parts: deepfake classification, deepfake localization, and manipulation order. Our method introduces novel contrastive MIL learning and multi-label ranking to address the classification and sequential manipulation aspects, respectively. The extended experimental results demonstrate the effectiveness and flexibility of the proposed formulation in solving the various deepfake tasks. The provided analyses are also reasonable to support the usefulness of our method.

doi:10.6342/NTU202301574

Chapter 6 Conclusion

In conclusion, this thesis has made significant contributions to the field of representation learning in the context of challenging classification problems. The research efforts have focused on addressing various data distributions and proposing novel techniques to enhance representation quality and classification performance.

The thesis begins by introducing representation learning as a powerful approach for tackling complex classification problems. It highlights the importance of learning effective representations that can capture and encode essential features from diverse data sources.

To address the multi-instance data distribution, an attention mechanism equipped with a query has been proposed. This mechanism enables the representation of a bag of instances, considering the relationships and dependencies among the instances. The approach has demonstrated promising results in accurately representing multi-instance data and achieving improved classification performance.

To improve the training efficiency of self-supervised learning in the unlabeled data distribution, a decoupled contrastive learning framework has been introduced. This framework enhances the learning process by decoupling the positive and negative samples, leading to more efficient and effective representation learning from unlabeled data.

In the context of real-world data distribution, a regularization term called ABC-Norm has been proposed. This term enhances the reliability of representations by incorporating fine-grained and long-tailed issues often encountered in real-world datasets. The ABC-Norm regularization contributes to more robust representations, resulting in improved

doi:10.6342/NTU202301574

classification performance on real-world data.

Furthermore, to tackle the ordering data distribution, a multi-label ranking objective combined with a contrastive multi-instance scenario has been introduced. This approach effectively addresses the challenges associated with deepfake images, which contain multiple manipulated components with ordering issues. The proposed objective facilitates accurate representation learning for deepfake images, enabling reliable identification and classification.

Collectively, the contributions of this thesis demonstrate the importance and effectiveness of representation learning techniques in addressing challenging classification problems. The proposed attention mechanism, decoupled contrastive learning, ABC-Norm regularization, and multi-label ranking objective offer valuable insights and solutions for different data distributions. The findings from this research advance the field of representation learning, providing practical tools and methodologies for improving classification performance.

In conclusion, this thesis provides a comprehensive exploration of representation learning approaches and their applicability in addressing challenging classification problems. The proposed methodologies contribute to the existing body of knowledge and open new avenues for future research in representation learning and its applications.

doi:10.6342/NTU202301574

References

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[2] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. CoRR, 2021.

[3] Kenneth E. Batcher. Sorting networks and their applications. In American Federation of Information Processing Societies: AFIPS Conference Proceedings, pages 307–314, 1968.

[4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In Int. Conf. Machine Learning (ICML), 2018.

[5] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 2011–2018, 2014.

[6] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Rank-loss support instance machines for miml instance annotation. In Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining, pages 534–542, 2012.

[7] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang

Yang. End-to-end reconstruction-classification learning for face forgery detection. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 4103–4112, 2022.

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 1565–1576, 2019.

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 1567–1578, 2019.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Eur. Conf. Comput. Vis. (ECCV), pages 213–229, 2020.

[11] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Eur. Conf. Comput. Vis. (ECCV), 2018.

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. CoRR, 2021.

[14] Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The devil is in the channels: Mutual-channel loss for fine-grained image classification. IEEE Trans. Image Process. (TIP), pages 4683–4695, 2020.

[15] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your ”flamingo” is my ”bird”: Fine-grained, or not. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 11476–11485, 2021.

[16] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res., pages 321–357, 2002.

[17] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deep-fake detection. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 18689–18698, 2022.

[18] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In AAAI, pages 1081–1088, 2021.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In Int. Conf. Machine Learning (ICML), 2020.

[20] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. CoRR, 2020.

[21] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2021.

[22] Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI), pages 1931–1947, 2006.

[23] Yixin Chen, Jinbo Bi, and James Ze Wang. MILES: multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI), pages 1931–1947, 2006.

[24] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5157–5166, 2019.

[25] François Chollet. Xception: Deep learning with depthwise separable convolutions. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 1800–1807, 2017.

[26] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In Int. Conf. Comput. Vis. (ICCV), pages 15088–15097, 2021.

[27] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 9268–9277, 2019.

[28] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5202–5211, 2020.

[29] Don Dennis, Chirag Pabbaraju, Harsha Vardhan Simhadri, and Prateek Jain. Multiple instance learning for efficient sequential data classification on resource-constrained devices. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 10976–10987, 2018.

[30] Lihi Dery. Multi-label ranking: Mining multi-label and label ranking data. CoRR, 2021.

[31] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, pages 31–71, 1997.

[32] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell., pages 31–71, 1997.

[33] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton-Ferrer. The deepfake detection challenge (DFDC) preview dataset. CoRR, 2019.

[34] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities from deep-

doi:10.6342/NTU202301574

fake with identity consistency transformer. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 9458–9468, 2022.

[35] Chris Drummond, Robert C Holte, et al. C4. 5, Class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Workshop on learning from imbalanced datasets II, pages 1–8, 2003.

[36] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In Eur. Conf. Comput. Vis. (ECCV), pages 153–168, 2020.

[37] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Pairwise confusion for fine-grained visual classification. In Eur. Conf. Comput. Vis. (ECCV), pages 70–86, 2018.

[38] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine grained classification. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 635–645, 2018.

[39] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 9588–9597, 2021.

[40] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In Int. Conf. Machine Learning (ICML), 2021.

[41] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 961–970, 2015.

[42] Ji Feng and Zhi-Hua Zhou. Deep miml network. In AAAI, 2017.

doi:10.6342/NTU202301574

[43] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 4438–4446, 2017.

[44] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 3034–3043, 2019.

[45] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In Int. Conf. Learn. Represent. (ICLR), 2018.

[46] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[47] Zhihao Gu, Taiping Yao, Yang Chen, Shouhong Ding, and Lizhuang Ma. Hierarchical contrastive inconsistency learning for deepfake video detection. In Eur. Conf. Comput. Vis. (ECCV), pages 596–613, 2022.

[48] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5356–5364, 2019.

[49] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2006.

[50] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new oversampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping (Steven) Zhang, and Guang-Bin Huang, editors, ICIC, pages 878–887, 2005.

[51] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2020.

[52] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In Int. Conf. Comput. Vis. (ICCV), pages 4918–4927, 2019.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 770–778, 2016.

[54] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In Int. Conf. Learn. Represent. (ICLR), 2019.

[55] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 14042–14051, 2020.

[56] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5375–5384, 2016.

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 4700–4708, 2017.

[58] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 1173–1182, 2016.

[59] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Int. Conf. Machine Learning (ICML), pages 2127–2136, 2018.

doi:10.6342/NTU202301574

[60] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer G. Dy and Andreas Krause, editors, Int. Conf. Machine Learning (ICML), pages 2132–2141, 2018.

[61] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 7607–7616, 2020.

[62] Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[63] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In Int. Conf. Learn. Represent. (ICLR), 2020.

[64] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In Int. Conf. Learn. Represent. (ICLR), 2020.

[65] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI), pages 4217–4228, 2021.

[66] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[67] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in GAN for real-time image editing. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 852–861, 2021.

[68] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, 2014.

doi:10.6342/NTU202301574

[69] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object representations for fine-grained categorization. In Int. Conf. Comput. Vis. Worksh. (ICCVW), pages 554–561, 2013.

[70] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

[71] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 951–958, 2009.

[72] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998. URL http://yann. lecun. com/exdb/mnist, page 34, 1998.

[73] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5000–5009, 2020.

[74] Xiaodan Li, Yining Lang, Yuefeng Chen, Xiaofeng Mao, Yuan He, Shuhui Wang, Hui Xue, and Quan Lu. Sharp multiple instance learning for deepfake video detection. In ACM Int. Conf. Multimedia (ACMMM), pages 1864–1872, 2020.

[75] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 10988–10997, 2020.

[76] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 3204–3213, 2020.

[77] Cong Han Lim and Steve Wright. A box-constrained approach for hard permutation problems. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, Int. Conf. Machine Learning (ICML), pages 2454–2463, 2016.

doi:10.6342/NTU202301574

[78] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Int. Conf. Comput. Vis. (ICCV), pages 2999–3007, 2017.

[79] Guoqing Liu, Jianxin Wu, and Z-H Zhou. Key instance detection in multi-instance learning. In Asian Conference on Machine Learning, pages 253–268, 2012.

[80] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 772–781, 2021.

[81] Kangjun Liu, Ke Chen, and Kui Jia. Convolutional fine-grained classification with self-supervised target relation regularization. IEEE Trans. Image Process. (TIP), pages 5570–5584, 2022.

[82] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Int. Conf. Comput. Vis. (ICCV), pages 9992–10002, 2021.

[83] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. CoRR, 2019.

[84] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In Int. Conf. Learn. Represent. (ICLR), 2017.

[85] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 289–297, 2016.

[86] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. CoRR, 2015.

[87] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring

the limits of weakly supervised pretraining. In Eur. Conf. Comput. Vis. (ECCV), pages 181–196, 2018.

[88] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. CoRR, 2013.

[89] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In Int. Conf. Learn. Represent. (ICLR), 2021.

[90] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 3111–3119, 2013.

[91] Nam Nguyen. A new svm approach to multi-instance multi-label learning. In IEEE International Conference on Data Mining, pages 384–392, 2010.

[92] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Eur. Conf. Comput. Vis. (ECCV), 2016.

[93] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aäron van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2019.

[94] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In Adv. Neural Inform. Process. Syst. Worksh. (NeurIPSW), 2017.

[95] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. CoRR, 2014.

[96] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Differentiable sorting networks for scalable sorting and ranking supervision. In Marina Meila and Tong Zhang, editors, Int. Conf. Machine Learning (ICML), pages 8546–8555, 2021.

[97] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Monotonic differentiable sorting networks. In Int. Conf. Learn. Represent. (ICLR), 2022.

[98] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Int. Conf. Machine Learning (ICML), pages 8748–8763, 2021.

[99] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review, pages 471–501, 2010.

[100] Hao Ren. A pytorch implementation of simclr. https://github.com/leftthomas/SimCLR, 2020.

[101] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In Int. Conf. Learn. Represent. (ICLR), 2021.

[102] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 10684–10695, 2022.

[103] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In Int. Conf. Comput. Vis. (ICCV), pages 1–11, 2019.

[104] Dvir Samuel, Yuval Atzmon, and Gal Chechik. From generalized zero-shot learning to long-tail with class descriptors. In IEEE Winter Conf. App. Comput. Vis. (WAVC), pages 286–295, 2021.

[105] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 41–48, 2004.

doi:10.6342/NTU202301574

[106] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Int. Conf. Comput. Vis. (ICCV), pages 618–626, 2017.

[107] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and recovering sequential deepfake manipulation. In Eur. Conf. Comput. Vis. (ECCV), pages 712–728, 2022.

[108] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In Eur. Conf. Comput. Vis. (ECCV), pages 467–482, 2016.

[109] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 18699–18708, 2022.

[110] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 761–769, 2016.

[111] Yangyang Shu, Baosheng Yu, Haiming Xu, and Lingqiao Liu. Improving fine-grained visual recognition in low data regimes via self-boosting attention mechanism. In Eur. Conf. Comput. Vis. (ECCV), pages 449–465, 2022.

[112] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, 2014.

[113] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 11659–11668, 2020.

[114] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Int. Conf. Machine Learning (ICML), pages 6105–6114, 2019.

[115] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In Adv. Neural Inform. Process. Syst. (NeurIPS), 2020.

[116] Kevin Thandiackal, Boqi Chen, Pushpak Pati, Guillaume Jaume, Drew F. K. Williamson, Maria Gabrani, and Orcun Goksel. Differentiable zooming for multiple instance learning on whole-slide images. In Eur. Conf. Comput. Vis. (ECCV), pages 699–715, 2022.

[117] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. VL-LTR: learning class-wise visual-linguistic representation for long-tailed visual recognition. In Eur. Conf. Comput. Vis. (ECCV), pages 73–91, 2022.

[118] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Eur. Conf. Comput. Vis. (ECCV), 2020.

[119] Yao-Hung Hubert Tsai, Martin Q. Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. In Int. Conf. Learn. Represent. (ICLR), 2021.

[120] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. CoRR, 2018.

[121] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 8769–8778, 2018.

[122] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[123] Hualiang Wang, Siming Fu, Xiaoxuan He, Hangxiang Fang, Zuozhu Liu, and Haoji Hu. Towards calibrated hyper-sphere representation via distribution overlap coefficient for long-tailed learning. In Eur. Conf. Comput. Vis. (ECCV), pages 179–196, 2022.

doi:10.6342/NTU202301574

[124] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In Eur. Conf. Comput. Vis. (ECCV), pages 20–36, 2016.

[125] Phil Wang. x-clip. https://github.com/lucidrains/x-clip, 2021.

[126] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. CoRR, 2020.

[127] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Int. Conf. Machine Learning (ICML), 2020.

[128] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. Pattern Recognition (PR), 74:15–24, 2018.

[129] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X. Yu. Long-tailed recognition by routing diverse distribution-aware experts. In Int. Conf. Learn. Represent. (ICLR), 2021.

[130] Xudong Wang, Ziwei Liu, and Stella X. Yu. Unsupervised feature learning by cross-level instance-group discrimination. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2021.

[131] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 4148–4157, 2018.

[132] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 7029–7039, 2017.

[133] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2018.

doi:10.6342/NTU202301574

[134] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Int. Conf. Machine Learning (ICML), pages 2048–2057, 2015.

[135] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In Eur. Conf. Comput. Vis. (ECCV), pages 420–435, 2018.

[136] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2019.

[137] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 4651–4659, 2016.

[138] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. CoRR, 2017.

[139] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Int. Conf. Machine Learning (ICML), pages 12310–12320, 2021.

[140] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Dahua Lin, and Chen Change Loy. OpenSelfSup: Open mmlab self-supervised learning toolbox and benchmark. https://github.com/open-mmlab/openselfsup, 2020.

[141] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E. Coupland, and Yalin Zheng. DTFD-MIL: double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 18780–18790, 2022.

[142] Min-Ling Zhang. A k-nearest neighbor based multi-instance multi-label learning algorithm. In IEEE International Conference on Tools with Artificial Intelligence, pages 207–212, 2010.

doi:10.6342/NTU202301574

[143] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Eur. Conf. Comput. Vis. (ECCV), 2016.

[144] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 2185–2194, 2021.

[145] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In Int. Conf. Comput. Vis. (ICCV), pages 15003–15013, 2021.

[146] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 5012–5021, 2019.

[147] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 9719–9728, 2020.

[148] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In Int. Conf. Comput. Vis. (ICCV), pages 14780–14789, 2021.

[149] Zhi-Hua Zhou and Min-Ling Zhang. Multi-instance multi-label learning with application to scene classification. In Adv. Neural Inform. Process. Syst. (NeurIPS), pages 1609–1616, 2007.

[150] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. Artificial Intelligence, pages 2291–2320, 2012.

[151] Benjin Zhu, Junqiang Huang, Zeming Li, Xiangyu Zhang, and Jian Sun. Eqco: Equivalent rules for self-supervised contrastive learning. CoRR, 2020.

[152] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z. Li. Face forgery detection by 3d decomposition. In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), pages 2929–2939, 2021.

doi:10.6342/NTU202301574

[153] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In ACM Int. Conf. Multimedia (ACMMM), pages 2382–2390, 2020.

doi:10.6342/NTU202301574