THE UNIVERSITY OF SYDNEY

DOCTORAL THESIS

Visual Pretraining on Large-Scale Image Datasets

Author:
Shixiang TANG

Supervisor:
Prof. Wanli OUYANG
Dr. Dong YUAN
Co-Supervisor:
Dr. Huaming CHEN

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

School of Electrical and Information Engineering
Faculty of Engineering

October 23, 2023

Abstract of thesis entitled

Visual Pretraining on Large-Scale Image Datasets

Submitted by

Shixiang TANG

for the degree of Doctor of Philosophy

at The University of Sydney

in October, 2023

Large-scale visual pretraining has emerged as a crucial component in the field of computer vision, enabling the development of advanced visual recognition models. This thesis highlights the significance of both supervised and unsupervised pretraining methods, improving their learning frameworks and objective functions.

The thesis explores the limitations of previous approaches, which either rely on single-instance-positive sample pairs or prototype-positive clustering, overlooking the relative nature of positive and negative concepts in the real world. To address these limitations, a novel technique called Relative Contrastive Loss (RCL) is proposed, which leverages relatively positive/negative pairs to learn feature representations that encompass more real-world semantic variations while respecting positive-negative relativeness.

Furthermore, this thesis presents UniVCL (Unified Visual Contrastive Learning), a unified framework for existing unsupervised visual contrastive learning methods. We discover that different designs of predictors can be treated as the special formulations of a graph neural network. Therefore, our UniVCL adopts a graph convolutional network (GCN) layer as the predictor layer, showcasing two major advantages in unsupervised object recognition. Firstly, through comprehensive experiments, the critical importance of neighborhood aggregation in the GCN

predictor is revealed, shedding light on the significance of this component in existing methods. Secondly, by viewing the predictor from a graph perspective, the integration of graph representation learning augmentations enhances unsupervised object recognition accuracy.

The classical pipeline of pretrain-finetune in visual learning is revisited, particularly in the context of unsupervised pretraining methods exhibiting superior transfer performance compared to supervised counterparts. This thesis presents a new perspective on the transferability gap between unsupervised and supervised pretraining, focusing on the multilayer perceptron (MLP) projector. Through extensive analysis, it is discovered that the MLP projector plays a key role in enhancing the transferability of unsupervised pretraining methods. Based on this finding, an MLP projector is incorporated before the classifier in supervised pretraining, resulting in reduced feature distribution distance, increased retention of intra-class variation, and decreased feature redundancy.

Finally, this thesis acknowledges the importance of human-centric perceptions in diverse industrial applications such as surveillance, autonomous driving, and the metaverse. To address the need for a general pretrain model for versatile human-centric downstream tasks, the thesis proposes HumanBench, a comprehensive benchmark consisting of 19 datasets across six diverse downstream tasks. Various pretraining methods are evaluated on this benchmark to assess their generalization abilities. Additionally, a novel pretraining method called PATH (Projector Assisted Hierarchical pretraining) is introduced to learn coarse-grained and fine-grained knowledge in human bodies.

The findings in this thesis confirm the effectiveness of the proposed methods in enhancing the practicality and performance of large-scale visual pretraining. Through rigorous testing on diverse datasets, the relative contrastive loss, unified contrastive learning framework combining visual and graph unsupervised pretraining, and supervised pretraining with MLP projectors have consistently yielded impressive results. These outcomes provide compelling evidence for the efficacy of the proposed methods in improving the design and training process of large-scale visual pretraining.

Visual Pretraining on Large-Scale Image Datasets

by

Shixiang TANG
B.S. *Fudan University*
M.Phil. *The Chinese University of Hong Kong*

A Thesis Submitted in Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

at

The University of Sydney
October, 2023

Originality Statement

I, Shixiang TANG, declare that this thesis titled, "Visual Pretraining on Large-Scale Image Datasets", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed: _____

Date: October 23, 2023

For Mama and Papa

Acknowledgements

The successful completion of this doctoral dissertation embodies a challenging yet enlightening academic journey. The realization of this project would not have been feasible without the indispensable contributions and unwavering support of numerous individuals and organizations.

I reserve my deepest gratitude for my family - my father, mother, younger sister, and all my cherished loved ones. Their unfaltering encouragement and steadfast faith in my abilities have provided a solid foundation upon which I have been able to build. Their consistent support has been instrumental in fortifying my resilience throughout this intellectual odyssey.

My lead supervisor, Prof. Wanli Ouyang, now affiliated with The Shanghai AI Laboratory, warrants a particular note of gratitude. His exceptional mentorship and rigorous guidance have indelibly shaped my research trajectory. Dr. Dong Yuan from the University of Sydney also deserves special acknowledgment. His insightful expertise and scholarly wisdom have equipped me with invaluable skills and knowledge that will undoubtedly guide me throughout my forthcoming career.

I am further grateful to Dr. Feng Zhu, Dr. Rui Zhao, Dr. Lei Bai, and Mr. Yizhou Wang, along with all the colleagues and collaborators I have worked with. Their intellectual collaboration, ongoing assistance, and inspiration have substantially enriched the academic environment during my postgraduate studies.

Lastly, I owe an immense debt of gratitude to The University of Sydney for their unyielding institutional support and the financial sponsorship they provided for my research. I extend my sincere appreciation to SenseTime Research as well. Their munificent provision of computational resources was integral in facilitating the progress and completion of my research endeavor.

This thesis is a testament to the combined effect of all your efforts.

It is with the deepest sense of gratitude that I acknowledge all your contributions.

Shixiang TANG
The University of Sydney
October 23, 2023

Authorship Attribute Statement

Chapter 3 of this thesis is published as

Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, and Wanli Ouyang, Relative contrastive loss for unsupervised representation learning, in European Conference on Computer Vision, 2022.

I hold the first authorship of this work, playing a pivotal role in the conception of the research method, the design and implementation of the code, the execution of a significant portion of the experiments, and in drafting and revising the manuscript. Dr. Lei Bai, Dr. Feng Zhu, Dr. Rui Zhao, and Prof. Wanli Ouyang have contributed significantly to the development of this work by providing critical input and constructive suggestions that have led to substantial improvements in the manuscript, the methodological approach, and the experimental designs.

Chapter 4 of this thesis is published as

Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, and Wanli Ouyang, Unifying visual contrastive learning for object recognition from a graph perspective, in European Conference on Computer Vision, 2022.

As the first author of this research work, I took on the responsibility of initiating and formulating the methodological approach, designing and implementing the requisite code, overseeing and executing the experiments, and penning and refining the draft manuscript. Dr. Feng Zhu, Dr. Lei Bai, Dr. Rui Zhao, and Prof. Wanli Ouyang made substantial scholarly contributions to this endeavour. Their roles entailed participating in the critical revision of the draft, offering intellectual input, and providing constructive suggestions that have enriched the quality of the

written content, refined the methodological approach, and enhanced the experimental designs.

Chapter 5 of this thesis is published as

Yizhou Wang*, **Shixiang Tang***, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang, Revisiting the transferability of supervised pretraining: an mlp perspective, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

As the co-first author of this research work, I took on the responsibility of initiating and formulating the methodological approach, designing the codebase, conducting the experiments (excluding the ablation study and improved supervised pretraining), and drafting and refining the manuscripts (excluding the experiments section).

Mr. Yizhou Wang, my co-first author, was instrumental in implementing and validating numerous innovative concepts associated with the transferability gap between self-supervised and supervised pretraining, as well as enhancing the transferability of supervised pretraining. He took on the responsibility of authoring the experiments section of the manuscript and conducted all the experiments related to the ablation study.

Dr. Lei Bai, Dr. Feng Zhu, and Prof. Wanli Ouyang, provided invaluable assistance with the manuscript revisions and offered insightful perspectives that significantly improved our methodology and experimental design. Furthermore, Dr. Rui Zhao and Prof. Donglian Qi enriched the breadth of our research by offering suggestions that enhanced the literature review. Their collective efforts have substantially elevated the quality and scholarly merit of this work.

Chapter 6 of this thesis is published as

Shixiang Tang*, Cheng Chen*, Qingsong Xie, Meilin Chen, Yizhou Wang, Yuanzheng Ci, Lei Bai, Feng Zhu, Haiyang Yang, Li Yi, et al. Humanbench: Towards general human-centric perception with projector assisted pretraining, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023.

I serve as co-first author and was chiefly responsible for proposing the research method, designing and executing the base codes, and conducting experiments centred on single-task verifications, multi-task training, and ablation studies. My responsibilities extended to drafting the initial version of the manuscript (with the exception of the experiments section), crafting the illustrative figures, and refining the draft manuscript.

My co-first author, Mr. Cheng Chen, played a pivotal role in implementing and validating a multitude of our ideas pertaining to person re-identification, pose estimation, semantic segmentation, attribute recognition, and pedestrian detection. His valuable contributions to the manuscript included authoring the experiments section and executing all fine-tuning and low-data regime experiments, with the collaborative assistance of Mr. Meilin Chen, Mr. Haiyang Yang, and Mr. Qingsong Xie.

Mr. Qingsong Xie and Mr. Li Yi facilitated our work by preparing the code framework, evaluation code, and datasets. Their engagement extended to participating in our discussions, contributing to the experimental design, and aiding in the refinement of the manuscript.

Dr. Lei Bai, Dr. Feng Zhu, and Prof. Wanli Ouyang provided vital assistance with manuscript revisions, offering insightful suggestions that enhanced our methodology and experimental designs. Dr. Rui Zhao and Prof. Donglian Qi also enriched the depth of our work, providing valuable suggestions to bolster the literature review. Their collective efforts have contributed to the comprehensive scholarly merit of this work.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Signed: _____

Date: October 23, 2023

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signed: _____

Date: October 23, 2023

Signed: _____

Date: October 23, 2023

List of Publications

* Equal Contribution

Conferences:

[1] **Shixiang Tang***, Cheng Chen*, Qingsong Xie, Meilin Chen, Yizhou Wang, Yuanzheng Ci, Lei Bai, Feng Zhu, Haiyang Yang, Li Yi, et al. Humanbench: Towards general human-centric perception with projector assisted pretraining, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

[2] **Shixiang Tang**, Feng Zhu, Lei Bai, Rui Zhao, and Wanli Ouyang. Unifying visual contrastive learning for object recognition from a graph perspective, in *European Conference on Computer Vision*, 2022.

[3] **Shixiang Tang**, Feng Zhu, Lei Bai, Rui Zhao, and Wanli Ouyang. Relative contrastive loss for unsupervised representation learning, in *European Conference on Computer Vision*, 2022.

[4] **Shixiang Tang**, Dapeng Chen, Jinguo Zhu, Shijie Yu, and Wanli Ouyang. Layerwise optimization by gradient decomposition for continual learning, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[5] **Shixiang Tang**, Dapeng Chen, Lei Bai, Yixiao Ge, and Wanli Ouyang. Mutual crf-gnn for few shot learning, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[6] **Shixiang Tang**, Peng Su, Dapeng Chen, and Wanli Ouyang. Gradient regularized contrastive learning for continual domain adaptation, in *AAAI Conference on Artificial Intelligence*, 2021

[7] Yizhou Wang*, **Shixiang Tang***, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang. Revisiting the transferability of supervised pretraining: an mlp perspective, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[8] Haiyang Yang, Xiaotong Li, **Shixiang Tang**, Feng Zhu, Yizhou Wang, Meilin Chen, LEI BAI, Rui Zhao, and Wanli Ouyang. Cycle-consistent masked autoencoder for unsupervised domain generalization, in *International Conference on Learning Representations*, 2023.

[9] Yizhou Wang, Meilin Chen, **Shixiang Tang**, Feng Zhu, Haiyang Yang, Lei Bai, Rui Zhao, Yunfeng Yan, Donglian Qi, and Wanli Ouyang. Unsupervised object detection pretraining with joint object priors generation and detector learning, in *Conference on Neural Information Processing Systems*, 2022.

[10] Haiyang Yang, **Shixiang Tang**, Meilin Chen, Yizhou Wang, Feng Zhu, Lei Bai, Rui Zhao, and Wanli Ouyang. Domain invariant masked autoencoders for self-supervised learning from multi-domains, in *European Conference on Computer Vision*, 2022.

PATENTS:

[1] **Shixiang Tang**, Guanxiong Cai, Quanyuan Zheng, Dapeng Chen, R Zhao, "Method, apparatus, device, medium and program for image detection and related model training", US Patent App. 17/718,585.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The opening of this chapter introduces the techniques for the design and training of neural networks that will be addressed in this thesis. We then highlight the key challenges in the field, motivations behind these challenges, as well as the efficient solutions contributed in this thesis to tackle these challenges. We conclude by outlining the organization of the remainder of this thesis.

1.1 Background

Following the revolutionary recognition performance of AlexNet [124] in the ImageNet competition [45], the field of artificial intelligence has seen significant advancements. Numerous representative deep neural networks have been proposed, including but not limited to VGG [207], ResNet [90], Inception [218], and LSTM [97]. Researchers typically gather and annotate task-specific samples and train their models on extensive datasets such as ImageNet for computer vision and Glove [184] and Skip-thought vectors [116] for natural language processing. This approach enables the resolution of many tasks end-to-end, circumventing the need for traditional handcrafted features and providing solutions for object detection [149, 55, 29], segmentation [89, 221, 214], and recognition [301, 229, 220] among others. The detailed process is summarized as "Traditional Development of Vision Algorithms" in Fig. 1.1. However, the way of developing algorithms in vision tasks suffer from two significant drawbacks. First, researchers are requied to train one model for every specific vision task, limiting the fast and convenient deployment

when we need to tackle a set of vision tasks. Second, the performance of every vision tasks is largely limited to the number of data collected for the exact task, unable to benefit from data and annotations collected for other tasks.

In recent years, there has been a notable shift in research to address this issue. This shift has emphasized the use of self-supervised or supervised pretraining on vast quantities of data, such as images or text, which is then fine-tuned for specific tasks. Rather than relying solely on task-specific annotations, the pretraining stage of this framework takes advantage of as many images as possible, with varied or even no annotations. The fine-tuning stage is dedicated to effectively adapting these pretrained models to perform specific downstream tasks, utilizing only a minimal amount of downstream data. This revamped pretrain-finetune approach has several key benefits. First, the pretrained model can address the diverse tasks by freezing the pretrained backbone and fine-tuning the task heads. Second, the performance of each downstream task can be enhanced, as the pretrained model absorbs knowledge from a large and potentially diverse pool of annotated data. Lastly, models for different downstream tasks can be efficiently adapted from the pretrained model at a low computational cost and with minimal need for task-specific annotated data.

1.2 Statement of Large-scale Pretraining

Large-scale Pretraining is a field of study that trains deep neural networks on massive datasets of labeled or unlabeled images, allowing them to learn intricate visual features, patterns, and representations. These models, often referred to as vision models or pretrained visual encoders, have the capacity to comprehend and extract meaningful information from images, enabling them to perform a wide range of tasks, including image classification, object detection, semantic segmentation, and image generation. Therefore, the pretrained visual encoders are expected to be able to improve the performance of various downstream tasks after efficient adaptation. According to whether labels are utilized in the

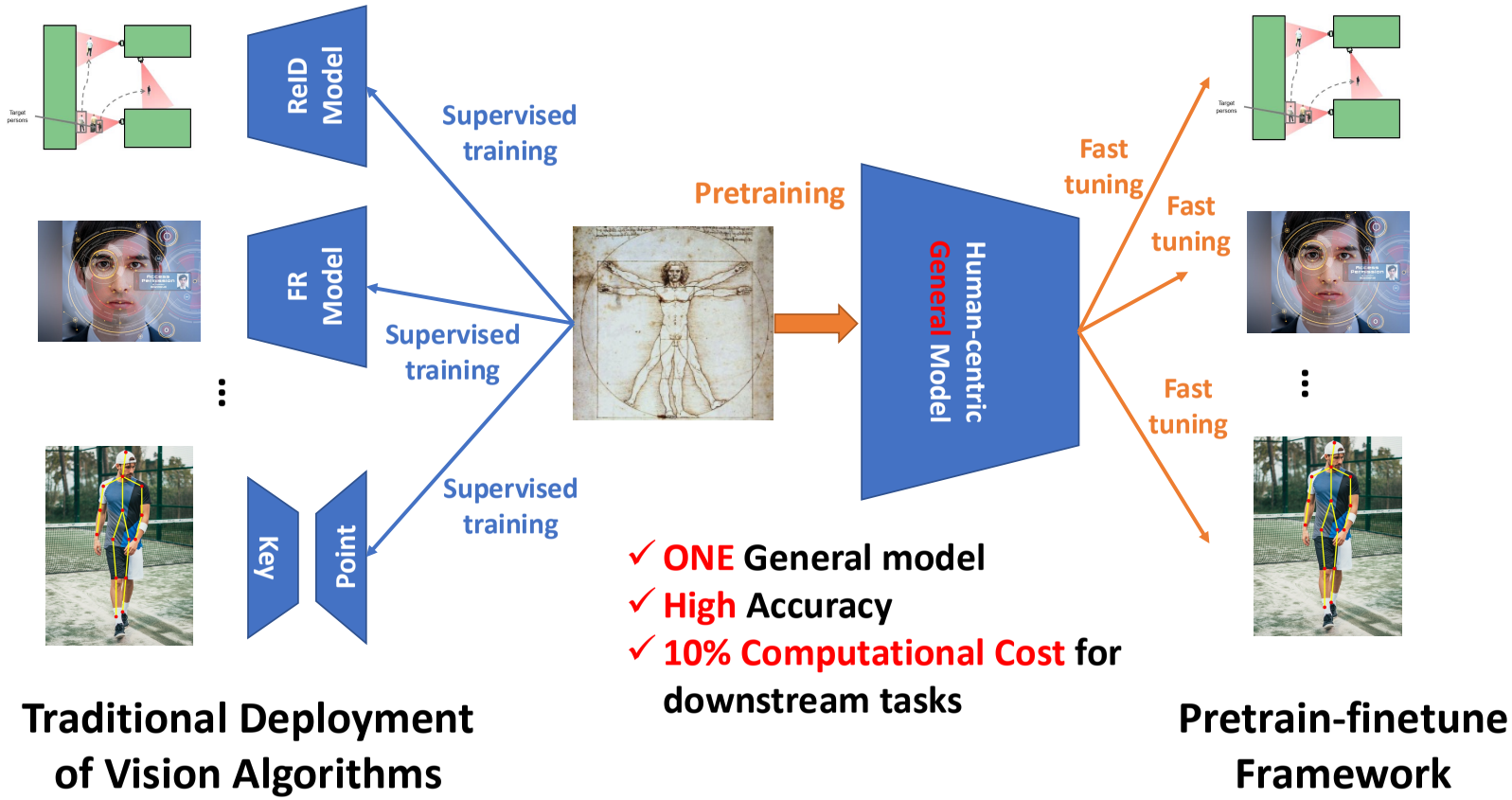


Figure 1.1: Comparison of traditional development of vision algorithms and pretrained-finetune framework for developing vision algorithms. Traditional development of vision algorithms utilize the randomly-initialized backbone and then train every downstream tasks by finetuning all parameters of the model. In pretrain-finetune framework, models are pretrained on large-scale images, and then finetune the pretrained model to various downstream tasks.

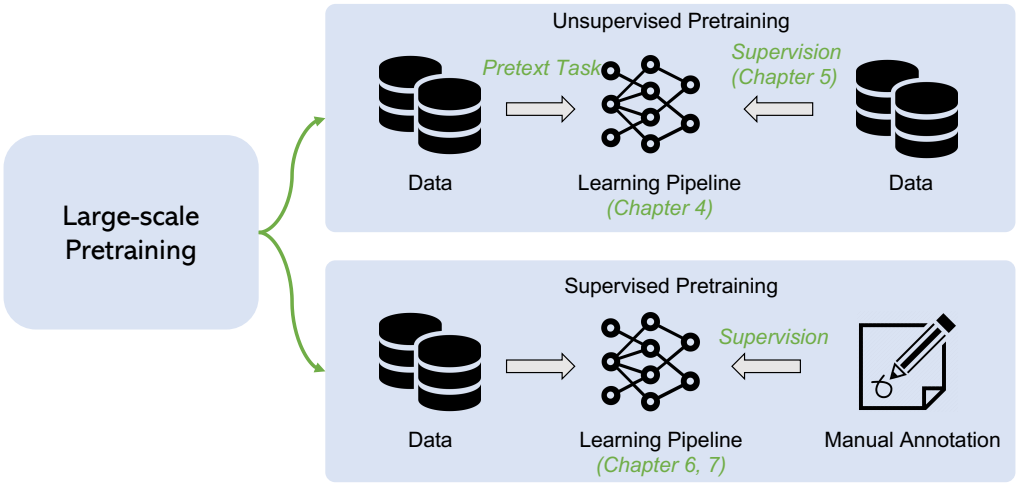


Figure 1.2: A schematic illustration of large-scale pretraining, including unsupervised pretraining and supervised pretraining.

pretraining, the large-scale pretraining can be generally divided into supervised pretraining and unsupervised pretraining.

Supervised Learning. Supervised pretraining is commonly performed using large-scale labeled datasets, such as ImageNet or COCO, which contain millions of annotated images across various object categories and scenes. By training on such datasets, the network can learn to capture a wide range of visual features, object shapes, textures, and spatial relationships. After the supervised pretraining phase, the network can be fine-tuned on a smaller labeled dataset specific to the target task, which might have different classes or categories. Fine-tuning involves updating the network's parameters using the labeled data from the target task, allowing it to specialize and adapt its knowledge to the specific visual recognition or analysis task at hand.

Unsupervised Pretraining. Unsupervised pretraining in computer vision refers to the process of training a deep neural network on a large dataset of unlabeled or weakly labeled images without explicit supervision or ground truth annotations. Instead of relying on labeled data, unsupervised pretraining aims to learn meaningful representations or

features from the visual data by uncovering underlying structure or patterns. After pretrained in an unsupervised manner, the models are expected to achieve improved performances after adapting to various downstream tasks.

1.3 Challenges, Motivations and Contributions

Despite advancements in large-scale pretraining, there are still numerous challenges that must be addressed to make the development of large-scale pretraining more effective. In this thesis, we will foucs on key challenges that are critical to developing large-scale pretraining methods, including unsupervised pretraining and supervised pretraining methods. Unsupervised pretraining and supervised pretraining are two popular frameworks for visual pretraining. Unsupervised pretraining does not use manual annotations but relies on well-designed proxy tasks to learn general knowledge from images. In contrast, supervised pretraining leverages manual annotations or pseudo-labels to learn general information from images. In particular, supervised pretraining can be further categorized into single task supervised pretraining and multitask supervised pretraining.

1.3.1 Unsupervised Pretraining

Unsupervised pretraining aims at learning general information from images without using any labels. A popular framework for unsupervised learning is contrastive learning. Contrastive learning aims to minimize the distance of different views of the same sample and maximize the distance of the representations of different samples. Through this learning framework, the representation distance between two semantically similar images will be smaller than between two semantically different images.

The first challenge entails devising an enhanced objective function within the existing contrastive learning framework. Contrastive learning [101, 87, 35, 37, 31, 259, 318, 80, 285, 36] optimizes deep networks by

simultaneously reducing the distance between representations of positive pairs and increasing the distance between representations of negative pairs in the latent feature space. Conventionally, positive and negative samples in contrastive loss are categorized based on different augmentations of the same image or samples with the same pseudo-labels using clustering [22] as positive samples [87]. In contrast, humans possess the ability to recognize relative similarities rather than categorically constructing positive and negative pairs, as demonstrated by Carl Linnaeus, a Swedish botanist, in his hierarchical description of biological organisms under seven levels: Kingdom, Phylum, Class, Order, Family, Genus, and Species, forming the current Linnaean taxonomy system [58]. Notably, prevalent computer vision benchmarks such as ImageNet [45], iNat21 [233], and Places365 [309] also follow the concept of positive-negative relativeness by incorporating hierarchical labels. For instance, within ImageNet, both trimarans and boats fall under the broader category of sailing vessels and vehicles. However, when distinguishing between different types of sailing vessels, trimarans and boats represent distinct classes. To account for the relative nature of human recognition, we propose the Relative Contrastive Loss, which introduces a set of semantic criteria to partially determine the positive or negative nature of a given sample pair. This approach captures real-world instance variations within a class in a relative manner. An image pair that meets more semantic criteria will be assigned as higher relative postiveness. Consequently, our relative contrastive loss pulls the features of image pairs with higher postiveness closer together compared to those with lower postiveness.

The second challenge entails demystifying the real effectiveness of each components among diversed unsupervised learning methods based on contrastive learning and designing an improved learning framework for unsupervised pretraining. Current contrastive-based SSL methods employ a Siamese architecture comprising two branches: an online branch and a target branch. Each branch consists of a backbone, a projector layer, and an optional predictor layer. While the backbone and projector layer processes are similar across these methods, significant differences exist in their predictor layer designs. For example, self-loop

operation in BYOL [80] and neighborhood aggregation in NNCL [56]. To elucidate the impact of these projectors on unsupervised learning performance, we begin by developing a comprehensive framework that unifies various projector designs by using a Graph Convolutional Network [310, 260, 199] (GCN) as the projector layer, since the graph neural network can both model self-loop operation and neighborhood aggregation simultaneously. Subsequently, we conduct thorough and unbiased experiments to evaluate the effectiveness of each component rigorously. Our meticulous experiments yield three noteworthy observations. Firstly, the inclusion of a neighborhood aggregation term in the GCN layer significantly enhances linear evaluation performance. Secondly, the choice of activation function noticeably influences linear evaluation performance. Lastly, if the other components of the GCN layer are well-designed, the performance difference between different non-linear activation functions is relatively small, indicating that the selection of a non-linear activation function is a less crucial factor. With this graph-based unified framework, we establish a connection between vision unsupervised pretraining and graph unsupervised pretraining, which is an active area of research in unsupervised pretraining. Remarkably, we discover that the pretext tasks and data augmentations utilized in graph unsupervised pretraining are also effective for vision unsupervised pretraining.

1.3.2 Supervised Pretraining

Supervised Pretraining [64, 250, 168] is commonly performed on various large-scale labeled datasets under a multitask learning framework. The challenges of the supervised pretraining lies in two aspects. The first is to bridge the generalization gap between supervised pertraining and unsupervised pretraining because supervised pretraining is longly argued for worse generalization ability than unsupervised pretraining. The other critical challenge for supervsied pretraining lies in the task conflicts when pretraining the network under the multitask learning framework [27, 164, 294, 183].

To bridge the generalization ability gap between unsupervised pretraining and supervised pretraining, we revisit the structural designs of both supervised pretraining and unsupervised pretraining and discovered that the projector layer before the objective function, which is usually a simple MLP module, is the key for the difference between unsupervised pretraining and supervised pretraining. Based on our findings, we propose a new supervised pretraining framework (SL-MLP) by adding an MLP projector before the objecive function in the supervised pretraining method. Extensive experimental results confirm that the newly proposed supervised learning can consistently improve the transferability of the model when adapting the model to various downstream tasks.

To address the challenge of significant task conflicts arising from diverse datasets with varying annotations in multitask learning, we introduce an extension of SL-MLP, called **P**rojector **A**ssis**T**ed **H**ierarchical Pre-training (**PATH**), within the multitask learning framework. Drawing inspiration from SL-MLP [250], which emphasizes the inclusion of an MLP projector before the task head to enhance supervised pretraining's generalization ability, we propose PATH as a projector-assisted pretraining approach that employs hierarchical weight sharing to address the task conflicts associated with diverse annotations in supervised pretraining. In this method, the backbone weights are shared across all datasets, while the projector weights are shared only among datasets of the same tasks. Additionally, the head weights are shared exclusively within a single dataset, creating a hierarchical weight-sharing structure. During the pretraining stage, we incorporate task-specific projectors before the dataset heads, but discard them when evaluating models on downstream tasks. Leveraging the hierarchical weight-sharing strategy, our pretraining method guides the backbone to learn from a shared knowledge pool, directs the projector to focus on task-specific knowledge, and enables the head to concentrate on the dataset's specific annotation and data distribution. Our extensive experiments, conducted on a large-scale multitask training framework utilizing a self-supervised pretrained backbone, demonstrate that PATH can serve as a powerful and efficient alternative to employing multiple tailored models for general human-centric tasks.

1.3.3 Summary

This thesis investigates the methods to address main challenges in the framework and objective function of visual pretraining, including: (1) designing a better relative contrastive loss to capture the relative semantic similarity of any image pair, (2) analyzing the key components in contrastive learning unsupervised framework and relating visual unsupervised pretraining and graph unsupervised pretraining, (3) demystifying and closing the transferaiblity gap between unsupervised pretraining and supervised pretraining, and (4) developing the efficient and unified multi-task model that can handle and benefit from large-scale multitask learning. Our major contributions include a new relative contrastive loss, a unified learning framework of visual contrastive learning and graph contratsive learning for unsupervised pertraining, an investigation of the transferability gap between supervised and unsupervised pretraining framework and a new multitask supervised pretraining framework to effecively tackle task conflicts. Our extensive and rigorous experiemnts demonstrate the effectiveness of the proposed methods in improving performance of large-scale pretraining on image datasets.

1.4 Thesis Outline

This thesis seeks to explore the challenges found in current methods of large-scale visual pre-training, while also introducing innovative solutions to these problems. The structure of this thesis is outlined over seven chapters. Chapter 1 offers a broad introduction to the subject of large-scale visual pre-training, while Chapter 2 delves into a comprehensive review of existing literature on the topic. The following chapters, from 3 to 6, introduce four distinct methods aimed at enhancing large-scale pretraining and venturing into novel research areas. Chapter 7 encapsulates a summary and conclusion of the proposed methods.

Chapter 2. Literature Review. In this chapter, we provide an in-depth review of the background information relevant to this thesis, including

learning pipelines and objective function for unsupervised learning, supervised multitask learning and task adaptation techniques that are crucial for large-scale visual pretraining. This information will help readers better understand the subsequent chapters.

Chapter 3. Improved Unsupervised Pretraining by Relative Contrastive Loss. In this chapter, we present a relative contrastive loss, the first method respecting the relative nature of positive/negative concepts in the real world. Motivated by the ability of humans in recognizing relatively positive/negative samples, we propose the Relative Contrastive Loss (RCL) to learn feature representation from relatively positive/negative pairs, which not only learns more real world semantic variations than the single-instance-positive methods but also respects positive-negative relativeness compared with absolute prototype-positive methods. The proposed RCL improves the linear evaluation for MoCo v3 on ImageNet.

Chapter 4. Unified Unsupervised Pretraining Pipeline and Improved Data Augmentation from a Graph Perspective. In this chapter, we propose to Unify existing unsupervised Visual Contrastive Learning methods by using a GCN layer as the predictor layer (UniVCL), which deserves two merits. First, by treating different designs of predictors in the existing methods as its special cases, our fair and comprehensive experiments reveal the critical importance of neighborhood aggregation in the GCN predictor. Second, by viewing the predictor from the graph perspective, we can bridge the vision self-supervised learning with the graph representation learning area, which facilitates us to introduce the augmentations from the graph representation learning to unsupervised object recognition and further improves the unsupervised object recognition accuracy. Extensive experiments on linear evaluation and the semi-supervised learning tasks demonstrate the effectiveness of UniVCL and the introduced graph augmentations.

Chapter 5. Improved Transferability of Supervised Pretraining from an MLP Perspective. In this chapter, we revisit the difference in transferability between unsupervised and supervised pretraining from the

standpoint of a multilayer perceptron (MLP). We identify the MLP projector as a crucial component for better transferability in unsupervised pretraining methods. To bridge the transferability gap, we introduce an MLP projector before the classifier in supervised pretraining, resulting in improved intra-class variation of visual features, reduced distribution distance between pretraining and evaluation datasets, and decreased feature redundancy. Experiments reveal that implementing an MLP projector can significantly enhance the transferability of supervised pretraining, achieving a **+7.2%** increase in top-1 accuracy on concept generalization tasks, a **+5.8%** rise in top-1 accuracy for linear evaluation on 12-domain classification tasks, and a **+0.8%** gain in Average Precision (AP) on the COCO object detection task. This development makes supervised pretraining comparable or even superior to unsupervised pretraining.

Chapter 6. Improved Extreme Multitask Supervised Pretraning for Human-Centric Perception. In this chapter, we propose to alleviate the significant task conflicts for pretraining various human-centric tasks in a multitask supervised manner. To learn both coarse-grained and fine-grained knowledge in human bodies, we further propose a **P**rojector **A**ssis**T**ed **H**ierarchical pretraining method (**PATH**) to learn diverse knowledge at different granularity levels. Comprehensive evaluations on HumanBench show that our PATH achieves new state-of-the-art results on 17 downstream datasets and on-par results on the other 2 datasets.

Chapter 7. Conclusion and Future Work. In this chapter, we summarize all contributions and propose promising research directions in the future based on my research in this thesis.

Chapter 2

Literature Review

2.1 Large-scale Visual Unsupervised Pretraining

2.1.1 Contrastive based Unsupervised Pretraining

Single-instance-positive Methods Instead of designing new pre-text tasks [48, 289, 178, 179, 22], recent unsupervised learning methods are developed upon contrastive learning, which tries to pull the representations of different augmented views of the same sample/instance close and push representations of different instances away [31, 101, 87, 35, 56, 38, 109, 83]. Contrastive methods require to define positive pairs and negative pairs in an absolute way, which violates the relativeness of human recognition. This issue of previous contrastive methods strongly motivates the need for relative-contrastive approaches that can reflect the nature of relativeness when human recognize objects. We achieve this goal by introducing a new relative contrastive loss. Instead of defining positive and negative pairs according to one absolute criterion, we assign a sample pair positive or negative by a set of different criteria to mimic the relative distinguish ability.

Clustering-based Methods. Instead of viewing each sample as an independent class, clustering-based methods group samples into clusters [22, 24, 287, 318]. Along this line, DeepCluster [22] leverages k-means assignments of prior representations as pseudo-labels for the new representations. SwAV [24] learns the clusters online through the Sinkhorm-Knopp transform [118, 28]. Our method is also related to these clustering-based

methods in that we instantiate our relative contrastive loss with an online hierarchical clustering. [23] leverages the hierarchical clustering to tackle non-curated data [224], instead of tackling curated data, *i.e.,* ImageNet-1K, in our relative contrastive loss. However, these clustering-based methods define positive and negative pairs explicitly. In our method, a pair of samples can be partially positive, respecting the relativeness of similarity between a pair of samples.

Neighborhood-based Methods. Neighborhood-based methods stand the recent states-of-the-art methods in unsupervised learning. NNCLR [56] replaces one of the views in single-instance-positive methods with its nearest neighbor in the feature space as the positive sample. MSF [209] makes a further step by using the k nearest neighbors in the feature space as the positive samples. Neighborhood-based methods perform better than single-instance-positive methods because they can capture more class-invariances that cannot be defined by augmentations and better than clustering methods because the query and the positive samples are more likely to belong to the same class. Our work also consider neighbors, but in a relative way.

2.1.2 Reconstruction based Unsupervised Pretraining

Inspired by BERT [47]'s Masked Language Modeling, Masked Image Modeling (MIM) has gained popularity as a pretext task for visual representation learning [7, 11, 86]. MIM aims to reconstruct masked tokens from a corrupted input. Current MIM approaches can be categorized into two groups based on their reconstruction targets. SimMIM [268] suggests that raw pixel values of randomly masked patches are effective reconstruction targets, and pretraining can be accomplished with a lightweight prediction head. In contrast, MAE [86] utilizes only the visible patches as input to the encoder, with mask tokens added between the encoder and decoder. This asymmetric design significantly reduces the computation overhead of the encoder. To further enhance the feature extraction capability of the encoder, CAE [34] explicitly separates

the encoder and decoder by introducing a feature alignment module between them. Jean-Baptiste et al. [4] propose learning representations by reconstructing original videos from synthetically mixed ones. Instead of manually constructing the reconstruction target, the use of a network to generate the reconstruction target has been widely employed. In such cases, an image tokenizer is utilized to convert an image into visual tokens. BEiT adopts a pretrained discrete VAE (dVAE) [191, 195] as the tokenizer, but the original MSE loss in dVAE is inadequate for enforcing the tokenizer to capture high-level semantics. PeCo [51] proposes applying perceptual similarity loss during dVAE training to drive the tokenizer to generate better semantic visual tokens, thereby enhancing pretraining. Furthermore, the offline pretrained tokenizer in BEiT limits the model's adaptability. To address this issue, iBOT [311] suggests using an online tokenizer to generate the visual tokens. Concurrent works explore the use of MAE on hierarchical Vision Transformers. UM-MAE [139] proposes a new masking strategy to adapt MAE for pretraining pyramid-based ViTs (e.g., PVT [245], Swin [160]). GreenMIM [104] also adapts MAE for hierarchical architectures, utilizing an optimal grouping algorithm to partition local windows into equal-sized groups.

2.2 Large-scale Graph Unsupervised Learning

Recent years witness the development of deep learning on graphs [279, 157, 102, 240, 115], since the graph-structured is ubiquitous in numerous domains, including e-commence [143], traffic [261], and knowledge base [155]. The biggest challenge of self-supervised learning on graphs lies in learning the topology information in the existing network. Contrastive learning methods are also cornerstones for self-supervised learning on graphs [215, 211, 239, 21, 193]. In particular, the contrastive loss uses two different augmented views of the same graph as the positive samples to maximize their mutual information. Typical augmentation methods include Node Feature Masking [102, 317], Edge Modification [103, 278], and Graph Diffusion [117, 85]. However, self-supervised Learning on graph has not been investigated for unsupervised object recognition. In this thesis, by incorporating GCN layers as the predictor in the deep

models for object recognition, we can introduce and verify the effectiveness of graph augmentations that are widely adopted in the graph self-supervised learning on the unsupervised object recognition.

2.3 Large-scale Visual Supervised Pretraining

2.3.1 Transferability Gap between Supervised Pretraining and Unsupervised Pretraining

Supervised ImageNet pretraining, which combines the ImageNet dataset [45] with deep neural networks [124], has demonstrated the ability to learn generic representations that benefit various applications, including high-level detection [76, 200], segmentation [162], low-level texture synthesis [69], and style transfer [70]. Remarkably, ImageNet pretraining has shown excellent performance even in domains with significant domain gaps, such as medical imaging [174] and depth estimation [152]. Extensive research has been conducted to investigate the transferability of ImageNet pretrained networks [3, 8]. Studies have also quantified the transferability across different layers of neural networks for image classification [275], revealing that reducing the dataset size has only a modest impact on transfer learning using AlexNet [105]. Furthermore, a correlation has been observed between ImageNet classification accuracy and transfer performance [121], and the benefits of ImageNet pretraining become marginal when the target task has sufficient data [88].

In addition to ImageNet transfer, efforts have been made to understand the structures and relationships between tasks in general transfer learning [205, 206]. Taskonomy [284] constructs a relation graph encompassing 22 visual tasks and systematically explores task similarities. Task cooperation and competition have been quantitatively measured in [210] to enhance transfer learning. Negative transfer has been observed in cases of task misalignment [251], and the number of shared layers among tasks depends on their similarity [234]. Alternative methods have also been proposed to measure task structure and similarity in other works [57, 231].

While existing studies predominantly focus on supervised pretraining for transfer learning, our research centers on analyzing transfer learning based on unsupervised pretraining, particularly contrastive learning with the instance discrimination task [54, 259, 87, 31]. Over the years, significant progress has been made in the research community regarding self-supervised learning [48, 49, 289, 75, 74] and contrastive learning [181, 318, 226, 94, 298], narrowing the performance gap with supervised learning for ImageNet classification. Recent works [79, 119, 23] have attempted to scale unsupervised learning to uncurated data beyond ImageNet. Furthermore, several studies on contrastive learning [87, 171] report superior transfer results compared to supervised counterparts for downstream tasks like detection, segmentation, and pose estimation. However, the reasons behind the improved transfer learning performance achieved through contrastive pretraining are still not well understood.

2.3.2 MLP in unsupervised learning methods

Adding a multilayer perceptron (MLP) projector after the encoder was first introduced in SimCLR [32] and followed by recent unsupervised learning frameworks [35, 80, 24, 267, 36, 33]. SimCLR claims that the MLP can reduce the loss of information caused by the contrastive loss, and various works [32, 35] have verified that the MLP projector can enhance the discriminative ability of unsupervised models on the unsupervised image classification task, where unsupervised training and evaluation are conducted on the same dataset, *e.g.*, ImageNet-1K. However, the relation between the MLP and the transferability of unsupervised learning methods is under-explored. in this thesis, we reveal that the MLP projector is also important for the desirable transferability of unsupervised learning.

2.3.3 MLP in supervised learning methods

The typical supervised learning method only uses the cross-entropy loss and shows inferior performance on various transfer tasks than recent unsupervised learning methods. Inspired by [87, 24, 264, 56], recent

researches [112, 108] introduced the contrastive loss equipped with an MLP projector into supervised learning to improve its transferability. Nonetheless, those works ignored the ablation on the MLP projector and attributed the better transfer performance to the contrastive mechanism in the loss. in this thesis, we propose that the MLP projector is important for the improved transferability of recent supervised learning methods [112, 108], and further provide some empirical and theoretical analysis to justify its importance.

2.4 Multitask Supervised Pretraining

Multi-task pretraining has garnered significant attention within the research community [156, 27, 16, 192, 81, 302, 248, 313]. A prevalent approach in multi-task pretraining involves the sharing of hidden layers from a backbone model across different tasks, commonly referred to as "hard-sharing" in the literature. However, it has been observed that such sharing is not always advantageous and often leads to performance degradation [281, 274, 252, 81]. To address this challenge, several lines of research have emerged, each proposing different solutions.

One approach is the utilization of a split architecture with parallel backbones for individual tasks [172, 156, 68]. For instance, Misra et al. [172] introduced a cross-stitch module that intelligently combines task-specific networks, obviating the need for exhaustive search through numerous architectures.

The second line of work focuses on optimizing the pretraining process itself [281, 274, 252, 138]. For instance, Yu et al.[281] mitigate gradient interference by directly modifying the gradients using a technique known as "gradient surgery." Wang et al.[252] address interference by de-conflicting gradients through projection. Li et al. [138, 144] employ distillation to mitigate interference, although their approaches are constrained to retrained settings, specifically either single-task multi-source or single-source multi-task scenarios.

Another direction of research aims to develop systematic techniques for determining which tasks should be jointly trained in a multi-task neural network to avoid harmful conflicts between unrelated tasks [65, 12, 14, 125, 1]. While these methods focus on improving the individual task performances through multi-task learning, they do not explicitly consider the transfer performance on downstream tasks. In a related study, Polyvit [147] applies a vision transformer to multiple modalities, achieving impressive performance. However, it solely addresses the classification task for the image modality and relies on a simplistic hard-sharing approach, leaving the challenges of multi-task learning unresolved. Another recent work by Ghiasi et al. [73] adopts a semi-supervised learning approach and constructs cross-task pseudo-labels with task-specific teachers, creating a comprehensive multi-task dataset for pre-training. Nevertheless, this work only considers the single-source setting, and its student training still adheres to a hard-sharing regime.

Chapter 3

Improved Unsupervised Pretraining by Relative Contrastive Loss

3.1 Introduction

The emerging developments in the field of visual representation learning have underscored the superior capabilities of unsupervised learning, also denoted as self-supervised learning in certain studies [31, 87, 259]. These methodologies enable the understanding of visual representations without necessitating manual annotations [2, 186, 48, 113, 127, 288, 267, 220]. Contrastive learning, which lies at the core of recent unsupervised learning techniques, simultaneously optimizes deep networks by minimizing the distance between representations of positive pairs and enlarging the distance between negative pairs in the latent feature space [101, 87, 35, 37, 31, 259, 318, 80, 285, 36].

One of the pivotal aspects of contrastive learning methods is the construction of positive and negative pairs. Single-instance-positive methods such as MoCo [87, 35], SimCLR [31, 33], and BYOL [80], utilize random image augmentations to derive different views of the same sample as positive pairs, optionally considering the augmentations of other samples as negative pairs. Yet, such augmentations fail to provide positive pairs with natural intra-class variances. Clustering-based methods [24,

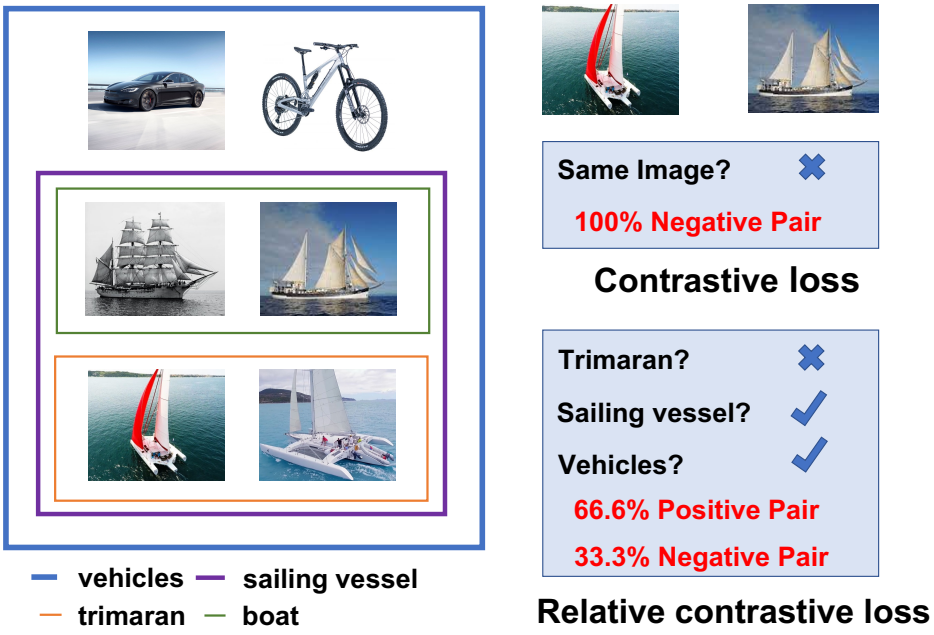


Figure 3.1: Motivation of the Relative Contrastive Loss. *Left*: Blue, purple and, orange rectangles denote vehicles, sailing vessels, and trimaran, respectively. The concepts of vehicles, sailing vessels, and trimarans show that the concepts of two images belonging to the same category depend on the level of hyponymy, motivating us to conduct relative contrastive learning in this chapter. *Right:* Any image pair in relative contrastive loss is determined positive or negative by multiple criteria.

22] and neighborhood-based methods [56, 209] mitigate these shortcomings by employing prototypes of pseudo-classes generated by clustering or k-nearest neighbors in the feature space as the positive samples. Nevertheless, these methods absolutely delineate positive and negative pairs, disregarding the relative nature of these concepts.

Contrasting previous self-supervised learning methods which dichotomously construct positive and negative pairs, human beings exhibit an ability to discern relative similarities. The Linnaean taxonomy system, developed by Swedish botanist Carl Linnaeus, elucidates the relative similarity of biological organisms across seven hierarchies [58]. Similarly, renowned benchmarks in computer vision such as ImageNet [45], iNat21 [233], and Places365 [309], respect the relative positiveness-negativeness and encompass hierarchical labels.

In response to this, the present study proposes a ground-breaking method for self-supervised learning that accounts for the inherent relativity in human recognition. The authors introduce a *relative contrastive loss*, which recognizes a sample pair as partially positive and negative based on a set of semantic criteria to capture real-world instance variation in a relative manner (Fig. 3.1(*right*)). This method involves inputting two images (query and key) into an encoder and momentum encoder to extract their corresponding features. The relatively positive-negative relations among these features are then determined by a set of criteria instantiated by hierarchical clustering, with each level in the clustering considered as a specific criterion. The proposed relative contrastive loss utilizes the query feature, the key feature, and their relatively positive-negative relations to supervise the training process.

The central contributions of this study are two-pronged. First, the authors introduce the innovative concept of relative contrastive loss for self-supervised learning. This concept represents a fresh approach to addressing the challenge of recognizing real-world instance variation. Second, they devise a framework incorporating online hierarchical clustering to instantiate this novel concept. The effectiveness of the proposed method is substantiated through a series of rigorous experiments. For instance, during ImageNet linear evaluation, the proposed method significantly enhances the top-1 accuracy of ResNet-50 by a gain of **+2.0%** (73.8% $\rightarrow$ 75.8%) compared with MoCov3. Furthermore, the experimental results underscore the efficacy of the proposed method for semi-supervised classification, object detection, and instance segmentation. This research thus adds a new dimension to the existing methodologies in self-supervised learning, opening avenues for a more nuanced understanding of visual representation learning.

3.2 Background: Contrastive Learning

Given an input image $\mathbf{x}$, two different augmentation parameters are employed to get two different images/views: image $\mathbf{v}$ and image $\mathbf{v}'$ for the query and the key branch, which output $\mathbf{q} = \mathcal{P}(\mathcal{Q}(\mathbf{v}, \theta), \theta_p)$ and

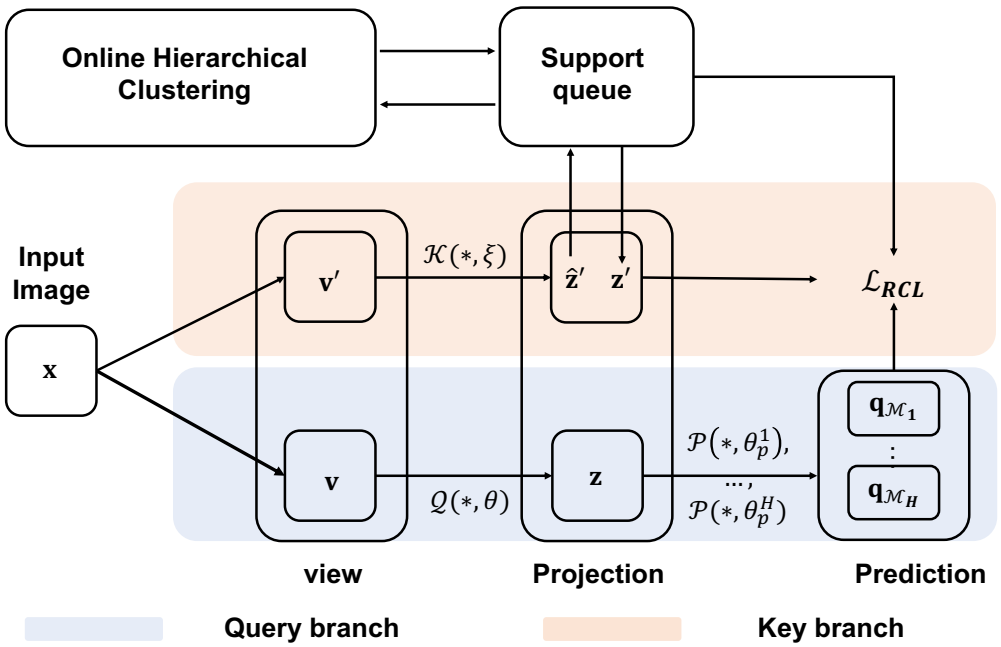


Figure 3.2: The pipeline of relative contrastive learning. In the key branch, the feature $\hat{\mathbf{z}}'$ after projection is used to search the relative keys $\mathbf{z}'$from the support queue by hierarchical clustering. For the feature $\mathbf{z}$ after projection in the query branch, we feed it into criterion-specific projectors to generate multiple predictions $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$. Multiple predictions, $\mathbf{z}$ and $\mathbf{z}'$ are then fed into the relative contrastive loss $\mathcal{L}_{RCL}$.

$\mathbf{z}' = \mathcal{K}(\mathbf{v}', \xi)$, respectively. Here, $\mathcal{Q}$ and $\mathcal{K}$ respectively denote feature transformations parameterized by θ and ξ. $\mathcal{P}$ is an optional prediction [36, 80, 186] of $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$ implemented by MLP. The contrastive loss is presented in InfoNCE [96], *i.e.*,

$$\mathcal{L}_{ctr}(\mathbf{x}, \theta) = -\log\left[\frac{\exp\left(\mathbf{q}^\top \mathbf{z}'\right)/\tau}{\exp\left(\mathbf{q}^\top \mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}^\top \mathbf{s}_k/\tau\right)}\right], \tag{3.1}$$

where $\mathcal{S} = \{\mathbf{s}_k | k \in [1, K]\}$ is a support queue storing negative features and $\tau = 0.1$ is the temperature. Contrastive loss pulls the features of the query-key pair $(\mathbf{q}, \mathbf{z}')$ together and pushes features of the query-negative pairs $(\mathbf{q}, \mathbf{s}_k)$ apart.

3.3 Method

We are interested in defining a query-key pair $(\mathbf{q}, \mathbf{z}')$ positive or negative relatively. Therefore we propose a relative contrastive loss and present an instantiation by online hierarchical clustering method to achieve it. Specifically, we generate a set of semantic criteria $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$ (H denotes the number of criteria) to define $(\mathbf{q}, \mathbf{z}')$ positive or negative by online hierarchical clustering (Sec. 3.3.3), and then compute our relative contrastive loss (Sec. 3.3.1). This loss (defined in Eq. 3.2) is obtained by aggregating the vanilla contrastive losses in Eq. 3.1 with $(\mathbf{q}, \mathbf{z}')$ defined as positive or negative by every semantic criterion $\mathcal{M}_i$ in $\mathcal{M}$.

Overview. As shown in Fig. 6.1, the relative contrastive learning has the following steps.

Step 1: Image $\mathbf{x}$ *to features* $\mathbf{z}$ *and* $\hat{\mathbf{z}}'$. Specifically, given two different views $(\mathbf{v}, \mathbf{v}')$ of an image $\mathbf{x}$, their projections can be computed by $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$ and $\hat{\mathbf{z}}' = \mathcal{K}(\mathbf{v}', \xi)$. Following [87, 80], the query branch $\mathcal{Q}(*, \theta)$ is a deep model updated by backward propagation, while the key branch $\mathcal{K}(*, \xi)$ is the same deep model as the query branch but with parameters obtained from the moving average of $\mathcal{Q}(*, \theta)$.

Step 2: Key-branch features $\hat{\mathbf{z}}'$ *to retrieved features* $\mathbf{z}'$. On the key branch, we retrieve key features $\mathbf{z}'$ from the support queue $\mathcal{S}$ with multiple criteria $\mathcal{M}$ implemented by hierarchical clustering (Sec. 3.3.3). On the query branch, similar to [80, 36], we add criterion-specific predictors $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_H\}$ on $\mathbf{z}$ to get $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$.

Step 3: Backpropagation using the relative contrastive loss. The retrieved feature $\mathbf{z}'$, multiple predictions $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$, and whether $(\mathbf{z}, \mathbf{z}')$ is positive or negative according to semantic criteria $\mathcal{M}$ (designed by online hierarchical clustering in Sec. 3.3.3) are then fed into the relative contrastive loss (Eq. 3.2).

3.3.1 Relative Contrastive Loss

In conventional contrastive learning, the positive-negative pairs are defined absolutely, *i.e.,* only augmentations of the same image are considered as positive pairs. Motivated by the relative recognition ability of human beings, we introduce a relative contrastive loss to explore the potential of relative positive samples defined in diverse standards.

Semantic Criteria for Assigning Labels. For a set of semantic criteria $\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$, relative contrastive loss determines any given query-key pair $(\mathbf{z}, \mathbf{z}')$ as positive or negative based on the criteria $\mathcal{M}_i$ for $i = 1, \ldots, H$. Denote $\mathcal{Y}_i(\mathbf{z})$ and $\mathcal{Y}_i(\mathbf{z}')$ respectively as the labels of $\mathbf{z}$ and $\mathbf{z}'$ generated using criterion $\mathcal{M}_i$. The query-key pair $(\mathbf{z}, \mathbf{z}')$ is defined positive under $\mathcal{M}_i$ if $\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')$, and negative under $\mathcal{M}_i$ if $\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')$. Different from the vanilla contrastive loss in Eq. 3.1, where $\mathbf{z}$ and $\mathbf{z}'$ are generated by different views of the same sample and naturally a positive pair, the $\mathbf{z}$ and $\mathbf{z}'$ in the relative contrastive loss can be generated by different samples and are considered positive or negative relatively. As an example in Fig. 3.1, the bicycle and the sailing ship have the same label when the semantic criterion is whether they are vehicles. Still, they have different labels when the semantic criterion is whether they are sailing vessels.

With the semantic criteria and their corresponding labels defined above, the relative contrastive loss is defined as

$$\mathcal{L}_{RCL}\left(\mathbf{z}, \mathbf{z}', \theta; \{\mathcal{M}_i\}_{i=1}^{H}\right) = \sum_{i=1}^{H} \alpha_i \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i), \tag{3.2}$$

where α_i is trade-off parameter among different criteria. $\alpha_i = 1/H$ in our implementation. Loss $\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)$ in Eq. 3.2 for criterion $\mathcal{M}_i$ can be defined as

$$\begin{aligned}
&\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i) \\
&= -\log\left[\frac{\mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] \cdot \exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')\right]}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}\right],
\end{aligned} \tag{3.3}$$

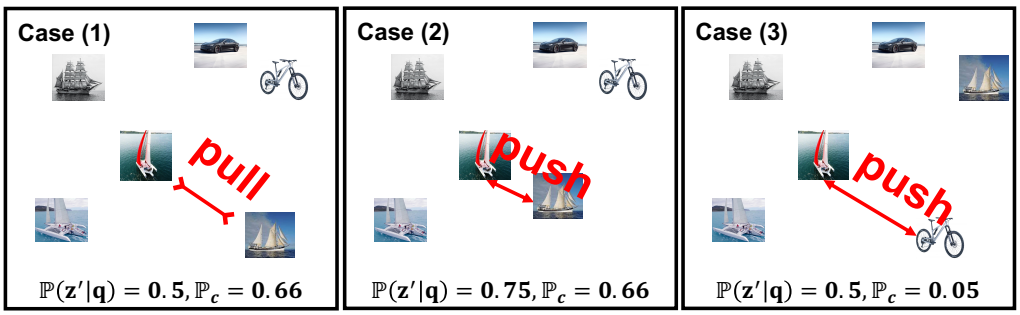


Figure 3.3: Analysis of the relative contrastive loss with multiple criteria. Both $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ and $\mathbb{P}_c$ represent the probability that $\mathbf{z}'$ and $\mathbf{q}$ have the same label. The difference is that $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ is based on the cosine similarity of $\mathbf{z}'$ and $\mathbf{q}$, and $\mathbb{P}_c$ is based on the set of defined semantic criteria. Whether to pull $(\mathbf{q}, \mathbf{z})$ together or push $(\mathbf{q}, \mathbf{z})$ apart is determined by $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c$. If $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c < 0$, $(\mathbf{q}, \mathbf{z})$ should be pulled together. If $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c > 0$, $(\mathbf{q}, \mathbf{z})$ should be pushed apart.

where $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$, $\mathbf{z}' = \mathcal{K}(\mathbf{v}', \xi)$, $\mathbf{s}_k$ is the feature in the support queue $\mathcal{S}$, K is the size of $\mathcal{S}$ and $\mathbb{I}(x)$ is an indication function, $\mathbb{I}(x) = 1$ when x is true, while $\mathbb{I}(x) = 0$ when x is false. $\mathbf{q}_{\mathcal{M}_i} = \mathcal{P}(\mathbf{z}, \theta_p^i)$ is the output of the criterion-specific predictor $\mathcal{P}(*, \theta_p^i)$ for the query projection $\mathbf{z}$, which is explained in the following.

Criterion-specific Predictor. Inspired by BYOL [80] and Simsiam [36], the predictor layer aims to predict the expectation of the projection $\mathbf{z}$ under a specific transformation. Therefore, we propose to use the multiple criterion-specific predictors, each of which is to estimate the expectation of $\mathbf{z}$ under its corresponding semantic criterion. Specifically, we add H MLPs, forming predictors $\{\mathcal{P}(*, \theta_p^1), \mathcal{P}(*, \theta_p^2), ..., \mathcal{P}(*, \theta_p^H)\}$ after the projectors in the query branch.

3.3.2 Analysis of Relative Contrastive Loss

In this section, we mathematically illustrate how relative contrastive loss supervises the feature distance between a query-key sample pair. We will show the feature distance of a image pair with higher possibility of being positive should be smaller than that with lower possibility of being positive.

We derive the gradient of our relative contrastive loss. The gradient of
$\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)$ in Eq. 3.3 is

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\partial \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)}{\partial \mathbf{q}_{\mathcal{M}_i}} \\ &= \left(\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right) \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\mathbf{z}'}{\tau} \\ &+ \sum_{k=1}^{K} \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) \frac{\mathbf{s}_k}{\tau},\end{aligned} \tag{3.4}$$

where

$$\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}'/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}'/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{s}_k/\tau\right)}, \tag{3.5}$$

$$\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{s}_k/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}')/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{s}_k/\tau\right)}. \tag{3.6}$$

The $\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right)$ and $\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right)$ above are the conditional probabilities of assigning the query prediction $\mathbf{q}_{\mathcal{M}_i}$ to the label of projection $\mathbf{z}'$ and the label of negative samples $\mathbf{s}_k$. We skip the analysis to the query-negative pair $(\mathbf{z}, \mathbf{s}_k)$ and focus on analyzing the dynamics between a uery-key pair $(\mathbf{z}, \mathbf{z}')$. Therefore, we drop the terms $(\mathbf{q}_{\mathcal{M}_i}, \mathbf{s}_k)$ in Eq. 3.4. When the gradient above for $\mathcal{L}$ is considered for the loss $\mathcal{L}_{RCL}$ defined in Eq. 3.2, $\mathbf{z}$ is optimized by gradient descent with the learning rate γ as

$$\mathbf{z} \leftarrow \mathbf{z} - \underbrace{\frac{\gamma}{\tau} \sum_{i=1}^{H} \alpha_i \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right) \mathbf{z}'}_{\eta}. \tag{3.7}$$

When $\eta > 0$, $\mathbf{z}$ and $\mathbf{z}'$ will be pushed apart, and when $\eta < 0$, $\mathbf{z}$ and $\mathbf{z}'$ will be pulled together. Following [230], we assume that $\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}$ is positive definite. Because γ, τ and α_i are positive, we define

$$\eta' = \sum_{i=1}^{H} \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right), \tag{3.8}$$

which is the only term that determines the sign of η.

In the following, we focus on η' for analyzing the dynamics of relative contrastive loss on network optimization in Eq. 3.7. We will reveal that the relativeness of positive-negative samples is based on 1) the probability $\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i})$ of assigning the query prediction $\mathbf{q}_{\mathcal{M}_i}$ to the label of projection $\mathbf{z}'$, and 2) the constructed criteria that determines the labeling function $\mathcal{Y}_i(\cdot)$.

Single Criterion. When there is only one criterion for determining query-key pairs positive or negative, *i.e.*, $\eta' = \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_1}) - \mathbb{I}\left[\mathcal{Y}_1(\mathbf{z}) = \mathcal{Y}_1(\mathbf{z}')\right]\right)$, our method collapses to the typical contrastive loss which pulls positive pairs close ($\mathbb{I}[\mathcal{Y}_1(\mathbf{z})=\mathcal{Y}_1(\mathbf{z}')] = 1$, $\eta' < 0$) and pushes negative pairs apart ($\mathbb{I}\left[\mathcal{Y}_1(\mathbf{z}) = \mathcal{Y}_1(\mathbf{z}')\right] = 0$ and $\eta' > 0$).

Multiple Criteria. When there are multiple criteria, to facilitate analysis, we assume the criterion-specific predictors are identical $\mathcal{P}_i = \mathcal{P}, i \leq H$ and thus predictions $\mathbf{q}_{\mathcal{M}_i} = \mathbf{q}, i \leq H$ are the same. With these assumptions, Eq. 3.8 is modified as

$$\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c), \tag{3.9}$$

where $\mathbb{P}_c = \sum_{i=1}^{H} \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] / H$ is possibility of $(\mathbf{z}, \mathbf{z}')$ being labeled by the H criteria as positive pair. We show the difference between the probability $\mathbb{P}_c$ define by the criteria and the probability $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ estimated from the model, *i.e.*, $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c$, will adaptively determine the relative decision of pushing or pulling. We use three different cases for illustration (Fig. 3.3). (1) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.50$ and $\mathbb{P}_c = 0.66$; (2) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.75$ and $\mathbb{P}_c = 0.66$; (3) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.50$ and $\mathbb{P}_c = 0.05$. **In case (1)**, $\mathbb{P}_c$ is large, *i.e.* most of the criteria label two samples as belonging to the same class. But $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.5$, *i.e.* the probability estimated from the learned features for $\mathbf{z}$ and $\mathbf{z}'$ belonging to the same class is not so high.

In this case, because the term $\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c)$ is negative, gradient descent will pull $\mathbf{z}$ towards $\mathbf{z}'$. **In case (2)**, since $\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c) > 0$, the loss will pull $\mathbf{z}$ and $\mathbf{z}'$ together. Comparing cases (1) and (2),

the loss changes its behavior from pushing samples away to pulling together because of the change of $\mathbb{P}(\mathbf{z}'|\mathbf{q})$. Cases (1) and (3) have the same estimated probability $\mathbb{P}(\mathbf{z}'|\mathbf{q})$. **In case (3)**, most of the criteria label the two samples as not belonging to the same class, *i.e.* $\mathbb{P}_c = 0.05$, and the loss will push $\mathbf{z}$ and $\mathbf{z}'$ away. Comparing cases (1) and (3), if the probability $\mathbb{P}_c$ defined by the criteria changes from high to low, the loss changes its behavior from pulling feature close to pushing features away.

3.3.3 Criteria Generation

We introduce an implementation of the semantic criteria $\mathcal{M}_{1:H} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$ used in the relative contrastive loss, where $\mathcal{M}_h$ is used for defining a query-key pair $(\mathbf{z}, \mathbf{z}')$ to be positive or negative. The criteria are implemented by online hierarchical clustering, which constrains the relativeness among different criteria with a hierarchy relationship, *i.e.*, $\mathcal{M}_1 \subset \mathcal{M}_2 \subset ... \subset \mathcal{M}_H$ (if $\mathcal{Y}_h(\mathbf{x}) = \mathcal{Y}_h(\mathbf{x}')$, then $\mathcal{Y}_j(\mathbf{x}) = \mathcal{Y}_j(\mathbf{x}'), \forall j > h$). At hierarchical clustering level h, a query-key pair $(\mathbf{x}, \mathbf{x}')$ in the same cluster are consider to be positive pair, *i.e.*, $\mathcal{Y}_h(\mathbf{x}) = \mathcal{Y}_h(\mathbf{x}')$. Inspired by [307], the implementation of hierarchical clustering is required to conform with the following property.

Online Cluster Refinement Stage. Initial clusters are not accurate due to the poor representations, and therefore need to be progressively adjusted along with the feature optimization. As illustrated in Fig. 3.4, for each training iteration t, the cluster refinement is conducted from the bottom to the top level, where a cluster contains the most samples. We take i-th cluster $\mathcal{C}_i^{h+1}$ at $(h+1)$-th level to elaborate the process of cluster split and merge.

Cluster Split. Cluster split aims to divide a cluster $\mathcal{C}_i^{h+1}$ into several smaller but more accurate clusters. To conform with the cluster preserve property, the basic units considered for splitting $\mathcal{C}_i^{h+1}$ are clusters in h-level whose samples all belong to $\mathcal{C}_i^{h+1}$, *i.e*, $\mathcal{U}_i^{h+1} = \{\mathcal{C}_j^h | \mathcal{C}_j^h \subset \mathcal{C}_i^{h+1}, j = 1, \ldots, k^h\}$, where k^h is the number of clusters in $\mathcal{H}_h$. Each unit in $\mathcal{U}_i^{h+1}$ is a cluster. When splitting $\mathcal{C}_i^{h+1}$ into m smaller clusters ($m < k^h$), m most dissimilar

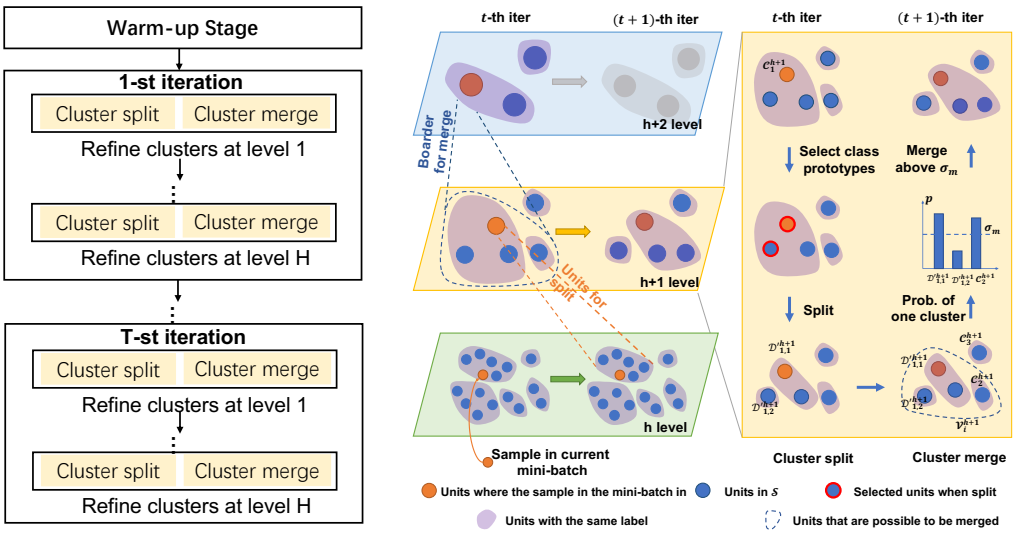


Figure 3.4: Online hierarchical clustering. The label refinement at $(h+1)$-th level from the t-th to the $(t+1)$-th iteration is constrained by labels at h-th level and $(h+2)$-th level. The clusters at the h-th level are the basic units for cluster split at the $(h+1)$-th level, and the clusters at the $(h+2)$-th level provides a boarder to identify clusters at the $(h+1)$-th level that may be merged.

split units in $\mathcal{U}_i^{h+1}$ are selected using the density peak selection algorithm [194] as the prototype of m different clusters, each of which contains one selected unit. The remaining units in $\mathcal{U}_i$ are merged to the m clusters according to their nearest prototype or label propagation [159]. With this procedure, cluster $\mathcal{C}_i^{h+1}$ is split into a set containing m divided clusters, denoted by $\mathcal{D}_i^{h+1} = \{\mathcal{D}_{i,j}'^{h+1}\}_{j=1}^{m}$.

Cluster Merge. Cluster merge aims to merge the divided clusters $\mathcal{D}_i^{h+1}$ and clusters at level $h+1$ if they are highly possible to one cluster. To conform with the cluster preserve property, we can only try to merge the clusters belonging to the same cluster $\mathcal{C}_{pa(i)}^{h+2}$ at the $(h+2)$-th level, where $\mathcal{C}_{pa(i)}^{h+2} \supset \mathcal{C}_i^{h+1}$ (clusters circled by the merge boarder in Fig. 3.4). Therefore, we construct a set of clusters that may be merged as $\mathcal{V}_i^{h+1} = \left\{\bigcup_j \mathcal{C}_{pa(j)=pa(i)}^{h+1}\right\} \bigcup \mathcal{D}_i^{h+1}$, and all elements in $\mathcal{V}_i^{h+1}$ belong to the same cluster $\mathcal{C}_{pa(i)}^{h+2}$. As shown in Fig. 3.4(Cluster merge), to merge clusters in $\mathcal{V}_i^{h+1}$, we compute the possibility of two clusters belonging to the same

class, *i.e.*, according to the distance of cluster centers or label propagation [96] (in Appendix C). Clusters whose possibilities of belonging to the same cluster are larger than a hyper-parameter σ_m will be merged.

3.4 Experiment

3.4.1 Implementation Details

Architecture. Our architecture is similar to MoCo-v2 and MoCo-v3. Compared with MoCo-v2, we use the symmetric loss proposed in BYOL [80] and add predictors after the projector in the query branch. Compared with MoCo-v3, we construct a negative queue as MoCo-v2. Specifically, we use ResNet-50 as our encoder following the common implementations in self-supervised learning literature. We spatially average the output of ResNet-50 which makes the output of the encoder a 2048-dimensional embedding. The projection MLP comprises 3 fully connected layers with output sizes $[2048, 2048, d]$, where d is the feature dimension applied in the loss, $d = 256$ if not specified. The projection MLP is *fc*-BN-ReLU for the first two MLP layers, and *fc*-BN for the last MLP layer. The architecture of the MLP predictor is 2 fully-connected layers of output size $[4096, d]$, which can be formulated as $fc_2(ReLU(BN(fc_1)))$.

Training. For fair comparison, we train our relative contrastive learning method on the ImageNet2012 dataset [45] which contains 1,281,167 images without using any annotation or class label. In the training stage, we train for 200, 400 and 800 epochs with a warm-up of 10 epochs and cosine annealing schedule using the LARS optimizer [276] by the relative contrastive loss Eq. 3.2. The base learning rate is set to 0.3. Weight decay of 10^{-6} is applied during training. As is common practice, we do not use weight decay on the bias. The training settings above are the same as BYOL. We also use the same data augmentation scheme as BYOL. We set temperature τ for loss computation in Eq. 3.2 to 0.1.

Method	Arch.	Epochs	Top1	Top5	Method	Arch.	epochs	Top1	Top5
ODC [287]	R50	100	57.6	-	PIRL [171]	R50	800	63.6	-
InstDisc [259]	R50	200	58.5	-	MoCo v2 [35]	R50	800	71.1	-
LocalAgg [318]	R50	200	58.8	-	SimSiam [36]	R50	800	71.3	90.7
MSF [209]	R50	200	71.4	-	SimCLR [31]	R50	800	69.3	89.0
MSF w/s [209]	R50	200	72.4	-	SwAV [24]	R50	800	71.8	-
CPC v2 [93]	R50	200	63.8	85.3	BYOL [80]	R50	1000	74.3	91.6
CMC [227]	R50	240	66.2	87.0	InfoMin Aug. [228]	R50	800	73.0	91.1
Adco [186]	R50	200	68.6	-	MoCo v3 [37]	R50	800	73.8	-
NNCLR [56]	R50	200	70.7	-	NNCLR [56]	R50	800	75.4	92.4
RCL (Ours)	R50	200	72.6	90.8	RCL (Ours)	R50	800	75.8	92.6

Table 3.1: Comparison with other self-supervised learning methods under the linear evaluation protocol [87] on ImageNet. We omit the result for SwAV with multi-crop for fair comparison with other methods.

Method	ImageNet 1%		ImageNet 10%	
	Top1	Top5	Top1	Top5
Supervised baseline [286]	25.4	48.4	56.4	80.4
Pseudo label [128]	-	-	51.6	82.4
UDA [266]	-	-	68.8†	88.5†
FixMatch [208]	-	-	71.5†	89.1†
MPL [185]	-	73.5†	-	-
InstDisc [259]	-	39.2	-	77.4
PCL [132]	-	75.6	-	86.2
SimCLR [31]	48.3	75.5	65.6	87.8
BYOL [80]	53.2	78.4	68.8	89.0
SwAV (multicrop) [24]	53.9	78.5	70.2	89.9
Barlow Twins [285]	55.0	79.2	69.7	89.3
NNCLR [56]	56.4	80.7	69.8	89.3
RCL (Ours)	57.2	81.0	70.3	89.9

Table 3.2: Comparison with the state-of-the-art methods for semi-supervised learning. Pseudo Label, UDA, FixMatch and MPL are semi-supervised learning methods. † denotes using random augment [43]. We use the same subset as in SwAV.

3.4.2 Comparison with State-of-the-art Methods

Linear Evaluations. Following the standard linear evaluation protocol [259, 318, 87, 35], we train a linear classifier for 90 epochs on the frozen 2048-dimensional embeddings from the ResNet-50 encoder using LARS [276] with cosine annealed learning rate of 1 with Nesterov momentum of 0.9 and batch size of 4096. Comparison with state-of-the-art methods is presented in Tab. 3.1. Firstly, our proposed RCL performs better than other state-of-the-art methods using a ResNet-50 encoder without multi-crop augmentations. Specifically, RCL improves MoCo v2 by 4.7% and MoCo v3 by 2.0%, which generates positive samples by implementing a different augmentation on the query image. Furthermore, our method is better than InfoMin Aug., which carefully designs the "good view" in the contrastive learning for providing positive samples by 2.8%. The significant improvements empirically verifies one of our motivations: manually designed augmentations cannot cover the visual variations in a semantic class. Compared with other state-of-the-art methods, our method also achieves higher performance than BYOL by

1.5%. Clustering-based methods, *e.g.*, SwAV [24], and nearest-neighbor-based methods go beyond *single positives*. Clustering-based methods utilize cluster prototypes as positive samples. However, our method also achieves 4.0% improvement without multi-crop augmentation. SwAV leverages an online clustering algorithm and uses only its cluster centers as its positives, which ignores the relative proximity built by our relative contrastive loss. NNCLR [56] is the recent state-of-the-art method, which utilizes the nearest neighbor as the positive sample. Our method is better than NNCLR at 200 epochs are comparable at 800 epochs because NNCLR defines positive samples without relativeness. Furthermore, our RCL can be the same as NNCLR when we set only one criterion and only cluster the nearest neighbor. We also compare our method with existing methods in various epochs, presented in Fig 3.5 (a). Our method performs better than SimCLR, Simsiam, MoCo-v3 and BYOL for 200, 400, and 800 epochs.

Semi-Supervised Learning Evaluations. To further evaluate the effectiveness of the learned features, we conduct experiments in a semi-supervised setting on ImageNet following the standard evaluation protocol [33, 31], thus fine-tuning the whole base network on 1% or 10% ImageNet data with labels without regularization after unsupervised pre-training. The experimental results are presented in Tab. 3.2. Firstly, our method outperforms all the compared self-supervised learning methods with the semi-supervised learning setting on ImageNet 1% subset, even when compared with the SwAV method with strong multi-crop augmentation (our RCL does not use multi-crop augmentation). Second, in the ImageNet 10% setting, our method still results better than most popular self-supervised learning methods, such as SimCLR, BYOL, Barlow Twins and NNCLR. The results indicate the good generalization ability of the features learned by our relative contrastive loss.

Transfer to Detection and Segmentation Tasks In this section, we provide the detection and segmentation results[1], when we transfer our model

[1]These experiments are not the improved version of the method RCL, just the generalization ability evaluation of the method.

Method	Object Detection		Instance Segmentation	
	AP-all bb	AP-50 bb	AP mk	AP-50 mk
Supervised	38.2	58.2	33.3	54.7
MoCo v2	**39.3**	58.9	**34.4**	55.8
SwAV	37.9	57.6	33.1	54.2
Simsiam	39.2	**59.3**	**34.4**	56.0
Barlow Twins	39.2	59.0	34.3	56.0
RCL	**39.3**	59.1	34.3	**56.1**

Table 3.3: Transfer learning from ImageNet with standard ResNet50 to COCO object detection and instance segmentation. All methods are evaluated on the test-dev dataset. bb: bounding box. mk: segmentation mask.

to detection and segmentation tasks. We strictly follow the evaluation protocol in MOCO [87]. Specifically, we do not freeze the batch normalization layer, and finetune the whole network by the COCO training set. We report the results on the COCO evaluation dataset in Table 3.3.

3.4.3 Ablation Study

Default Settings. The size of the support set $\mathcal{S}$ is set to be 1.5×2^{16} and the batch size of our algorithm is 4096. We train for 200 epochs with a warm-up of 10 epochs. The learning rate is 0.3, and we leverage the cosine annealing schedule using the LARS optimizer [276]. The results in this section are tested by linear evaluations on ImageNet.

Different Clustering Methods. To illustrate the effectiveness of our online hierarchical clustering method, we compare it with K-means and DBSCAN. Because both K-means and DBSCAN are offline clustering methods, we extract the features of all images in ImageNet-1K, and conduct clustering on these features before each epoch. For K-means, we set the number of clusters to (250000, 500000, 1000000), where we verify there are about 73.88% samples that conform to the hierarchy in Sec. 3.3.3. For DBSCAN, we keep the minimum number of samples within r to 4, and select $r = 0.8, 0.7, 0.6$ to construct hierarchical label banks, leading to 97.3% samples conforming the hierarchy. As shown in Tab. 3.5, we can

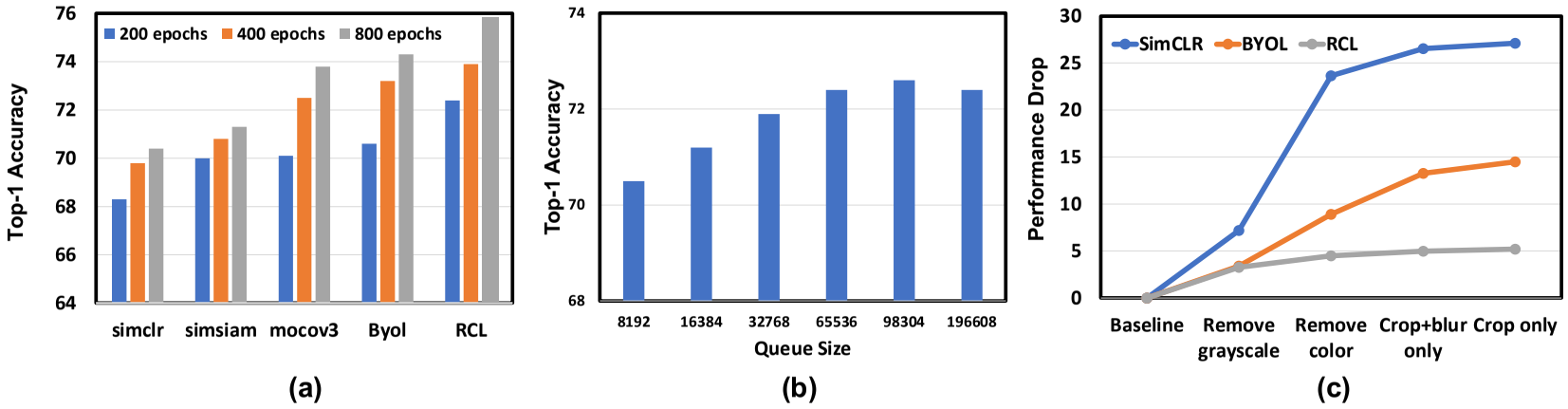


Figure 3.5: Ablation studies. (a) Comparison with state-of-the-art methods when training 200, 400 and 800 epochs under linear evaluation on ImageNet. (b) ImageNet top-1 accuracy with different sizes of the support queue. (c) Top-1 Accuracy drop (Y-axis) by removing augmentations (X-axis).

No.	#Predictors	#Hierarchies	Top1
1	1	1	70.2
2	1	2	71.3
3	1	3	71.8
4	1	4	71.4
5	3	3	72.6

Table 3.4: Ablation studies on multiple predictions and the number of levels in the hierarchical clustering. #Predictors: number of criterion-specific predictors. #Hierarchies: number of levels in the hierarchical clustering.

see that K-means improves the NNCLR by 1.1%, which verifies the effectiveness of relativeness. DBSCAN is better than K-means, which verifies the effectiveness of the hierarchical labels. Our online hierarchical clustering is better than above methods, because it can refine labels along with network optimization, which avoids the problem that label refinement is slower than network optimization when using offline clustering. Our online hierarchical clustering is faster than offline clustering algorithms, *e.g.,* K-means and DBSCAN, because it only deals with samples in the current mini-batch while K-means and DBSCAN needs to operate on the whole dataset. Compared with NNCLR, our method is about 18% slower but shows better performance on 200 epochs setting.

Number of Levels in Online Hierarchical Clustering. To assess the effectiveness of relativeness, we ablate on a different number of levels in the hierarchical label bank. As illustrated in Tab. 3.4, the top-1 accuracy improves from 70.2% to 71.3% by 1.1% when we change the number of levels, which indicates the adding relativeness can benefit the contrastive learning in self-supervised image classification tasks. When we continue to increase the number of levels, we can see the top-1 accuracy improves by 0.5% from 2 levels to 3 levels but will decrease to 71.4% when we change 3 levels to 4 levels. This phenomenon motivates us to design more appropriate criteria for future work when implementing relative contrastive loss in the feature.

Method	Hierarchy	Online	Top1	Time / Ep
No (NNCLR [56])	x	x	70.7	659s
K-means	low	x	71.8	1056s
DBSCAN	high	x	72.3	986s
Online Hierarchical Clustering	✓	✓	72.6	776s

Table 3.5: Ablation studies on different clustering methods. Mixed precision time for training 1 epoch using 64 GeForce GTX 1080 Tis with 64 samples in each GPU is reported.

Multiple Predictors. Multiple predictors are used to predict the multiple projection expectations $\{\mathbb{E}_{\mathcal{T}_1}(\mathbf{z}_\theta), \mathbb{E}_{\mathcal{T}_2}(\mathbf{z}_\theta), ..., \mathbb{E}_{\mathcal{T}_H}(\mathbf{z}_\theta)\}$ based on the various image transformations that wilBl not change the label under different criteria. When implementing a single predictor after the projection, we actually impose to predict the expectation of the projection regardless of the semantic criterion. When using multiple predictors, we impose each predictor to predict the projection expectation based on the image transformation that will not change the label under a specific criterion. Comparing Exp. 3 and Exp. 5 in Tab. 3.4, we can conclude that multiple predictors can outperform single predictor by 0.7%.

Size of Support Queue. Similar with MoCo that utilizes a memory bank to store the representations of other samples, our method has a support queue to provide diverse image variations. We evaluate the performance of our method with different support queue size in Fig. 3.5(b). As can be observed, when the size of the support queue increases to 98304, the performance of our method also improves, reflecting the importance of using more diverse variation as positive samples. Specifically, increasing the size from 65536 to 98304 leads to 0.36% top-1 accuracy improvement. However, further increasing the size of the support queue does not provide further improvement.

Sensitivity to Augmentations. Previous methods leverage the manually designed augmentations to model the visual variation between a semantic class, and therefore augmentations are very critical to their

σ_m	num of clusters (H=3)	linear evalutaion
0.10	32	15.4
0.30	565	37.2
0.40	2752	43.2
0.50	5253	64.3
0.55	8795	70.9
0.60	9321	**72.6**
0.70	23246	72.3
0.80	38842	72.2
0.90	89642	71.5

Table 3.6: Sensitivity of Cluster Merge Threshold σ_m

self-supervised learning methods. In contrast, we utilize similar samples/images in the dataset to be positive samples. As illustrated in Fig. 3.5(c), Our proposed RCL is much less sensitive to image augmentations when compared with SimCLR and BYOL.

Sensitivity of the Cluster Merge Threshold σ_m. The cluster merge threshold is a hyper-parameter and determines when two clusters can be merged. In this part, we analysis the influence of σ_m to the model's performance under the linear evaluation setting. To ease the hyper-parameter tuning process, we simply set the merge threshold the same throughout all levels in the hierarchical clustering. As can be observed in Table 3.6, when σ_m is small, there are only a few clusters in the last level and the linear classification results are very bad. We attribute the failure to too many samples wrongly grouped in same cluster. Even when we set $\sigma_m = 0.4$ (the number of clusters equal to 2752), when linear evaluation results are still poor, which indicates the large number of noisy labels in the hierarchical label bank. The model achieves best performance when setting σ_m to 0.6, which leads to a moderate cluster size compared to $\sigma_m = 0.1$ and $\sigma_m = 0.9$. The results demonstrate that it is important to make a good balance between learning more diverse semantic variance and maintain suitable discriminative ability. Besides, we also observe that the accuracy change is small when the threshold σ_m is larger than 0.55, showing that the model is not sensitive to the value of σ_m when it is large enough.

No.	#Predictors	#Hierarchies	LP	Acc
1	1	1	Yes	70.2
2	1	2	Yes	**71.3**
3	1	2	No	68.5
4	1	3	Yes	**71.8**
5	1	3	No	65.5
6	1	4	Yes	**71.4**
7	1	4	No	67.8
8	3	3	Yes	**72.6**

Table 3.7: The effectiveness of using label propagation and number of hierarchies.

Clusters with Different Epochs. To explore how the number of clusters changes with the increase of training epochs, we depict the number of clusters in the support set $\mathcal{S}$ with different training epochs in Figure 3.6. We find the number of clusters decreases consistently during the network training, which demonstrates that the network can learn semantic knowledge from the dataset. Besides, with the increase of hierarchical level, the number of clusters decrease, which obeys the *cluster preserve property* of the hierarchical label. To further understand the process of hierarchical clustering, we illustrate the clustering results of the support set $\mathcal{S}$ in Figure 3.7. As shown in the figure, positive samples at lower hierarchical level (e.g. $H = 1$) are more similar to each other, while positive samples at higher hierarchical level (e.g. $H = 3$) are more dissimilar to each other visually and semantically. For example, at epoch 200, the $H = 1$ positives share the same color, shape, and semantic meaning (mushroom) with query image, but the $H = 3$ positives only share the similar shape with the query image but have different colors and possibly different semantic meanings (mushroom v.s. rockets). With the increase of training epochs, samples in the same hierarchical level are more similar to each other visually and semantically, because the CNN features are learned better..

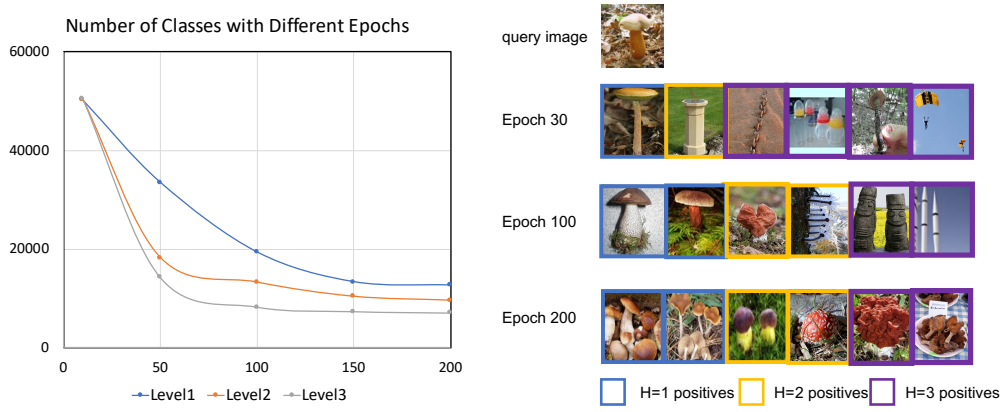


Figure 3.6: Left: Number of classes with different epochs. Blue line, orange line and gray denote the number of clusters in the hierarchical label bank at $H = 1$, $H = 2$ and $H = 3$, respectively. **Right:** Visualization of positives samples at different levels in the hierarchical label bank. Comparing with images in the hierarchical label at different epochs (epoch=30, 100, 200), the samples in the hierarchical label bank at all levels are becoming more and more visually similar with the query image. When we focus on the samples in only one epoch, we find that with the increase of level of the hierarchical label bank, the number of images increases, the images are less visually similar to the query image than those in the hierarchical label bank at relative low levels. Specifically, we find $H = 1, 2$ positives are more similar to the query image than $H = 3$ positives.

3.5 Limitations and Conclusions

In this chapter, we propose a new relative contrastive loss for unsupervised learning. Different from the typical contrastive loss that defines a query-key pair to be positive or negative, relative contrastive loss can treat a query-key pair relatively positive, which is measured by a set of semantic criteria. An online hierarchical clustering in our method instantiates the semantic criteria. Representations learned by the relative contrastive loss can capture diverse semantic criteria motivated by human recognition and better fit the relationship among samples. Extensive results on self-supervised learning, semi-supervised learning, and transfer learning settings show the effectiveness of our relative contrastive loss. While our relative loss primarily benefits from multiple criteria, the optimal criteria design is still under-explored.

Chapter 4

Unified Unsupervised Pretraining Pipeline and Improved Data Augmentation from a Graph Perspective

4.1 Introduction

Self-supervised learning (SSL) has recently been a focal point of research within the computer vision discipline, with an array of pertinent studies emerging in this sphere [2, 186, 48, 113, 127, 288, 267, 253]. One particular framework that has emerged as central to modern unsupervised learning methodologies is contrastive learning [101, 87, 35, 37, 31, 259, 318, 80, 285, 36]. It aims to minimize the distance between augmented views of the same image (positive samples) while maximizing distance between different images (negative samples), offering a compelling potential to elicit robust visual representations that rival those obtained via supervised learning. Moreover, this approach has proven superior performance in various visual tasks when models are pretrained without labels.

Recent contrastive SSL methodologies typically employ a Siamese architecture bifurcated into online and target branches. Each branch comprises a backbone, a projector layer, and optionally, a predictor layer.

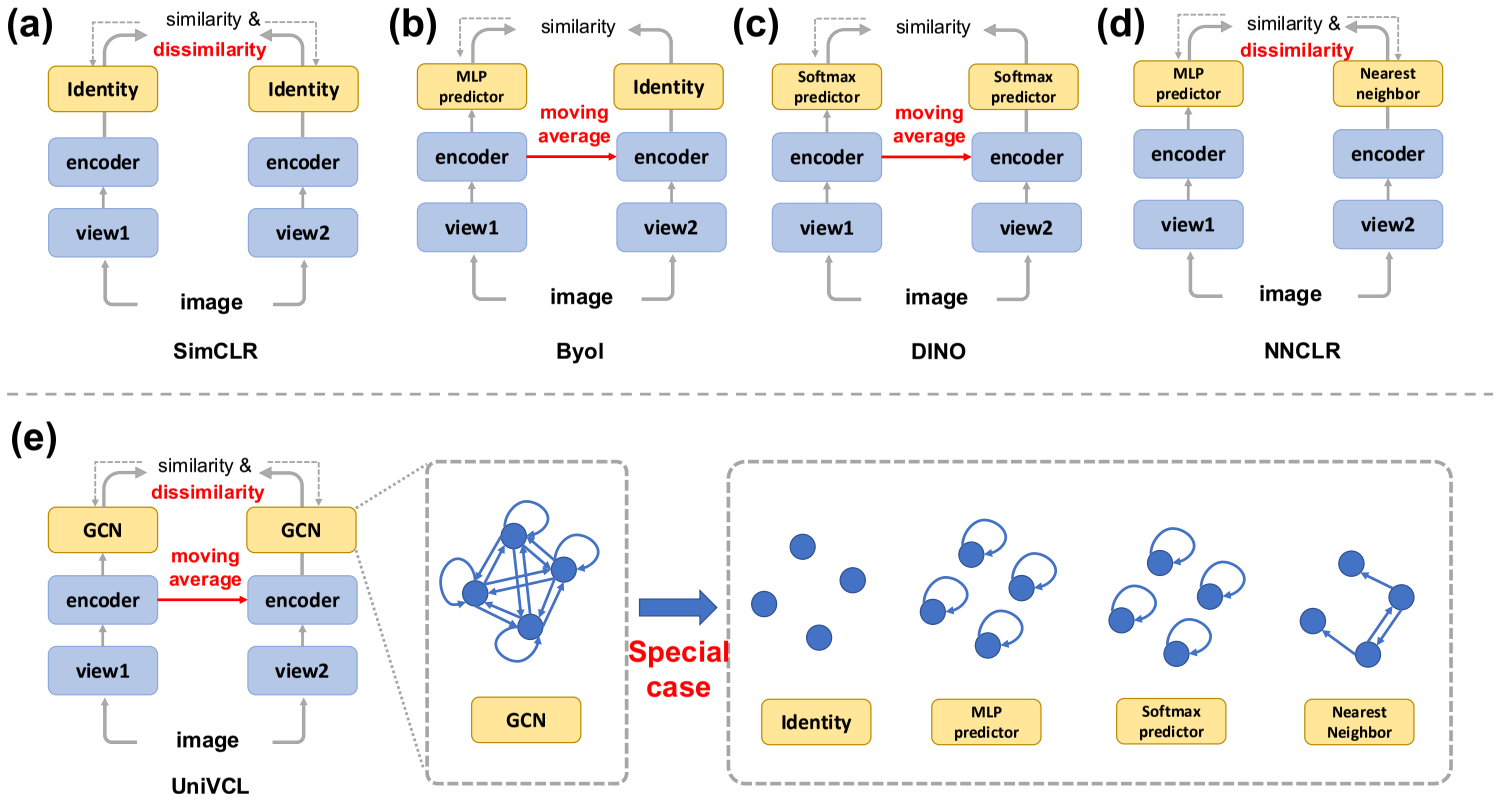


Figure 4.1: (a-d): Existing self-supervised learning methods share similar encoder designs but have highly different predictors. The encoder in the picture includes a backbone and a projector. (e): Our UniVCL unifies different designs by a GCN layer, which has neighborhood aggregation and self-loop terms. The arrow denotes the aggregation operation. Specifically, the MLP and Softmax predictors are self-loop terms with different activation functions. The nearest neighbor retrieval can be viewed as the neighborhood aggregation term in the GCN layer.

Although there is a degree of uniformity across the backbone and projector layer processes in these works, there are pronounced discrepancies in their predictor layer designs, as depicted in Fig. 4.1(a-d).

In essence, four types of predictor layers have been identified: 1) SimCLR [31], MOCOv1 [87], MOCOv2 [35], which do not apply any operation post the projector layer, thereby being mathematically analogous to an identity layer as the predictor; 2) BYOL [80], SimSiam [36], and MOCOv3 [37], which utilize an MLP predictor on the online branch; 3) DINO [25], SEED [62], and TWIST [241] leverage a softmax layer on both online and target branches; and 4) cutting-edge methodologies such as NNCLR [56] and MSF [120] engage a K-nearest neighbor layer on the target branch for contrastive learning. Although these designs appear vastly dissimilar in their representational learning approaches, they can potentially be consolidated into a unified framework with the aid of graph neural network-based modifications to predictor design.

In light of these observations, we propose the Unified Vision Contrastive Learning (UniVCL) framework. This model comprehensively represents the four types of contrastive-based SSL methodologies identified above, as seen from a graph-centric perspective. The projected feature and its K nearest neighbors in the feature space are modeled as graph nodes in our framework. The Graph Convolution Network (GCN) layer [114, 236, 269], with its self-loop term and neighborhood aggregation term, can formulate diverse predictor designs as its specialized variant (refer to the second row of Fig. 4.1). The identity mapping, the MLP layer, and the softmax layer can be construed as the self-loop term, differing only in activation functions. In contrast, the K-nearest neighbor retrieval corresponds to the neighborhood aggregation term in the GCN layer. From this perspective, we comprehensively evaluate different predictor designs in extant methods, maintaining the same learning schedules, data augmentations, and objective functions. Our meticulous and equitable experiments yield three notable observations. First, the neighborhood aggregation term in the GCN layer significantly enhances linear evaluation performance by **+2.1%**. Second, the activation function substantially influences linear evaluation performance, with non-linear

activation functions resulting in approximately **+0.4%** performance gain over identity activation functions. Lastly, the performance variance between different non-linear activation functions is negligible, given that the other components of the GCN layer are well-engineered, suggesting that the selection of the non-linear activation function is a relatively insignificant factor.

Further, this graph-centric unified framework facilitates a link between vision SSL and graph SSL, the latter being another active research area in SSL. This connection enables the exploration of the efficiency of various pretext tasks in graph SSL for vision SSL. Specifically, the novel data augmentation technique, namely, graph augmentation, proven effective in graph SSL, can be extended to vision SSL. Graph augmentation introduces feature variations in accordance with the graph structure, thereby regulating network optimization. Our investigation on ImageNet-1K reveals that graph augmentation employing message passing throughout the network can bolster the efficiency of self-supervised learning methods in image classification by **+0.9%**. Our comprehensive analysis, employing the purity metric [120], confirms that these augmentations introduce edge noise in the GCN predictor, consequently directing the encoder towards learning more robust image features.

In summation, our contributions can be categorized into three segments: **(1)** We introduce a general framework (UniVCL) to amalgamate recent leading contrastive learning methodologies in the domain of vision SSL. **(2)** Through detailed and impartial experimental comparisons, we underline the significance of the neighborhood aggregation term and the non-linear activation function of the GCN layer in the UniVCL framework. **(3)** Capitalizing on the graph design of UniVCL, we establish a connection between vision SSL and graph SSL and introduce typical graph augmentations into self-supervised image classification, a strategy empirically validated to augment linear evaluation and semi-supervised learning performances.

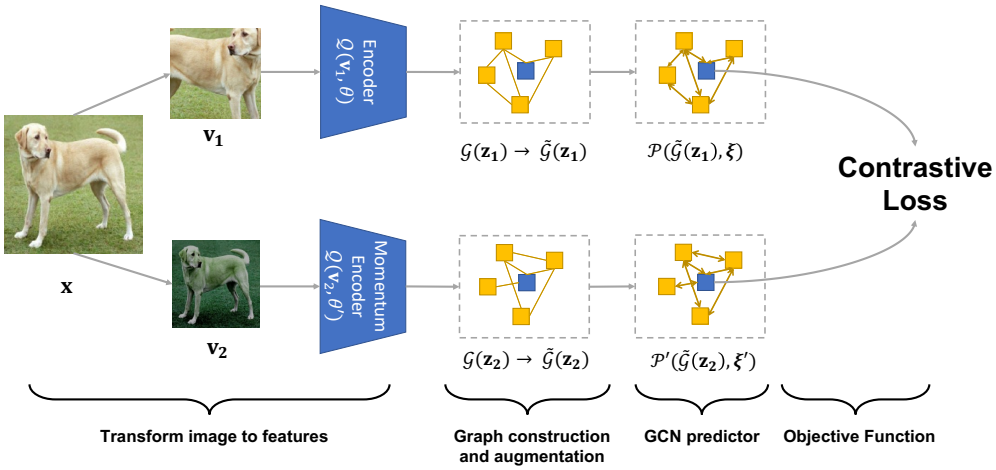


Figure 4.2: The framework of UniVCL. It includes four steps. First, Given an image $\mathbf{x}$, two augmented views $\mathbf{v}_1$ and $\mathbf{v}$ are generated. Then the features $\mathbf{z}_1$ and $\mathbf{z}_2$ are extracted by encoder $\mathcal{Q}(*, \theta)$ and $\mathcal{Q}(*, \theta')$, respectively. Second, we retrieve the K nearest neighborhood samples of $\mathbf{z}_1$ and $\mathbf{z}_2$ from the support queue $\mathcal{S}$, forming graph $\mathcal{G}(\mathbf{z}_1)$ and $\mathcal{G}(\mathbf{z}_2)$ respectively. Then, we implement the graph augmentation on $\mathcal{G}(\mathbf{z}_1)$ and $\mathcal{G}(\mathbf{z}_2)$, generating augmented graphs $\widetilde{\mathcal{G}}(\mathbf{z}_1)$ and $\widetilde{\mathcal{G}}(\mathbf{z}_2)$. Third, we input the augmented graph into the GCN predictor layer, generating the predicted features $\mathbf{q}_1$ and $\mathbf{q}_2$. Last, we compute the alignment loss based on $\mathbf{q}_1$ and $\mathbf{q}_2$. The encoder denotes both the backbone network and the projector layer.

4.2 UniVCL

We are interested in using the Graph Convolution Network (GCN) to unify different predictor designs in different methods. Specifically, we maintain a support queue $\mathcal{S} = \{\mathbf{s}_i | i \in [1, \cdots, m]\} \in \mathbb{R}^{m \times d}$ in the same way as [56], where m is the size of the support queue, and d is the feature dimension. We only use embeddings from the target view to update the support set. As shown in Fig. 4.2, our proposed UniVCL has the following steps.

Step1: *Transform Image to Features.* Specifically, given two different augmented views $(\mathbf{v}_1, \mathbf{v}_2)$ of an image $\mathbf{x}$, the features of two views can be computed by $\mathbf{z}_1 = \mathcal{Q}(\mathbf{v}_1, \theta)$ and $\mathbf{z}_2 = \mathcal{Q}(\mathbf{v}_2, \theta')$, respectively. Following [87, 80, 35, 37], the online branch $\mathcal{Q}(*, \theta)$ is a neural network updated by backward propagation, while the target branch $\mathcal{Q}(*, \theta')$ is a network with the same architecture as the online branch but with parameters obtained from the moving average of $\mathcal{Q}(*, \theta)$.

Step2: Graph Construction and Augmentation (Sec. 4.2.3). Give $\mathbf{z}_1$ and $\mathbf{z_2}$ from Step 1, we respectively construct the fully connected graph $\mathcal{G}(\mathbf{z}_1)$ and $\mathcal{G}(\mathbf{z}_2)$, where nodes in $\mathcal{G}(\mathbf{z}_1)$ and $\mathcal{G}(\mathbf{z}_2)$ are K nearest neighbors of $\mathbf{z}_1$ and $\mathbf{z}_2$ in support queue $\mathcal{S}$, respectively. Then we implement typical graph augmentations (Sec. 4.2.3) to generate the augmented graphs $\widetilde{\mathcal{G}}(\mathbf{z}_1)$ and $\widetilde{\mathcal{G}}(\mathbf{z}_2)$.

Step 3: *GCN Predictor (Sec. 4.2.1).* The augmented graphs $\widetilde{\mathcal{G}}(\mathbf{z}_1)$ and $\widetilde{\mathcal{G}}(\mathbf{z}_2)$ are respectively transformed to prediction features $\mathbf{q}_1$ and $\mathbf{q}_2$ through GCN predictors $\mathcal{P}(*, \xi)$ and $\mathcal{P}'(*, \xi')$.

Step 4: *Backward propagation using the alignment loss.* Given the prediction features $\mathbf{q}_1$ and $\mathbf{q}_2$, the parameters θ in $\mathcal{Q}(*, \theta)$ and the parameters ξ, ξ' in $\mathcal{P}(*, \xi)$, $\mathcal{P}'(*, \xi')$ are learned by using alignment loss. In our design, the alignment loss is implemented as the contrastive, *i.e.,*

$$\mathcal{L} = -\log \frac{\exp(\mathbf{q}_1^\top \mathbf{q}_2)}{\exp(\mathbf{q}_1^\top \mathbf{q}_2) + \sum_{i=1}^{m} \exp(\mathbf{q}_1^\top \mathbf{s}_i)}, \tag{4.1}$$

where $\mathbf{s}_i$ is the feature stored in the support queue $\mathcal{S}$.

4.2.1 General Predictor Layers as GCNs

Since UniVCL appends the GCN predictor after the encoder $\mathcal{Q}$ and $\mathcal{Q}'$, we only analyze the GCN predictor $\mathcal{P}$ on the online branch. All analysis below is applicable to the GCN predictor $\mathcal{P}'$ on the target branch.

Given the feature $\mathbf{z}_1$ obtained by $\mathcal{Q}$, the input of the GCN layer is defined as

$$\mathcal{G}(\mathbf{z}_1) = (\mathbf{z}_1, \mathcal{N}_1(\mathbf{z}_1), \mathcal{N}_2(\mathbf{z}_1), ..., \mathcal{N}_K(\mathbf{z}_1)) \tag{4.2}$$

where K is the number of samples retrieved from $\mathcal{S}$, and $\mathcal{N}_i(\mathbf{z}_1)$ denotes the features of the i-th nearest neighbor in the support queue $\mathcal{S}$. The GCN predictor $\mathbf{q}_1 = \mathcal{P}(\mathcal{G}(\mathbf{z}_1))$ can be represented as a stack of graph convolution layers $\mathcal{F}_l$, where l is the layer index, *i.e.*, $\mathcal{P}(\mathcal{G}(\mathbf{z}_1), \theta) = \mathcal{F}_L(\mathcal{F}_{L-1} \cdots (\mathcal{F}_2(\mathcal{F}_1(\mathcal{G}(\mathbf{z}_1)))))$, where L is the number of stacked GCN layers. Here, $\mathcal{F}_l$ is presented as

$$\mathbf{F}_{l+1} = \mathcal{F}_l(\mathbf{F}_l) = \sigma_l(\underbrace{\mathbf{W}_l \mathbf{A} \mathbf{F}_l}_{\text{neighborhood aggregation}} + \underbrace{\mathbf{W}'_l \mathbf{F}_l}_{\text{self-loop}}), \tag{4.3}$$

where the affinities $\mathbf{A} = \{a_{i,j}\} \in \mathbb{R}^{(K+1)\times(K+1)}, 0 \leq i, j \leq K$ are defined as

$$a_{i,j} = \begin{cases} \mathcal{N}_i(\mathbf{z}_1)^\top \mathcal{N}_j(\mathbf{z}_1), & i \neq j, \\ 0, & i = j, \end{cases} \tag{4.4}$$

where $e_{i,j}$ is the affinity between $\mathcal{N}_i(\mathbf{z}_1)$ and $\mathcal{N}_j(\mathbf{z}_1)$, and we denote $\mathcal{N}_0(\mathbf{z}_1) = \mathbf{z}_1$.

4.2.2 Unifying Unsupervised Contrastive Learning

As UniGrad [222] has explored the equivalence of different objective functions in the existing methods both theoretically and experimentally, we focus on the predictor designs among these different self-supervised learning methods.

As shown in Tab. 4.1 and Fig. 4.1(a-d), the predictor designs of different self-supervised learning methods can be categorized into four types: the identity predictor, the MLP predictor, the Softmax predictor, and the nearest neighbor predictor. Based on the the formal formulation of a

Method	Venue	Type	Online Branch	Target Branch
MOCOv2	Arxiv'21	a	identity	identity
SimCLR	ICML'20	a	identity	identity
Barlow Twins	ICML'21	a	identity	identity
MOCOv3	ICCV'21	b	MLP	identity
BYOL	NeurIPS'20	b	MLP	identity
SimSiam	CVPR'21	b	MLP	identity
DINO	ICCV'21	c	softmax	softmax
SEED	ICLR'21	c	softmax	softmax
TWIST	Arxiv'21	c	softmax	softmax
NNCLR	ICCV'21	d	MLP	K nearest
MSF	ICCV'21	d	MLP	K nearest

Table 4.1: The implementation of predictor layer the existing self-supervised learning methods. We omit the comparison of objective functions in different methods because they are not the focus in this chapter. The type number here denotes one of the four types described in Sec. 6.1 and Fig. 4.1(a-d).

graph convolution layer in Eq. 4.3, Tab. 4.2 shows that these different designs are special cases of Eq. 4.3. The detailed derivation for Tab. 4.2 is presented below.

The Identity Predictor (Fig. 4.1(a)). SimCLR [31] and MOCOv2 [35] do not append an explicit predictor after the projector, which is mathematically equivalent to appending an identity predictor. In this case, the predictor can be formulated as

$$\mathbf{F}_{l+1} = \mathbf{F}_l. \tag{4.5}$$

The formulation above can be obtained by setting $\sigma_l = \mathbf{I}$, $\mathbf{W}_l = \mathbf{0}$, $\mathbf{W}'_l = \mathbf{I}$ in Eq. 4.3 for our unified graph convolution predictor. In this case, the neighborhood aggregation term is ignored, and the self-loop term is exactly the identity mapping.

The MLP Predictors (Fig. 4.1(b)). Popular unsupervised learning methods such as MOCOv3 [37], BYOL [80] and Simsiam [31] use MLP layers as predictors. The typical MLP is a stack of *fc-bn-relu* layers (perception

Method	activation	neighborhood aggregation			self-loop	
	σ	existence	$\mathbf{A}$	$\mathbf{W}$	existence	$\mathbf{W}'$
Identity	$\mathbf{I}$	x	-	$\mathbf{0}$	✓	$\mathbf{I}$
Percepton	ReLU(BN)	x	-	$\mathbf{0}$	✓	train
fc	$\mathbf{I}$	x	-	$\mathbf{0}$	✓	train
softmax	Softmax	x	-	$\mathbf{0}$	✓	$\mathbf{I}$
nearest neigh.	$\mathbf{I}$	✓	$\mathbf{1}$	$\mathbf{I}$	x	$\mathbf{0}$

Table 4.2: Illustrate the simplification to different predictor layers based the formal formulation of Graph Convolution predictor in Eq. 4.3.

layer), where *fc-bn-relu* layer can be formulated as

$$\mathbf{F}_{l+1} = \mathrm{ReLU}(\mathrm{BN}(\mathbf{W}'_l\mathbf{F}_l)). \tag{4.6}$$

The *fc-relu-bn* above can be obtained by setting $\sigma_l = \mathrm{ReLU}(\mathrm{BN})$ in Eq. 4.3. In this case, the neighborhood aggregation term is also ignored and only the self-loop term is presented. Some unsupervised learning methods such as MOCOv3, BYOL and Simsiam uses the fully-connected layer as the last layer in the constructed MLP predictor, which can be obtained by setting the activation function as the identity matrix, *i.e.*, $\sigma_l = \mathbf{I}$.

The Softmax Predictors (Fig. 4.1(c)). DINO presents a softmax predictor to obtain the logits for the following KL-divergence loss. The softmax operation can be presented as

$$\mathbf{F}_{l+1} = \mathrm{Softmax}(\mathbf{W}'_l\mathbf{F}_l). \tag{4.7}$$

The softmax predictor above can be achieved by setting $\sigma_l = \mathrm{Softmax}$ for the graph convolution predictor (Eq. 4.3). In this case, the neighborhood aggregation term is ignored and only the self-loop term is presented.

The K Nearest Neighbors Predictors (Fig. 4.1(d)). Recent states-of-the-art methods treats the sample and its K nearest neighbors in the feature space as the positive samples. Given $\mathbf{F}_l = (\mathbf{f}_l^0, \mathbf{f}_l^1, ..., \mathbf{f}_l^K)$. The K nearest

neighbor predictor can be presented as

$$\begin{aligned}\mathbf{f}_{l+1}^{i} &= \frac{1}{K}(\mathcal{N}_1(\mathbf{f}_l^i) + \mathcal{N}_2(\mathbf{f}_l^i) + ... + \mathcal{N}_K(\mathbf{f}_l^i)) \\ &= \frac{1}{K}(0,1,1,...,1)^\top(\mathbf{f}_l^i, \mathcal{N}_1(\mathbf{f}_l^i), \mathcal{N}_2(\mathbf{f}_l^i), ..., \mathcal{N}_K(\mathbf{f}_l^i)) \\ &= \frac{1}{K}(0,1,1,...,1)^\top(\mathbf{f}_l^0, \mathbf{f}_l^1, ..., \mathbf{f}_l^K) \\ &= \frac{1}{K}(0,1,1,...,1)^\top\mathbf{F}_l.\end{aligned} \tag{4.8}$$

Therefore, the output of the l-th GCN predictor can be formulated as

$$\mathbf{F}_{l+1} = \frac{1}{K}(\mathbf{0},\mathbf{1},\mathbf{1},...,\mathbf{1})^\top(\mathbf{f}_{l+1}^1, \mathbf{f}_{l+1}^2, ..., \mathbf{f}_{l+1}^K) = \frac{1}{K}(\mathbf{0},\mathbf{1},\mathbf{1},...,\mathbf{1})^\top\mathbf{F}_{l+1}. \tag{4.9}$$

Compared with typical graph convolution layer (Eq. 4.3), the K-nearest-neighbor layer can be obtained by setting $\sigma_l = \mathbf{I}$ and the affinity matrix $\mathbf{A} = \frac{1}{K}(\mathbf{0},\mathbf{1},\mathbf{1},...,\mathbf{1})^\top \in \mathbb{R}^{(K+1)\times(K+1)}$. In this case, only neighborhood aggregation term is considered and the self-loop term is ignored.

4.2.3 Graph Augmentations for Unsupervised Visual Learning

To better take advantage of pretext tasks in graph contrastive learning, the proposed GCN predictor layer leverages an augmented graph $\widetilde{\mathcal{G}}(\mathbf{z}_1)$ as the input [193, 316, 21, 239]. Given an input graph $\mathcal{G}(\mathbf{z})$ defined by Eq. 4.2, we implement the typical graph augmentations in self-supervised graph contrastive learning on $\mathcal{G}(\mathbf{z}_1)$, achieving $\widetilde{\mathcal{G}}(\mathbf{z}_1) = (\widetilde{\mathcal{V}}, \widetilde{\mathbf{A}}) = (t(\mathcal{V}), s(\mathbf{A}))$, where t and s are augmentations on the node features $\mathcal{V}$ and affinities $\mathbf{A}$, respectively. After graph augmentation, we will change the input node features and affinity as $\widetilde{\mathcal{V}}$ and $\widetilde{\mathbf{A}}$ in Eq. 4.3 as the input of the GCN predictor.

Node feature masking (NFM). As shown in Fig. 4.3, Node Feature Masking (NFM) randomly masks the features of a portion of nodes within $\mathcal{G}(\mathbf{z})$. In particular, we can completely mask seleceted feature vectors with zeros, or partially mask a number of selected feature channels with

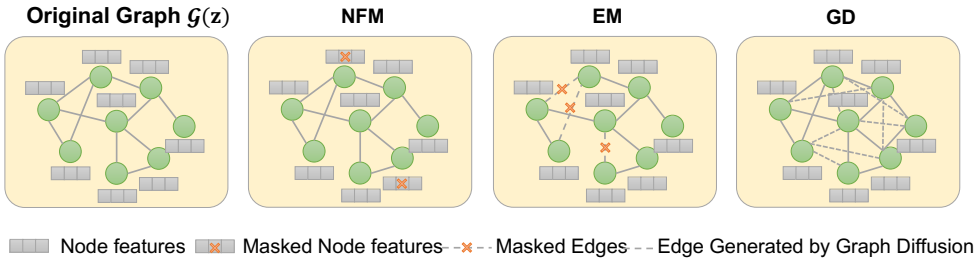


Figure 4.3: Graph augmentations. There are three typical graph augmentations, *i.e.*, node feature masking (NFM), edge modification (EM), and graph diffusion (GD).

zeros. This operation can be formulated as

$$\widetilde{\mathcal{V}} = t(\mathcal{V}) = \mathbf{M}_f \circ \mathcal{V}, \quad \widetilde{\mathbf{A}} = s(\mathbf{A}) = \mathbf{A}, \tag{4.10}$$

where $\mathbf{M}_f$ is the feature masking matrix with the same shape of $\mathcal{V}$, and $\circ$ denotes the Hadamard (element-wise) product. The elements in $\mathbf{M}_f$ are initialized to one and masking entries are 30% randomly assigned to zero.

Edge modification (EM). Edge modification (EM) randomly drops the affinities, which means setting the affinities to zeros. This process is formulated as

$$\widetilde{\mathcal{V}} = t(\mathcal{V}) = \mathcal{V}, \quad \widetilde{\mathbf{A}} = s(\mathbf{A}) = \mathbf{M}_e \circ \mathbf{A}, \tag{4.11}$$

where $\mathbf{M}_e$ is the edge dropping matrix, and $\circ$ denotes the Hadamard product.

Graph diffusion (GD). Graph diffusion is also a type of affinity augmentations, which injects the global affinity information to the given affinity by recomputing the affinity with diffusion operations. The overall diffusion operation can be formulated as

$$\widetilde{\mathcal{V}} = t(\mathcal{V}) = \mathcal{V}, \quad \widetilde{\mathbf{A}} = s(\mathbf{A}) = \sum_{n=0}^{\infty} \Theta_n \mathbf{T}^n, \tag{4.12}$$

where Θ_n and $\mathbf{T}$ are weighing coefficient and transition matrix, respectively. The diffusion above have two common instantiations: heat diffusion [117, 223] and PPR diffusion [52, 67]. The heat diffusion formulates $\Theta_n = \frac{\exp(-\eta)t^n}{n!}$, and $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$, achieving $\widetilde{\mathbf{A}} = \exp(\eta\mathbf{A}\mathbf{D}^{-1} - \eta)\mathbf{A}$, where $\mathbf{A}$ is the affinity matrix, $\mathbf{D}$ is the diagonal degree matrix, $\eta \in (0, 1)$ is the diffusion time. The PPR diffusion formulates $\Theta_n = \gamma(1-\gamma)^n$, and $\mathbf{T} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, achieving $\widetilde{\mathbf{A}} = \gamma(\mathbf{I} - (1-\gamma)\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\mathbf{A}$, where $\gamma \in (0, 1)$ is the teleport probability in random walk.

4.3 Experiment

4.3.1 Implementation Details

Architecture. Our architecture is similar to MOCOv2. Specifically, we use ResNet-50 as our backbone following the common implementations in the self-supervised literature. We spatially average pool the output of ResNet-50 which makes the output of the feature transformation a 2048-dimensional embedding. The projection layer is composed of 3 fully connected layers having output sizes $[2048, 4096, d]$, where d is the feature dimension applied in the loss and $d = 2048$ if not specified. Besides, batch normalization and ReLU activation function is employed in the projection layer following other SSL works [37, 36, 80]. The architecture of the predictor is the GCN layers, which is formulated in Eq. 4.3.

Training. For a fair comparison, we train our method on the ImageNet2012 dataset, which contains 1,281,167 images without using any annotation or class labels. In the training stage, we train for 200 and 800 epochs with a warm-up of 10 epochs and cosine annealing schedule using the LARS optimizer. The base learning rate is set to 0.3. Weight decay of 10^{-6} is applied during training. As is common practice, we do not use weight decay on the bias. The training details above are the same as MOCOv2. We also use the basic data augmentation scheme (i.e., random crop, color jittering) as MOCOv2 and do not include the multi-crop strategy [24] for a fair comparison with the most majority of works.

drop probability	NFM	EM
0	71.9	71.9
0.1	72.1	72.0
0.3	72.1	**72.2**
0.5	71.3	71.7
0.7	70.1	70.5

Table 4.3: Ablation study of node feature masking and edge modification with different drop probabilities.

4.3.2 Ablation study

Exploring the critical factors in GCN predictor. Previous self-supervised learning methods have different predictor designs in activation functions, and the using of neighborhood aggregation. For example, MOCOv2 [35] and SimCLR [31, 33] uses the linear activation, MOCOv3 [37] and BYOL [80] use the ReLU(BN) as the activation function in the online branch, DINO [25] uses Softmax layer as the activation function. Furthermore, the recent state-of-the-art methods, *i.e.*, MSF and NNCLR, retrieve the nearest neighbors in the feature space in the target branch, which can be mathematically viewed as the neighborhood aggregation as analyzed in Tab. 4.1. To strictly ablate the importance of different components in the GCN predictor layer, we keep the batch size, objective function, learning rate schedule, optimizer exactly the same, and then train the ImageNet-1K for 200 epochs. For the MLP predictor, we stack three GCN layers as the common practice with ReLU(BN) being its activation function except the last GCN layer.

Effectiveness of activation function σ. We have three findings from Tab. 4.4. First, comparing Exp. (a) and Exp. (b,e), we can see that with a learnable transformation matrix $\mathbf{W}'$, the non-linear activation is better than the identity activation by about $+0.9\%$, which is consistent with the finding in MOCOv3 [37]. Second, the trainable transformation $\mathbf{W}'$ plays an important role when the activation function is ReLU(BN), but plays an unimportant role when the activation function is Softmax and Identity mapping. We consider the difference may result from the information

Method	Online Branch				Target Branch				Linear
	Act.	Neigh. Agg.		Self-loop	Act.	Neigh. Agg.		Self-loop	
	σ	$\mathbf{A}$	$\mathbf{W}$	$\mathbf{W}'$	σ	$\mathbf{A}$	$\mathbf{W}$	$\mathbf{W}'$	
Baseline									
MOCOv2	$\mathbf{I}$	-	$\mathbf{0}$	$\mathbf{I}$	$\mathbf{I}$	-	$\mathbf{0}$	$\mathbf{I}$	68.6
Effectiveness of Activation Function									
Exp.(a)	$\mathbf{I}$	-	$\mathbf{0}$	train	$\mathbf{I}$	-	$\mathbf{0}$	$\mathbf{I}$	68.5
Exp.(b)	Re(BN)	-	$\mathbf{0}$	train	$\mathbf{I}$	-	$\mathbf{0}$	$\mathbf{I}$	69.3
Exp.(c)	Re(BN)	-	-	$\mathbf{I}$	$\mathbf{I}$	-	-	$\mathbf{I}$	67.5
Exp.(d)	Softmax	-	$\mathbf{0}$	$\mathbf{I}$	Softmax	-	$\mathbf{0}$	$\mathbf{I}$	69.3
Exp.(e)	Softmax	-	$\mathbf{0}$	train	Softmax	-	$\mathbf{0}$	$\mathbf{I}^*$	69.4
Effectiveness of neighborhood aggregation									
Exp.(f)	Re(BN)	-	$\mathbf{0}$	train	$\mathbf{I}$	$\mathbf{1}$	$\mathbf{I}$	$\mathbf{0}$	71.4
Exp.(g)	Re(BN)	-	$\mathbf{0}$	train	$\mathbf{I}$	$\mathbf{1}$	$\mathbf{I}$	$\mathbf{I}$	71.8
Exp.(h)	Re(BN)	$\mathbf{1}$	train	train	$\mathbf{I}$	$\mathbf{1}$	$\mathbf{I}$	ema	71.9

Figure 4.4: Ablation of the importance of parameters in the GCN predictor. $\mathbf{I}^*$ denotes the centering operation proposed in DINO which is used to avoid network collapse. "train" denotes the trainable parameters. "ema" denotes the exponential moving average of the parameters in the online branch.

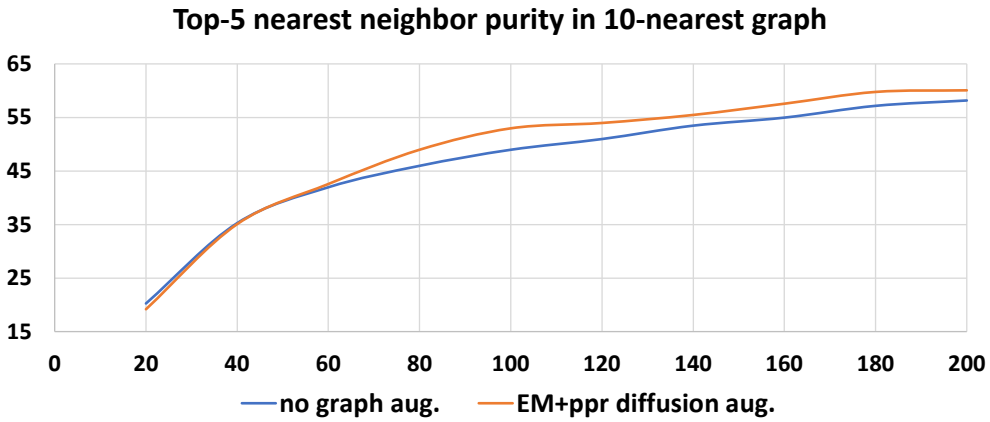


Figure 4.5: Top-5 neighbor purity evolution by graph augmentations.

Method	online branch	target branch	Linear eval
Exp. (i)	No	No	72.2
Exp. (j)	No	Heat Diffusion	72.6
Exp. (k)	No	PPR Diffusion	72.8
Exp. (l)	Heat Diffusion	Heat Diffusion	72.4
Exp. (m)	PPR Diffusion	PPR Diffusion	72.5
Exp. (n)	Heat Diffusion	PPR Diffusion	**72.9**
Exp. (o)	PPR Diffusion	Heat Diffusion	72.8

Table 4.4: Ablation study of different graph diffusion methods on the online and target branches.

loss of ReLU. Third, comparing Exp. (b) and Exp. (e), we find the performance between the GCN layer with different activation functions are quite small if other components are well-designed, which indicates the activation function is a less important factor.

Effectiveness of neighborhood aggregation. Different from self-supervised learning methods that do not use supervision from other samples, neighborhood based methods uses K nearest neighbors as their positive samples. This can be achieved by using the neighborhood aggregation term in Eq. 4.3 in GCN predictor. As shown in Tab. 4.4, we have three findings. First, the linear evaluation performances of using neighboring information on the target branch are significantly higher than those self-supervised learning methods by a considerable 2.1% gain by comparing Exp. (b) and Exp. (f) . Second, comparing Exp. (f) and Exp. (g), we can see adding the self-loop term in the target branch can only boost the performance by 0.4%. Third, when adding the neighborhood aggregation and self-loop term in both online branch and the target branch, the performance can be further improved by 0.1%, which is not significant by comparing Exp. (g) and Exp. (h).

Evaluating graph augmentation in unsupervised image classification. Owing to the GCN predictor, we can naturally bridge the vision SSL with graph SSL, which benefit us to introduce the graph augmentations on the constructed graph $\mathcal{G}(\mathbf{z})$ before the GCN predictor. Refer to Tab. 4.4, we use the Exp (h) as the baseline (using a GCN predictor in both online branch and target branch) in this part and extend it with diverse graph augmentations. Specifically, we explore the effectiveness of three common graph augmentations in the following.

Node feature masking (NFM) and Edge masking (EM). According to the mechanism of NFM and EM described in [157], we randomly remove the node features or edges in both $\mathcal{G}(\mathbf{z})$ and $\mathcal{G}(\mathbf{z}')$ with different drop probabilities. As shown in Tab. 4.3, the performance first increases from 71.9 to 72.1 with NFM and 72.2 with EM, respectively, when we increase the dropping probability from 0 to 0.3. Further increasing the drop probability (e.g., to 0.7) will harm the performance. The results demonstrate

that both node and edge masking graph augmentations are beneficial for vision SSL with a proper drop probability.

Graph diffusion (GD). The graph diffusion propagates the global information in the graph to affinities by diffusion. Based on the results about masking-based augmentations, we further integrate the diffusion-based augmentations with EM (drop probability is 0.3) and explore the influence of heat diffusion and PPR diffusion on both online and target branches.

The experimental results are presented in Tab. 4.4. We can observe that using graph diffusion to incorporate the graph information can significantly improve the the performance baseline by 0.9%. In detail, both heat diffusion and PPR diffusion can benefit vision SSL above the EM augmentation. Besides, using graph diffusion in both branches is more powerful than only using the diffusion in the target branch.

Analysis. In this section, we explain why the graph diffusion operation can improve the unsupervised learning performance empirically. We find the graph diffusion operation can correct the affinity of some visually different but semantically same samples in $\mathcal{G}(\mathbf{z})$. To better illustrate this, we utilize the setting in Exp.(e). Inspired by [209], we compare the **top-5** purity in different epochs between the original 10-nearest graph $\mathcal{G}(\mathbf{z}_2)$ and the 10-nearest augmented graph $\widetilde{\mathcal{G}}(\mathbf{z}_2)$. The purity for a single feature $\mathbf{z}$ is the percentage of $\mathcal{N}_1$ to $\mathcal{N}_K$ in the top-K nearest neighbors which have the same class as $\mathbf{z}$. Final purity is calculated by averaging the purities of all samples. The results are presented in Fig. 4.5. We find the purity of top-5 nearest neighbor in the augmented graph is higher than that in the original graph. By using affinity as the aggregation weight in GCN layer, we can conclude that the features can be aggregated with more accurate neighbors by using graph augmentations and therefore provided better target predictions for the online branch to learn.

Method	Architecture	epochs	Top1	Top5
ODC [287]	ResNet-50	100	57.6	-
InstDisc [259]	ResNet-50	200	58.5	-
LocalAgg [318]	ResNet-50	200	58.8	-
MOCOv2 [35]	ResNet-50	200	68.6	-
MSF [209]	ResNet-50	200	71.4	-
MSF w/s [209]	ResNet-50	200	72.4	-
CPC v2 [93]	ResNet-50	200	63.8	85.3
DINO [93]	VIT-S/16	300	72.5	-
CMC [227]	ResNet-50	240	66.2	87.0
Adco [186]	ResNet-50	200	68.6	-
NNCLR [56]	ResNet-50	200	70.7	-
UniVCL	ResNet-50	200	72.9	-
PIRL [171]	ResNet-50	800	63.6	-
MOCOv2 [35]	ResNet-50	800	71.1	-
SimSiam [36]	ResNet-50	800	71.3	90.7
SimCLR [31]	ResNet-50	800	69.3	89.0
SwAV [24]	ResNet-50	800	71.8	-
InfoMin Aug. [228]	ResNet-50	800	73.0	91.1
BYOL [80]	ResNet-50	1000	74.3	91.6
Adco [186]	ResNet-50	800	72.8	-
Barlow Twins [285]	ResNet-50	1000	73.2	91.0
MoCov3 [37]	ResNet-50	800	73.8	-
NNCLR [56]	ResNet-50	800	75.4	92.4
UniVCL	ResNet-50	800	75.7	93.1

Table 4.5: Comparison with other self-supervised learning methods under the linear evaluation protocol [87] on ImageNet. We omit the result for SwAV with multi-crop for fair comparion with other methods.

Method	ImageNet 1%		ImageNet 10%	
	Top1	Top5	Top1	Top5
Supervised [286]	25.4	48.4	56.4	80.4
Pseudo label [128]	-	-	51.6	82.4
UDA [266]	-	-	68.8†	88.5†
FixMatch [208]	-	-	71.5†	89.1†
MPL [185]	-	73.5†	-	-
InstDisc [259]	-	39.2	-	77.4
PIRL [171]	-	57.2	-	83.8
PCL [132]	-	75.6	-	86.2
SimCLR [31]	48.3	75.5	65.6	87.8
BYOL [80]	53.2	78.4	68.8	89.0
SwAV(multicrop) [24]	53.9	78.5	70.2	89.9
Barlow Twins [285]	55.0	79.2	69.7	89.3
NNCLR	56.4	80.7	69.8	89.3
UniVCL	58.6	81.8	71.8	91.4

Table 4.6: Comparison with the state-of-the-art methods for semi-supervised learning. Pseudo Label, UDA, FixMatch and MPL are semi-supervised learning methods. † denotes using random augment [43]. We follow the exact data split in SwAV [24].

4.3.3 Comparison with State-of-the-art Methods

In this section, we utilize the optimal hyperparameters explored in the previous sections. Specifically, we apply the edge masking, followed by heat diffusion on the online brach and PPR diffusion on the target branch. For GCN predictor, we apply the setting in Exp. (h), where we add a GCN predictor in both online branch and target branch, and the parameters of GCN layers in the target branch are updated from the online branch in a momentum update manner.

Linear evaluations Following the standard linear evaluation protocol [259, 318, 87, 35], we train a linear classifier for 90 epochs on the frozen 2048-dimensional embeddings from the ResNet-50 encoder using LARS [276] with cosine annealed learning rate of 1 with Nesterov momentum of 0.9

and batch size of 4096. Comparison with state-of-the-art methods is presented in Tab. 4.5. Firstly, UniVCL achieves better performance compared to the state-of-the-art methods, using a ResNet-50 backbone without multi-crops augmentations. In 200 epochs training setting, UniVCL improves MOCOv2 by 4.5%, which uses the identity layer in both online branch and target branch. UniVCL still improves the DINO by 0.6% although standard DINO [25] leverages more powerful backbone (VIT-S/16) and more training epochs (300 epochs). The significant improvements by UniVCL verifies the significance of our GCN predictor in the unsupervised vision contrastive learning. Secondly, MSF and NNCLR also leverage the neighboring information, but not in a GCN way. The results of our UniVCL is also higher than MSF and NNCLR [56] by 0.7% and 2.4%, respectively, because of grpah formulation and the introducing of graph augmentations from the graph SSL domain with negligible additional computational cost (less than 2%).

Semi-Supervised Learning Evaluations. We conduct experiments in a semi-supervised setting on ImageNet following the standard evaluation protocol [33, 31], which fine-tunes the whole base network on 1% or 10% labeled ImageNet data without regularization after unsupervised pretraining. The experimental results are presented in Tab. 4.6.

4.3.4 Transfer to 12 cross-domain classification tasks

In this section, we provide the comparison of our UniVCL with other state-of-the-art methods when we transfer our model to 12 cross-domain classification tasks. Specifically, we follow the setup in BYOL [80]. The datasets for classification task include Food101 [18], CIFAR10 [123], CIFAR100 [123], Birdsnap [15], SUN397 [263], Cars [122], Aircraft [167], VOC2007 [60], DTD [41], Pet2 [182], Caltech-101 [63], and Flowers [80]. Specifically, we freeze the backbone of our pretrained models, and train a classifier on the training set of the datasets mentioned above. We test our models on the testing set of the corresponding dataset. We present these results in Table 4.7.

Method	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech	Flowers	Avg
BYOL	75.3	91.3	78.4	57.2	62.2	67.8	60.6	82.5	75.5	90.4	**94.2**	96.1	77.6
SimCLR	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6	72.0
Sup-IN	72.3	93.6	78.3	53.7	61.9	66.7	61	82.8	74.9	91.5	94.5	94.7	77.2
NNCLR	76.7	**93.7**	79.0	61.4	62.5	67.1	64.1	**83.0**	75.5	91.8	91.3	95.1	78.4
UniVCL	**76.9**	93.1	**79.3**	**64.3**	**62.8**	**67.9**	**64.7**	82.5	75.3	**93.0**	93.5	**96.6**	**79.1**

Table 4.7: Transfer learning results on fine-grained classification tasks. Specifically, we fix the pretrained backbone, and then train the classifier with the training set of the 12 cross-domain classification datasets. We report the evaluation results by testing the model on the testing set of the dataset.

As shown in Table 4.7, the transfer results of our method is better than the recent state-of-the-art methods. Specifically, our method is better than other state-of-the-art methods on some fine-grained classfication datasets, *e.g.,* Food101 [18], Birdsnap [15], Cars [122], Aircraft [167], Pets [182] and Flowers [80]. We consider that the reason is that we leverage the neighboring information by the GCN layer, which could exploit the fine-grained information that exist in the ImageNet-1K. For those datasets that have a large domain gap with the ImageNet-1K, *i.e.,* SUN, our method can not help improve the generalization ability. Considering most of the images in ImageNet-1K are object-centric, the scene understanding tasks in SUN can not benefit a lot the self-supervised pretraining from ImageNet-1K.

4.4 Conclusions and Discussions

In this chapter, we unify the recent state-of-the-art methods in our proposed UniVCL. Specifically, we propose the GCN predictor to unify the diverse structural designs of predictor layers in various self-supervised learning methods. Then, fairly and comprehensively experiments are conducted to explore the critical factors in the GCN predictor, revealing the key point of a good predictor is to aggregate neighboring information in the feature space. Owing to the graph perspective, we further verify the effectiveness of graph augmentations in the vision contrastive learning. In the future, we will extend UniVCL from two perspectives, 1) further link the graph self-supervised learning and vision self-supervised learning by exploring other non-contrastive frameworks with graph self-supervised learning, such as reconstruction, attribute prediction, and 2) validating the effectiveness on other vision tasks, *e.g.,* detection, segmentation.

Chapter 5

Improved Transferability of Supervised Pretraining from an MLP Perspective

5.1 Introduction

While Supervised Learning with the cross-entropy loss[1] (SL) were the de facto pretraining paradigm in computer vision for a long period, recent unsupervised learning methods [32, 35, 87, 33, 80, 285, 36, 30, 25, 56, 267] show better transfer learning performance on various visual tasks [80, 108, 299].

This raised the question of why unsupervised pretraining surpasses supervised pretraining even though supervised pretraining uses annotations with rich semantic information.

Several works have attempted to explain the better transferability of unsupervised pretraining than supervised pretraining by the following two reasons: (1) *Learning without semantic information in annotations* [59, 254, 299, 198], which makes the backbone less-overfitted to semantic labels to preserve instance-specific information which may be useful in transfer tasks, and (2) *Special design of the contrastive loss* [299, 108, 112], which helps the learned features to contain more low/mid-level information for effective transfer to downstream tasks. From the supervision

[1] In the chapter, we specifically use the notation "SL" to indicate the conventional supervised learning with the cross-entropy loss.

and loss design perspective, these works provide intuitive explanations for better transferability.

This chapter sheds new light on understanding transferability by considering the multilayer perception (MLP) projector. While previous works [32, 80, 35] verified its effectiveness on the unsupervised image classification task: unsupervised training and evaluating the model on the same ImagNet-1K dataset, they did not explore its effectiveness on transfer tasks thoroughly and rigorously. It is not straightforward to extend the effectiveness of MLP on the unsupervised image classification task to downstream tasks if not supported by rigorous experiments or theoretical analysis, because the performance on the pretraining task is not always predictive of the performance on transfer tasks when there exists a large semantic gap [59, 190, 244]. To our best knowledge, we are the first to identify the MLP projector as the core factor for transferability with deep empirical and theoretical analysis. With this new viewpoint, we find that a simple yet effective method, adding an MLP projector, can promote the transferability of the conventional supervised pretraining methods with the cross-entropy loss (SL) to be comparable or even better than representative unsupervised pretraining methods.

Specifically, we use the *concept generalization task* [198] on ImageNet-1K, where the pretraining and the evaluation datasets have a large semantic distance, as a probe to analyze the transferability of different models. Our experimental results and corresponding analysis indicate that the MLP projector in unsupervised pretraining methods is important for their better transferability. Motivated by this observation, we insert an MLP projector before the classifier in SL, forming SL-MLP. The added MLP can improve the transferability of supervised pretraining, making supervised pretraining comparable or even better than unsupervised pretraining. Experimental results on SL and SL-MLP show three interesting findings: 1) The added MLP preserves the intra-class variation on the pretraining dataset. 2) The added MLP decreases the feature distribution distance between the pretraining and the evaluation dataset; 3) The added MLP decreases the feature redundancy in the pretraining dataset. We also provide a theoretical analysis of how the preserved

intra-class variation and the decreased feature distribution distance improve the performance on the target dataset by adding an MLP projector.

Extensive experimental results confirm that adding an MLP projector into the supervised pretraining method (SL) can consistently improve the transferability of the model on various downstream tasks. Specifically, on the concept generalization task [198], SL-MLP boosts the top-1 accuracy compared to SL (55.9%→63.1%) by **+7.2%**. It also achieves better performance (64.1%) than Byol (62.3%) by **+1.8%** on the 300-epochs pretraining setting. In classification tasks on 12 cross-domain datasets [108], SL-MLP improves SL by **+5.8**% accuracy on average. Moreover, SL-MLP shows better transferability than SL on COCO object detection [149] by **+0.8**% AP. These improvements brought by the MLP projector can largely bridge the transferability gap between supervised and unsupervised pretraining as detailed in Sec. 5.4.2.

The main contributions described in this chapter are three-fold. (1) We reveal that the MLP projector is the main factor for the transferability gap between existing unsupervised and supervised learning methods. (2) We empirically demonstrate that, by adding an MLP projector, supervised pretraining methods can have comparable or even better transferability than representative unsupervised pretraining methods. (3) We theoretically prove that the MLP projector can improve the transferability of pretrained models by preserving intra-class feature variation.

5.2 Transferability Analysis of the Unsupervised and Supervised Pretraining Methods

5.2.1 The Concept Generalization Task

We use the concept generalization task [198] to analyze the transferability gap between the unsupervised and supervised pretraining methods.

Data preparation. Sariyildiz *et al.*[198] evaluated the transferability of methods when the pretraining and evaluation dataset have semantic distance. Their experimental results show that larger semantic distance will lead to more accuracy differences among different pretraining methods. Therefore, we enlarge the semantic gap between the pretraining and the evaluation dataset to help us compare different pretraining methods. Sariyildiz *et al.*[198] use the hierarchy in WordNet [170] and divide ImageNet-21K [45] into six class-exclusive datasets with different semantic distance – one for pretraining, and others for evaluation. Without loss of generality, we construct a smaller pretraining dataset (pre-D) and evaluation dataset (eval-D) based on ImageNet-1K [196] to reduce the experimental burden. Pre-D contains 652 classes mostly of organisms, and eval-D contains the other 348 classes of instrumentality.

Transferability evaluation. Following [198], to assess the transferability, we freeze all parameters in the pretrained backbone [2], and finetune the classifier with the ImageNet-1K training samples in eval-D for reporting top-1 accuracy on ImageNet-1K validation samples in eval-D.

5.2.2 Stage-wise Evaluation on Existing Methods

Motivated by works [299, 108], we make a more thorough stage-wise investigation of the conventional supervised pretraining method (SL) and the existing representative unsupervised pretraining methods (Mocov1, Mocov2, Byol) by evaluating the transferability of intermediate feature maps (Fig. 5.1). After pretraining the model on pre-D, we freeze all model parameters and use the extracted intermediate feature maps of images in eval-D to finetune a stage-wise classifier for a stage-wise linear evaluation.

The evaluation results of these existing methods are depicted in Fig. 5.2 (underlined on the legend). Our stage-wise evaluation shows two new

[2] All experiments in Sec. 5.2 and Sec. 5.3 are conducted with ResNet50.

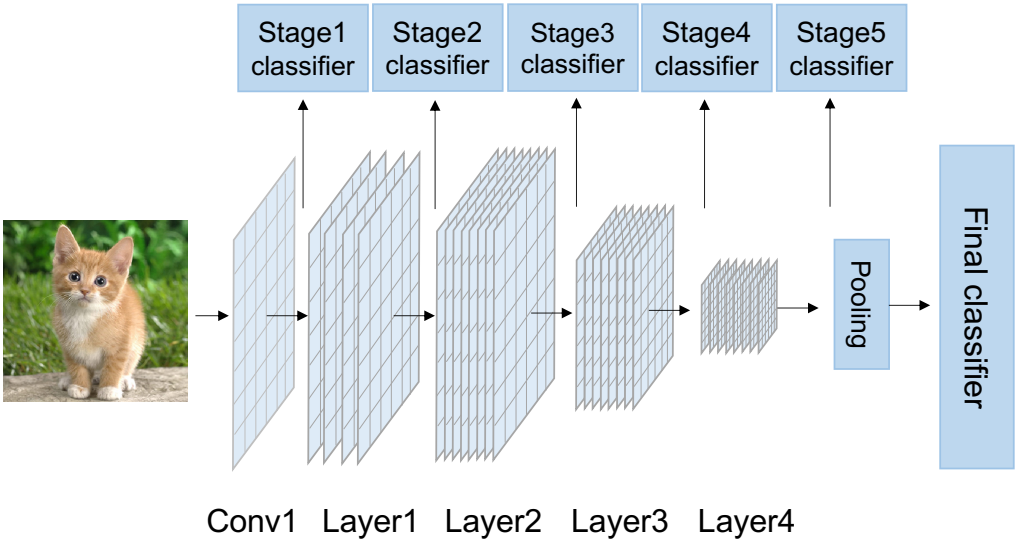


Figure 5.1: Schematic illustration of stage-wise evaluation. We flatten intermediate feature maps from different stages and then use them to train stage-wise classifiers. Top-1 accuracy is reported by evaluating images in eval-D with the stage-wise classifiers.

findings that existing works have not reported. First, on stage-wise evaluation from stage 1 to stage 4, SL is consistently higher than Byol, Mocov1, and Mocov2, which suggests that the semantic information in annotations can benefit the transferability of low/middle-level feature maps. Second, on stage-wise evaluation from stage 4 to stage 5, the performance of Byol and Mocov2 still increase while SL and Mocov1 have a transferability drop. By carefully inspecting these methods, we notice an architectural difference between SL, Mocov1, Mocov2, and Byol after stage 5: An MLP projector is inserted after stage 5 in Byol and Mocov2, which does not exist in SL and Mocov1. Such difference, together with the experimental results in Fig. 5.2, leads to a new hypothesis that the MLP projector might be the core factor of the desirable transferability of unsupervised pretraining.

5.2.3 MLP Improves the Transferability of Unsupervised Pretraining Methods

To confirm our hypothesis of the effectiveness on unsupervised pretraining methods, we ablate the MLP projectors on existing unsupervised

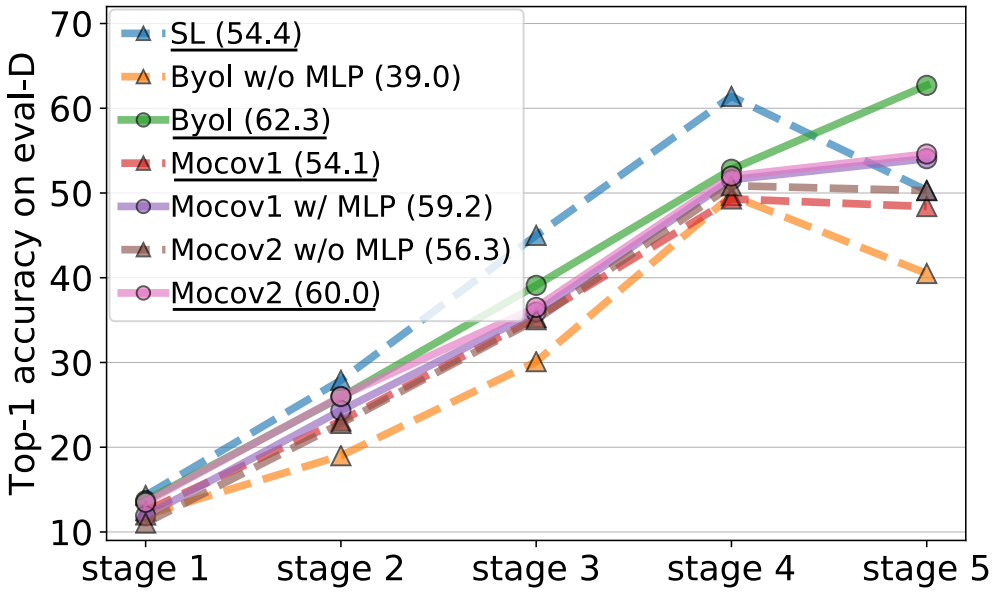


Figure 5.2: Top-1 accuracy of stage-wise evaluation. All methods use ResNet50 as their backbones and are trained by 300 epochs with the setting in the original papers. The results of linear evaluation of layer4-pooled-features (see Fig. 5.1) are reported in the legend.

methods, [3] using stage-wise evaluation. Specifically, we remove the MLP projector in Byol and Mocov2 as Byol w/o MLP and Mocov2 w/o MLP, and add an MLP projector in Mocov1 as Mocov1 w/ MLP. The stage-wise evaluation results of these ablations are summarized in Fig. 5.2. We use solid lines for methods with an MLP projector and dash lines for those that do not have.

These ablation results offer us two observations. First, when evaluating the layer4-pooled-features (depicted in the legend), unsupervised learning methods with an MLP projector achieve better transferability than their variants without the MLP projector, *e.g.*, Byol, Mocov1 w/ MLP, Mocov2 achieve higher accuracy than Byol w/o MLP, Mocov1, and Mocov2 w/o MLP by +23.3%, +5.1% and +3.7%, respectively. Second, on stage-wise evaluation from stage 4 to stage 5, the MLP projector can help unsupervised learning methods without the MLP projector to avoid the transferability drop. These consistent improvements by adding an

[3] We do not directly compare Mocov1 with Mocov2 because Mocov2 has more augmentations and a different learning rate schedule.

MLP projector empirically show that the MLP projector is important for the transferability of unsupervised pretraining. While there might exist some other non-linear structures that can boost the transferability, we only explore from an MLP perspective in this chapter because of its simplicity and demonstrated effectiveness.

5.3 MLP Can Enhance Supervised Pretraining

5.3.1 SL-MLP: Adding an MLP Projector to SL

Motivated by the empirical results in Sec. 5.2, an interesting question is whether the MLP projector can also promote the transferability of supervised pretraining? We attempt to insert an MLP projector before the classifier on SL for better transferability. We denote this supervised pretraining method as SL-MLP (see Fig. 5.3 for their comparison). Specifically, SL-MLP includes a feature extractor $f(\cdot)$, an MLP projector $g(\cdot)$, and a classifier $\mathbf{W}$. Given an input image $\mathbf{x}$, the feature extractor outputs a feature $\mathbf{f} = f(\mathbf{x})$. For example, $f(\mathbf{x})$ transforms an image $\mathbf{x}$ to a 2048 dimensional feature $\mathbf{f}$ when using the ResNet-50 backbone. The MLP projector maps $\mathbf{f}$ into a projection vector $\mathbf{g} = g(\mathbf{f})$. Following Byol, the MLP projector consists of two fully connected layers, a batch normalization layer, and a ReLU layer, which can be mathematically formulated as $g(\mathbf{f}) = fc_2(ReLU(BN(fc_1(\mathbf{f})))) \in \mathbb{R}^{D_g}$, where fc_1 and fc_2 are fully connected layers, the hidden feature dimension in the MLP projector is 4096, and D_g is 256. Given the label denoted by y for image $\mathbf{x}$, the objective function for SL-MLP can be formulated as

$$\mathcal{L}(\mathbf{x}) = \mathrm{CE}(\mathbf{W} \cdot g(f(\mathbf{x})), y), \tag{5.1}$$

where $\mathrm{CE}(\cdot)$ is the cross-entropy loss. Same as SL, only the learned feature extractor $f(\cdot)$ is utilized in downstream transfer tasks after supervised pretraining.

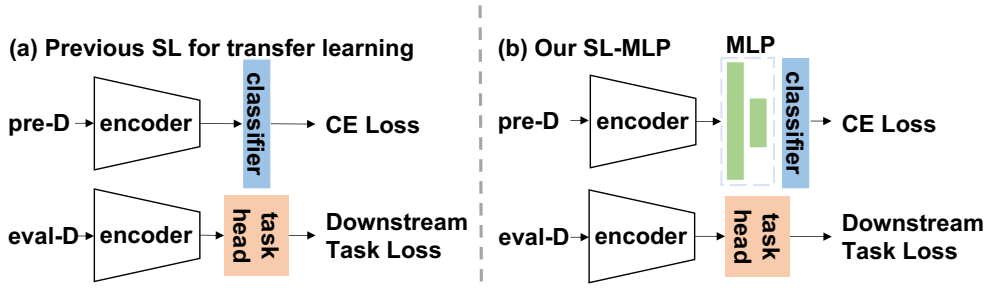


Figure 5.3: The difference between SL and SL-MLP. Our SL-MLP adds an MLP before the classifier compared to SL. Only the encoders in both methods are utilized for downstream tasks.

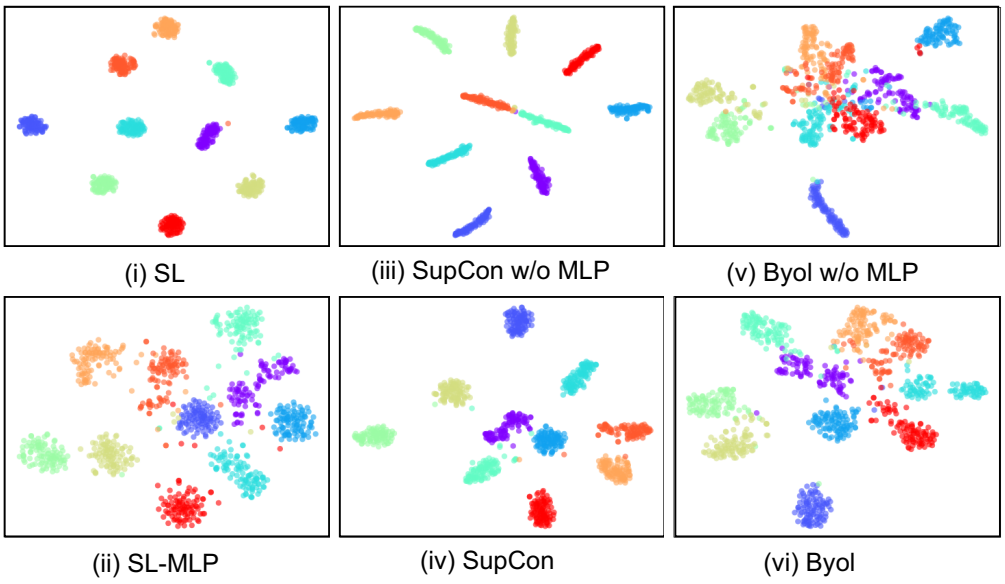


Figure 5.4: Visualization of different methods with 10 randomly selected classes on pre-D. Different colors denote different classes. Features extracted by pretrained models without an MLP projector (top row) have less intra-class variation than those extracted by pretrained models with an MLP projector (bottom row).

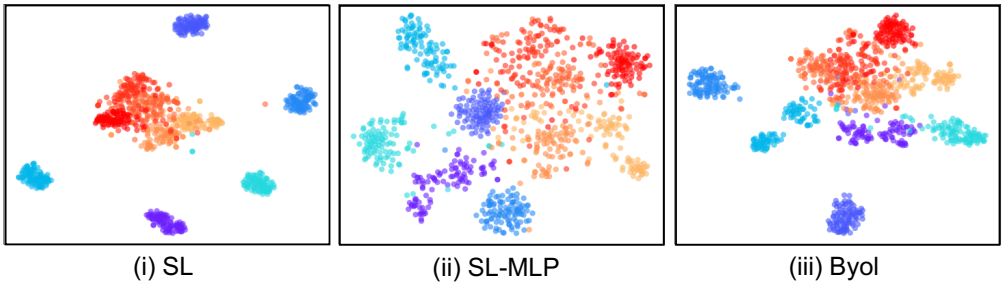
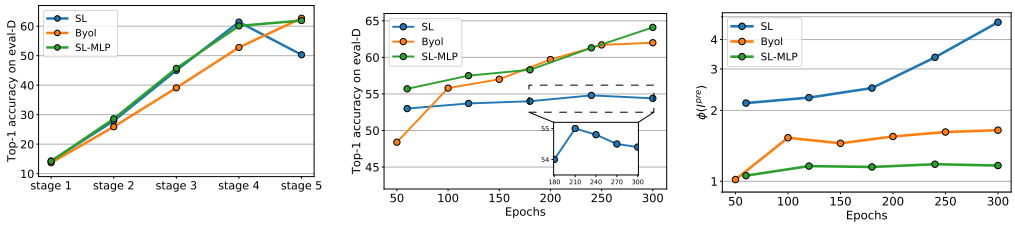
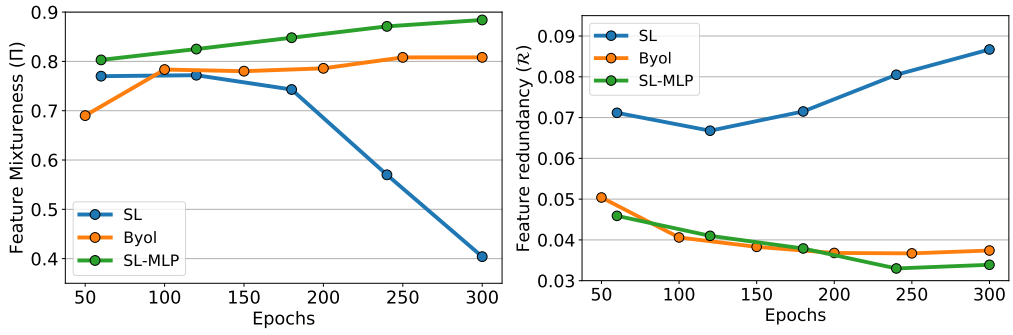


Figure 5.5: Visualization of Feature Mixtureness between pre-D and eval-D. Cold colors denote features from 5 classes randomly selected from pre-D, and warm colors denote features from 5 classes randomly selected from eval-D.



(a) **(b)** **(c)**

Figure 5.6: (a) Stage-wise evaluation on eval-D. **(b)** Linear evaluation accuracy on eval-D. **(c)** Discriminative ratio of features on pre-D. Following [88, 80], we pretrain SL, SL-MLP, and Byol for 300 epochs.



(a) **(b)**

Figure 5.7: (a) Feature Mixtureness between pre-D and eval-D. **(b)** Redundancy $\mathcal{R}$ of pretrained features during different epochs. Following [88, 80], we pretrain SL, SL-MLP, and Byol for 300 epochs.

Method	EP	Top-1(↑)	$\mathcal{R}(\downarrow)$
SL	100	55.9	0.078
SL-MLP	100	**63.1**	**0.035**
SL	300	54.4	0.087
SL-MLP	300	**64.1**	**0.034**
Byol w/o MLP	300	39.0	0.247
Byol	300	**62.3**	**0.037**
Mocov1	300	54.1	0.069
Mocov1 w/ MLP	300	**59.2**	**0.058**

Table 5.1: Redundancy $\mathcal{R}$ of pretrained features. Methods with an MLP obtain lower channel redundancy and transfer better.

5.3.2 Empirical Findings of MLP in SL-MLP

MLP projector avoids transferability drop at stage 5 in supervised pretraining. We conduct stage-wise evaluation as Sec. 5.2.2 again to see whether the transferability drop from stage 4 to stage 5 exists in SL-MLP. In Fig. 5.6(a), the transferability of SL-MLP continuously increases from stage 1 to 5, avoiding a decrease at stage 5 as SL. Besides, we observe that the transferability of SL-MLP is higher than that of Byol from stage 1 to 4, indicating that annotations benefit the transferability of intermediate feature maps.

MLP projector enlarges the intra-class variation of features. According to [299, 108], features with large intra-class variation can preserve more instance discriminative information, which is beneficial for transfer learning. We reveal that adding an MLP projector also can enlarge the intra-class variation. We compare two supervised pretraining methods, *i.e.,* SL, SupCon [112], and one unsupervised pretraining method, *i.e.,* Byol, with their variants with/without MLP. Qualitatively, we visualize their features learned on pre-D by t-SNE in Fig. 5.4. The intra-class variation of features from SL-MLP, SupCon, and Byol are larger than that from SL, SupCon w/o MLP, and Byol w/o MLP, respectively. Quantitatively, following LDA [10], we utilize a discriminative ratio $\phi(I^{pre})$ to measure

intra-class variation on pre-D, where I^{pre} denotes pre-D (mathematically defined in Sec. 5.3.3). Smaller discriminative ratio ϕ usually means larger intra-class variation[4]. Comparing Fig. 5.6(c) with Fig. 5.6(b), we can see Byol and SL-MLP have smaller $\phi(I^{pre})$ but higher accuracy on eval-D than SL, which shows larger intra-class variation can benefit transferability. Furthermore, when inspecting SL only, we can see a process where the accuracy on eval-D first rises and then descends (after 210 epochs) along with $\phi(I^{pre})$ increasing. This phenomenon can be theoretically explained in Sec. 5.3.3. We additionally provide the visualization of intra-class variation on different pretraining epochs in Appendix D.

MLP projector reduces feature distribution distance between pre-D and eval-D. According to [17, 153], decreasing the feature distribution distance between pre-D and eval-D in the representation space can benefit transfer learning. Intuitively, the distribution distance between two sets of features is small when features are well mixed (visualization provided in Appendix D.1). Therefore, we compare the mixtureness of features in pre-D and eval-D to indicate the feature distribution distance between SL and SL-MLP. Graphically, we visualize features from pre-D and eval-D by t-SNE in Fig. 5.5. We observe that features from pre-D and eval-D are more mixed comparing SL and SL-MLP, indicating that MLP projector can mitigate the distribution distance between pre-D and eval-D. Quantitatively, we define **Feature Mixtureness** Π in the feature space as

$$\Pi = 1 - \frac{1}{C}\sum_{i=1}^{C}\left|\frac{top_k^{eval}(i)}{k} - \frac{C^{eval}}{C}\right|, \tag{5.2}$$

where $C = 1000$ is total number of classes in ImageNet-1K, C^{eval} represents the number of classes in eval-D, and $top_k^{eval}(i)$ represents the number of classes in eval-D found by top k neighbors search of any class $i \in C$. Since the percentage of finding a sample from eval-D in k nearest neighbors is C^{eval}/C when pre-D and eval-D are uniformly mixed, Feature Mixtureness measures the similarity of the current and

[4]Strictly speaking, larger intra-class variation is relative to inter-class distance, which is theoretically defined as discriminative ratio. We use "intra-class variation" to be consistent with previous work [108, 299].

the uniformly mixed distribution between pre-D and eval-D in the feature space. We examine Feature Mixtureness of SL, SL-MLP, and Byol during different pretraining epochs in Fig. 5.7(a). Feature Mixtureness of SL gradually decreases, which indicates that SL will enlarge the distribution difference between pre-D and eval-D. In contrast, SL-MLP and Byol show consistently high Feature Mixtureness, indicating that the MLP projector can reduce the distribution distance between pre-D and eval-D. We visualize the evolution of Feature Mixtureness in Appendix D.2.2.

MLP projector reduces feature redundancy. Inspired by [285], high channel redundancy limits the capability of feature expression in deep learning. Mathematically, we compute Pearson correlation coefficient among feature channels to evaluate feature redundancy $\mathcal{R}$, *i.e*,

$$\mathcal{R} = \frac{1}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}|\rho(i,j)|, \quad \rho(i,j) = \frac{\sum_{n=1}^{N}\mathbf{f}_{n,i}\cdot\mathbf{f}_{n,j}}{\sqrt{\sum_{n=1}^{N}||\mathbf{f}_{n,i}||^2}\sqrt{\sum_{n=1}^{N}||\mathbf{f}_{n,j}||^2}} \tag{5.3}$$

where $d = 2048$ is the feature dimension, $\rho(i,j)$ is Pearson correlation coefficient of feature channel i and j. As shown in Fig. 5.7(b), SL-MLP has lower feature redundancy than SL, which indicates that the MLP projector can reduce feature redundancy. In Tab. 5.1, we further confirm that the MLP projector can reduce the feature redundancy and thus increase the accuracy on eval-D by ablating the MLP projector on various pretraining methods.

5.3.3 Theoretical Analysis for Empirical Findings

In this section, we provide a theoretical analysis to reveal that: 1) maximizing the discriminative ratio $\phi(I^{pre})$ of a model on the pretraining dataset above a certain threshold will lead to a transferability descrease (shown by blue/green lines in Fig. 5.8); 2) the threshold is smaller when the semantic gap between the pretraining and evalutaion dataset is larger ($t_l < t_s$ in Fig. 5.8).

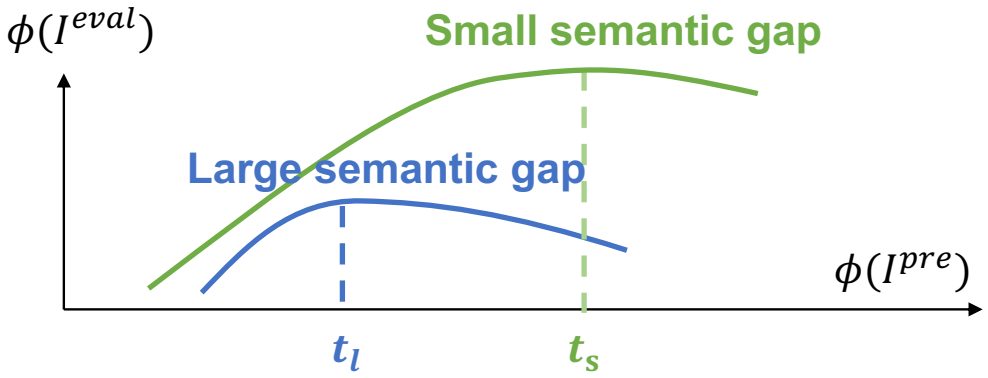


Figure 5.8: Insights for transferability. $\phi(I^{pre})$ and $\phi(I^{eval})$ are the discriminative ratios (Eq. 5.4) on the pretraining and evalution datasets. Higher $\phi(I^{eval})$ indicates better model transferability. Green and Blue line show the performance curve on the evaluation dataset with small and large semantic gap, respectively.

Mathematically, the discriminative ratio $\phi(I)$ on the dataset I can be defined by LDA [10] as

$$\phi(I) = D_{inter}(I)/D_{intra}(I), \tag{5.4}$$

where $D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} ||\mu(I_j) - \mu(I_k)||^2$ is the inter-class distance, $D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} (\frac{1}{|I_j|} \sum_{(x_i,y_i)\in I_j} ||\mathbf{f}_i - \mu(I_j)||^2)$ is the intra-class distance, and C is the number of classes. $\mu(I_j) = \frac{1}{I_j} \sum_{(x_i,y_i)\in I_j} \mathbf{f}_i$ is the center of features in class I_j, and $\mathbf{f}$ is the feature in Sec. 5.3.1. Higher discriminative ratio ϕ indicates higher classification accuracy. Inspired by [150], we analyze the relation between $\phi(I^{pre})$ and $\phi(I^{eval})$ in Theorem 1.

Theorem 1 *Given $\phi_1(I^{pre}) < \phi_2(I^{pre})$, $\phi_1(I^{eval}) > \phi_2(I^{eval})$ when $\phi_1(I^{pre}) > t$, where t is a threshold that is negatively related to the feature distribution distance.*

Insights for the intra-class variation. Theorem 1 reveals that continuously minimizing the intra-class variation (maximizing the discriminative ratio) on the pretraining dataset will decrease the transferability of the model when the discriminative ratio $\phi(I^{pre})$ is larger than t. It explains the observation in Fig. 5.6(b) and Fig. 5.6(c) that training with more than

210 epochs leads to better performance on pre-D, but a worse transferability on eval-D. This insight suggests that we should not make the intra-class variation on the pretraining dataset too small when designing the objective function or network architecture (adding an MLP projector is such a design).

Insights for the relation between the feature distribution distance and threshold t. When the feature distribution distance between the pretraining and evaluation dataset is large, the threshold t is small, in which case it is easier to have the undesirable effect of increasing the discriminative ratio $\phi(I^{pre})$ on pre-D leading to decreasing the discriminative ratio $\phi(I^{eval})$ on eval-D (and thus the accuracy on the evaluation data). This insight suggests that we should maintain more intra-class variation on the pretraining dataset when transferring the model to a target dataset which has a larger semantic distance from the pretraining dataset.

5.3.4 Proof of Theorem 1

Denote the pretrained feature extractor with the parameters $\grave{}$ as $f(\cdot;\grave{})$. The softmax function is built upon the feature representation of the backbone $\mathbf{f}_i = f(\mathbf{x}_i;\theta) \in \mathbb{R}^D$, where $\mathbf{x}_i$ is an image, and D is the dimension of features. We compute the class center $\mu(I_j)$ for class j as the mean of the feature embeddings as

$$\mu(I_j) = \frac{1}{I_j} \sum_{(\mathbf{x}_i, y_i) \in I_j} \mathbf{f}_i, \tag{5.5}$$

where I_j denotes the images in the j-th class. Then we define the inter-class distance $D_{inter}(I)$, and intra-class distance $D_{intra}(I)$ on a datatset with C classes as

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1, k\neq j}^{C} ||\mu(I_j) - \mu(I_k)||^2, \tag{5.6}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} (\frac{1}{|I_j|} \sum_{(\mathbf{x}_i, y_i) \in I_j} ||\mathbf{f}_i - \mu(I_j)||^2). \tag{5.7}$$

Substituting Eq. 5.5 into Eq. 5.6 and Eq. 5.7, we have

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} \left(\frac{1}{2|I_j||I_k|} \sum_{(\mathbf{x}_i,y_i)\in I_j} \sum_{(\mathbf{x}_l,y_l)\in I_k} ||\mathbf{f}_i - \mathbf{f}_l||^2 \right), \tag{5.8}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} \left(\frac{1}{2|I_j|^2} \sum_{(\mathbf{x}_i,y_i)\in I_j} \sum_{(\mathbf{x}_l,y_l)\in I_j} ||\mathbf{f}_i - \mathbf{f}_l||^2 \right). \tag{5.9}$$

Taking expectation to Eq. 5.8 and Eq. 5.9, for any pair of data $(\mathbf{x}_i, y_i), (\mathbf{x}_l, y_l) \in I$, we have

$$\mathbb{E}(||\mathbf{f}_i - \mathbf{f}_l||^2) = \begin{cases} 2D_{intra}(I), y_i = y_l \\ 2D_{inter}(I), y_i \neq y_l \end{cases}. \tag{5.10}$$

For ease of analysis, we denote I^{pre}, I^{eval} as pre-D and eval-D, respectively. For any pair of data $(\mathbf{x}'_i, y'_i), (\mathbf{x}'_l, y'_l) \in I^{eval}$ in eval-D in the same class, *i.e.*, $y'_i = y'_l$, we have

$$\begin{aligned} D_{intra}(I^{eval}) &= \frac{1}{2}\mathbb{E}\left(||\mathbf{f}'_i - \mathbf{f}'_l||^2\right) \\ &= \frac{1}{2}\mathbb{E}\left[P(y_i = y_l)2D_{intra}(I^{pre}) + P(y_i \neq y_l)2D_{inter}(I^{pre})\right] \\ &= PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre}), \end{aligned} \tag{5.11}$$

where y_i is the label of an image $\mathbf{x}_i$ assigned by the classifier trained on pre-D, and $\mathbf{f}' = f(\mathbf{x}', \text{`})$. Here, P represents the possibility that a pair of images in eval-D that belong to the same class is classified into the same classes in pre-D.

We denote $\psi(\phi^{-1}(I^{pre})) = D_{inter}(I^{eval})/D_{inter}(I^{pre})$ as the ratio of the model's inter-class distance on eval-D and the model's inter-class distance on pre-D. When the model is optimized on pre-D, its discriminative ratio on pre-D $\phi(I^{pre})$ becomes larger with the increase of $D_{inter}(I^{pre})$ and the decease of $D_{intra}(I^{pre})$. In most cases, $D_{inter}(I^{eval})/D_{inter}(I^{pre})$ is a monotonic decreasing function of $\phi(I^{pre})$, and is a monotonic increasing function of $\phi^{-1}(I^{pre})$, which has been empirically proven by [150].

Mathematically, it can be formulated as

$$\psi(\phi_2^{-1}(I^{pre})) > \psi(\phi_1^{-1}(I^{pre})), \text{ if } \phi_2^{-1}(I^{pre}) > \phi_1^{-1}(I^{pre}). \tag{5.12}$$

By substituting $D_{intra}(I^{eval}) = PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre})$ (Eq. 5.11) into the discriminative ratio inequality $\phi_2(I^{eval}) < \phi_1(I^{eval})$ given $\phi_2(I^{pre}) > \phi_1(I^{pre})$, we have

$$\phi_2(I^{eval}) < \phi_1(I^{eval}) \tag{5.13}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{D^2_{intra}(I^{eval})} < \frac{D^1_{inter}(I^{eval})}{D^1_{intra}(I^{eval})} \tag{5.14}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{PD^2_{intra}(I^{pre}) + (1-P)D^2_{inter}(I^{pre})} < \frac{D^1_{inter}(I^{eval})}{PD^1_{intra}(I^{pre}) + (1-P)D^1_{inter}(I^{pre})}, \tag{5.15}$$

$$\iff P < \frac{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})}}{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} \cdot \left(1 - \frac{D^2_{intra}(I^{pre})}{D^2_{inter}(I^{pre})}\right) - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})} \cdot \left(1 - \frac{D^1_{intra}(I^{pre})}{D^1_{inter}(I^{pre})}\right)}, \tag{5.16}$$

$$\iff P < \frac{\psi(\phi_1^{-1}(I^{pre})) - \psi(\phi_2^{-1}(I^{pre}))}{\psi(\phi_1^{-1}(I^{pre}))\left(1 - \phi_2^{-1}(I^{pre})\right) - \psi(\phi_2^{-1}(I^{pre}))\left(1 - \phi_1^{-1}(I^{pre})\right)}, \tag{5.17}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre}))}, \tag{5.18}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + r\psi(\phi_1^{-1}(I^{pre}))}, \tag{5.19}$$

$$\iff r\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{5.20}$$

$$\iff \frac{d\phi_1^{-1}(I^{pre})}{d\psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{5.21}$$

$$\iff \frac{d\phi^{-1}(I^{pre})}{P^{-1} - 1 + \phi^{-1}(I^{pre})} < \frac{1}{\psi(\phi^{-1}(I^{pre}))} d\psi(\phi^{-1}(I^{pre})), \tag{5.22}$$

where

$$r = \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))} \tag{5.23}$$

$$\approx \frac{d\phi^{-1}(I^{pre})}{d\psi(\phi^{-1}(I^{pre}))}, \text{ when } \phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre}) \to 0. \tag{5.24}$$

We take integration of Eq. 5.22 as

$$\iff \int_0^{\phi^{-1}(I^{pre})} \frac{d\phi^{-1}(I^{pre})}{P^{-1} - 1 + \phi^{-1}(I^{pre})} < \int_{\psi(0)}^{\psi(\phi^{-1}(I^{pre}))} \frac{1}{\psi(\phi^{-1}(I^{pre}))} d\psi(\phi^{-1}(I^{pre})), \tag{5.25}$$

$$\iff \ln\left[\phi^{-1}(I^{pre}) + P^{-1} - 1\right] < \ln\left[\psi(\phi^{-1}(I^{pre}))\right] + \ln\left(\frac{P^{-1} - 1}{\psi(0)}\right), \tag{5.26}$$

$$\iff \phi^{-1}(I^{pre}) + P^{-1} - 1 < \psi(\phi^{-1}(I^{pre}))\frac{P^{-1} - 1}{\psi(0)}, \tag{5.27}$$

$$\iff \phi^{-1}(I^{pre}) < 1 - P^{-1} + \psi(\phi^{-1}(I^{pre}))\frac{P^{-1} - 1}{\psi(0)}, \tag{5.28}$$

$$\iff \phi^{-1}(I^{pre}) < (\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1)(P^{-1} - 1) \tag{5.29}$$

$$\iff \phi(I^{pre}) > t \tag{5.30}$$

where the threshold t is defined as

$$t = \left[(\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1)(P^{-1} - 1)\right]^{-1}. \tag{5.31}$$

According to Formulation 5.12, $\psi(\phi^{-1}(I^{pre})) > \psi(0)$ because $\phi^{-1}(I^{pre}) > 0$. Therefore, $\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1 > 0$, which means that increasing P will lead to increasing the threshold t.

In the following, we explain how P in Equation 5.11 can be theoretically computed, and how P negatively relates to the feature distribution distance briefly.

Computational Method of P Given a fixed backbone pretrained $f(\cdot;\grave{})$ on pre-D, we denote the classifier trained by pre-D as $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{C^{pre}})$. The possibility of an image $\mathbf{x}$ of the class j in eval-D classified by the classifier $\mathbf{W}$ into the class k in pre-D can be defined as

$$P_{jk} = \frac{1}{|I_j^{eval}|} \sum_{(\mathbf{x}_i, y_i) \in I_j^{eval}} \frac{\exp(\mathbf{w}_k \cdot f(\mathbf{x};\grave{}))}{\sum_{k'=1}^{C^{pre}} \exp(\mathbf{w}_{k'} \cdot f(\mathbf{x};\grave{}))}, \tag{5.32}$$

where $|I_j^{eval}|$ denotes the number of images in the j-th class in eval-D. Then the probability of a pair of samples in the same class j in eval-D classified into the same class in eval-D is

$$P_j = \sum_{k=1}^{C^{pre}} P_{jk}^2. \tag{5.33}$$

The average probability of P_j is

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j. \tag{5.34}$$

P is Negatively Related to the Feature Distribution Distance In this part, we only use two extreme cases to briefly analyze the relation between P and the feature distribution distance. Specifically, we first deduce the upper bound and the lower bound of P. We find that the upper bound is reached when the feature distribution distance between pre-D and eval-D is extremely small, and the lower bound is reached when the feature distribution distance between pre-D and eval-D is extremely large, which indicates P is negatively related to the feature distribution distance.

For the upper bound of P,

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j \tag{5.35}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \sum_{k=1}^{C^{pre}} P_{jk}^2 \tag{5.36}$$

$$\leq \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \left(\sum_{k=1}^{C^{pre}} P_{jk} \right)^2 \tag{5.37}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} 1 \tag{5.38}$$

$$= 1, \tag{5.39}$$

where Inequality 5.37 is derived by Cauchy Schwarz Inequality [256], and if and only if $P_{jk} = 1$ and $P_{jk'} = 0$ for $\forall k' \neq k$, P reaches its upper bound 1.

For the lower bound of P,

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j \tag{5.40}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \sum_{k=1}^{C^{pre}} P_{jk}^2 \tag{5.41}$$

$$\geq \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \left(\sum_{k=1}^{C^{pre}} P_{jk} \right)^2 \tag{5.42}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \tag{5.43}$$

$$= \frac{1}{C^{pre}}, \tag{5.44}$$

where Inequality 5.42 is derived by Fundamental Inequality [13], and if and only if $P_{jk} = \frac{1}{C^{pre}}$ for $\forall k \in [1, C^{pre}]$, P reaches its lower bound $\frac{1}{C^{pre}}$.

Analysis on Small Feature Distribution Distance between pre-D and eval-D. When pre-D and eval-D have small feature distribution distance, a pair of two images $(\mathbf{x}_m, y'_m)$ and $(\mathbf{x}_n, y'_n)$ belong to the same class j in eval-D, *i.e.*,

$y'_m = y'_n$ will be classified to the same class k in pre-D when classified by $\mathbf{W}$ with high confidence. That is, only P_{jk} will have high confidence close to 1 and $P_{jk'}, \forall k' \neq k$ will be close to 0, which is similar to the condition when P reaches its upper bound.

Analysis on Large Feature Distribution Distance between pre-D and eval-D. When pre-D and eval-D have large feature distribution distance, a pair of two images $(\mathbf{x}_m, y'_m)$ and $(\mathbf{x}_n, y'_n)$ belong to the same class in eval-D, *i.e.*, $y'_m = y'_n$ will be randomly classified to the classes in pre-D using $\mathbf{W}$. Mathematically, $P_{jk} \approx \frac{1}{C^{pre}}$, which is similar to the condition when P reaches its lower bound.

Based on the analysis above, we can conclude that P is negatively related to feature distribution distance, and larger P often means less feature distribution distance.

5.4 Experiment

5.4.1 Experimental Setup

Datasets . For backbone analysis, we keep using the concept generalization setting described in Sec. 5.2.1. For generalization to other classification tasks, we follow the setup in [108], which pretrains the models on the whole ImageNet-1K dataset and then evaluates the transferability on 12 classification datasets [247, 173, 180, 41, 92, 225, 126, 39, 243, 176, 177, 42] from different domains. Furthermore, the COCO [149] dataset is used to evaluate the performance of SL-MLP pretrained by ImageNet-1K [196] on object detection task.

Details . For SL and SL-MLP pretraining, the cross-entropy is deployed as the loss function. The MLP projector deployed in SL-MLP is described in Sec. 5.3.1. Following [90], we use the SGD optimizer with a cosine decay learning rate of 0.4 to optimize SL and SL-MLP, and set the batch size to 1024. Byol is used as a representative method for comparisons in backbone analysis and object detection. Following [80], we use LARS

Method	Architecture	Labels	MLP	Epochs	Top-1(↑)
SL	ResNet50	✓		100	55.9
SL-MLP	ResNet50	✓	✓	100	63.1
Byol	ResNet50		✓	300	62.3
SL	ResNet50	✓		300	54.4
SL-MLP	ResNet50	✓	✓	300	**64.1**
SL	ResNet34	✓		100	50.1
SL-MLP	ResNet34	✓	✓	100	55.0
Byol	ResNet34		✓	300	54.8
SL	ResNet34	✓		300	50.2
SL-MLP	ResNet34	✓	✓	300	**55.8**
SL	ResNet101	✓		100	56.0
SL-MLP	ResNet101	✓	✓	100	63.6
SL	ResNet101	✓		300	53.9
SL-MLP	ResNet101	✓	✓	300	**64.7**
SL	MobileNetv2(s=1.4)	✓		200	54.5
SL-MLP	MobileNetv2(s=1.4)	✓	✓	200	**61.5**
SL	EfficientNetb2	✓		100	57.6
SL-MLP	EfficientNetb2	✓	✓	100	**64.2**

Table 5.2: Concept generalization task. We report Top-1 accuracy on eval-D of SL-MLP, Byol, and SL on various backbones. SL-MLP and Byol share the same MLP projector.

optimizer [277] with a cosine decay learning rate schedule and a warm-up of 10 epochs to pretrain the network. The initial learning rate is set to 4.8. We set the batch size to 4096 and the initial exponential moving average parameter τ to 0.99. Except for the backbone analysis, we use ResNet50 as the default backbone.

5.4.2 Experimental Results

Generalize to unseen concepts with diverse backbones. We verify the effectiveness of the added MLP projector on SL using *concept generalization task* with different backbones. Following evaluation method mentioned in Sec. 5.2.1, we train a linear classifier with the frozen backbone for 100 epochs, and report the top-1 accuracy on eval-D in Tab. 5.2.

Method	Sup.	Unsup.	Epoch	object detection		
				AP	AP50	AP75
SL	✓		100	38.9	59.6	42.7
SL-MLP	✓		100	39.7	60.4	43.1
InsDis† [259]		✓	200	37.4	57.6	40.6
PIRL† [171]		✓	200	37.5	57.6	41.0
SwAV† [24]		✓	200	38.5	60.4	41.4
Mocov2† [35]		✓	200	38.9	59.4	42.4
Byol [80]		✓	300	39.4	60.4	43.2
SL-MLP	✓		300	**40.7**	**61.8**	**44.2**

Table 5.3: Object detection results. All methods are pretrained on ImagNet-1K, then finetuned on COCO using Mask-RCNN (R50-FPN) based on Detectron2 [258]. Sup. and Unsup. are short for supervised learning and unsupervised learning, respectively. Results of methods† are from [264].

Firstly, SL-MLP obtains better performance than SL among different backbones. Specifically, with ResNet50, SL-MLP improves SL to 63.1 (+7.2%) when we pretrain the model by only 100 epochs, which bridges the performance gap between SL and Byol. In 300 epochs setting, SL has a transferability drop compared to 100 epochs setting (55.9%→54.4%), but the transferability of SL-MLP continue to increase (63.1%→64.1%). Secondly, SL-MLP (64.1%) performs better than Byol (62.3%) when both are trained by 300 epochs. Experimental results in Tab. 5.2 also confirm that SL-MLP can consistently improve the transferability of SL on various backbones, *e.g.,* ResNet101 [90], MobileNetv2[197], and EfficientNetb2 [219].

Generalize to other classification tasks. To evaluate if the added MLP can help SL to transfer better on cross-domain tasks, following [108], we pretrain the model on ImageNet-1K, and evaluate the transferability on 12 classification datasets from different domains. As illustrated in Tab. 5.4, supervised pretraining methods with the MLP projector, *i.e.,* SL-MLP and SupCon, outperform their no MLP counterparts, *i.e.,* SL and SupCon w/o MLP on linear evaluation, by 5.79%, 13.71% on the averaged Top-1 accuracy, respectively. Consistent results can be observed

Method	ChestX	CropDisease	DeepWeeds	DTD	EuroSAT	Flowers102	Kaokore	Omniglot	Resisc45	Sketch	SVHN	ISIC	Average
linear evaluation													
SL	45.45	96.80	84.02	66.22	95.07	83.69	75.40	64.14	85.36	67.82	67.13	79.58	75.89
SL-MLP	49.89	99.02	87.86	72.61	96.63	93.46	81.12	76.73	91.66	74.51	75.16	81.53	**81.68**
SupCon w/o MLP	41.38	91.52	73.16	62.93	89.84	73.23	66.38	44.54	76.55	55.21	61.45	68.54	67.06
SupCon	47.71	98.79	85.66	74.20	95.83	92.24	79.42	73.42	91.14	76.80	74.26	79.78	<u>80.77</u>
SelfSupCon†	48.08	99.06	87.88	72.71	96.97	89.62	81.67	69.66	90.88	69.12	69.95	81.51	79.70
finetuned with 1000 training samples													
SL	40.86	94.31	86.95	62.12	94.05	88.94	78.22	46.16	80.32	14.17	82.16	78.28	70.54
SL-MLP	42.34	94.48	89.64	63.90	95.30	90.20	77.98	46.66	83.13	17.32	80.19	78.82	**71.66**
SupCon w/o MLP	41.72	93.52	84.95	58.09	95.15	88.23	78.95	45.68	80.63	14.39	82.25	77.96	70.12
SupCon	41.84	93.46	88.70	61.81	94.54	91.28	78.35	46.02	81.62	15.84	81.85	78.51	<u>71.15</u>
SelfSupCon†	43.09	93.95	88.10	62.95	95.47	88.92	79.41	45.33	81.14	10.57	82.37	78.27	70.88
5-ways 5-shots few-shot classification													
SL	25.64	89.07	54.32	78.58	82.96	93.14	46.14	92.82	84.17	87.06	38.03	41.22	67.76
SL-MLP	26.89	93.45	59.08	83.04	87.16	96.88	50.77	95.73	89.00	89.84	41.96	46.76	**71.71**
SupCon w/o MLP	23.62	75.64	49.34	73.04	73.90	82.16	38.10	67.87	75.18	81.01	34.92	35.16	59.16
SupCon	26.18	94.09	59.36	85.02	87.97	96.55	51.02	94.49	89.01	89.75	41.67	43.48	<u>71.55</u>

Table 5.4: Linear evaluation on fixed backbone, full network finetuning, and few-shot learning performance on 12 classification datasets in terms of top-1 accuracy. All models are pretrained for 300 epochs with the same code base except for SelfSupCon† (Mocov2) which pretrained for 400 epochs using the results illustrated in [108]. Average results style: **best**, <u>second best</u>.

Exp	Components				+Params	Top-1
	Input FC	BN	ReLU	Output FC		
(a)					/	55.9
(b)	✓				4.196M	56.6
(c)	✓	✓		✓	8.395M	61.0
(d)	✓		✓	✓	8.391M	60.1
(e)		✓	✓		0.004M	60.5
SL-MLP	✓	✓	✓	✓	8.395M	62.5

Table 5.5: Empirical analysis of architectural design of the MLP projector. We incrementally add different components to the MLP projector. We pretrain models over 100 epochs and set the output dimension to 2048. Top-1 accuracy on eval-D is reported.

on finetuning and few-shot learning settings. Besides, by comparing SupCon, SL-MLP and SupCon w/o MLP, SL, we conclude that the MLP projector instead of the contrastive loss plays the key role in increasing transferability. Our conclusion contrasts with previous works [299, 108] because they ignore the MLP projector before the contrastive loss.

Generalize to object detection. We assess the transferability improvement by the MLP projector on COCO object detection task. We follow the settings in [87] to finetune the whole network with $1\times$ schedule. In Tab. 5.3, we report results using Mask-RCNN (R50-FPN). When both are pretrained over 100 epochs, SL-MLP performs better than SL (without MLP) on object detection by **+0.8 AP**. If MLP is used by both supervised and unsupervised pretraining, SL-MLP pretrained by 100 epochs achieves better performance than unsupervised pretraining (*e.g.*, SwAV and Mocov2) which are pretrained with 200 epochs. When both pretrained over 300 epochs, SL-MLP shows *better* performance than Byol with **+1.3 AP**.

5.4.3 Ablation Study

Effectiveness of different components in MLP. In this part, we investigate the influence of different components in the MLP projector. We set

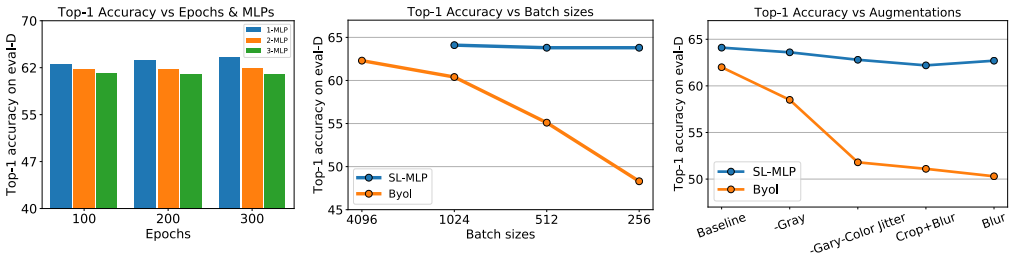


Figure 5.9: (Left to right) **(a)** Top-1 accuracy with different pretraining epochs and number of MLP projectors. **(b)** Top-1 accuracy with different batch sizes shows that SL-MLP has more robust transferability to small batch sizes. **(c)** Top-1 accuracy with different pretraining augmentations shows SL-MLP is robust to augmentations.

the hidden units and output dimension of MLP to be 2048 to retain the dimension of output features the same as SL. Variants are constructed by adding the components incrementally: (a) no MLP projector; (b) only Input FC; (c) Input FC+BN+output FC; (d) Input FC+ReLU+output FC; (e) BN+ReLU. All experiments are pretrained on pre-D over 100 epochs. As shown in Tab. 5.5, SL-MLP achieves the best accuracy among all variants. Besides, we also observe an interesting phenomenon on Tab. 5.5(e) that only inserting a lightweight BN-ReLU also achieves good transfer performance. As this is not our main focus, we will investigate it in future works.

Epochs and layers. Fig. 5.9(a) shows that adding one MLP projector achieves the optimal transferability. In addition, larger pretraining epochs benefit the transferability of SL-MLP when one MLP projector is added, but it has little influence when more MLP projectors are used.

SL-MLP is less sensitive to batch size. Most unsupervised methods depend on big mini-batches to train a representation with strong transferability. To investigate the sensitivity of SL-MLP to batch size, we do experiments with batch size from 256 to 4096 for Byol and to 1024 for SL-MLP over 300 epochs. As shown in Fig. 5.9(b), the transferability of Byol drops when the batch size decreases. In contrast, the transferability of SL-MLP retains when batch size changes.

SL-MLP is less sensitive to augmentation. Unsupervised methods benefit from a broader space of colors and more intensive augmentations during pretraining, which always lead to undesirable degradation when some augmentations are missing. Supervised models trained merely with horizontal flipping may perform well [299]. We set Byol's augmentations as a baseline setting for both SL-MLP and Byol. We then compare the robustness on augmentation between SL-MLP and Byol by removing augmentation step by step. Experiments of SL-MLP and Byol are all constructed on their default condition with only augmentations changed. The results are illustrated on Fig. 5.9(c). We find that SL-MLP inherits the robustness of SL and shows a little accuracy drop with simple augmentations.

5.5 Limitations and Conclusions

This chapter studies the transferability gap between supervised and unsupervised pretraining. Based on empirical results, we identify that the MLP projector is a key factor for the good transferability of unsupervised pretraining methods. Adding an MLP projector into supervised pretraining methods closes the gap between supervised and unsupervised pretraining and even makes supervised pretraining better. Our finding is confirmed with extensive experiments on diverse backbone networks and various downstream tasks, including the concept generalization tasks, cross-domain image classifications, and objection detection. While the MLP is a simple design for better transferability, some straightforward designs might exist on the objective function, which we leave for future work.

Chapter 6

Improved Extreme Multitask Supervised Pretraning for Human-Centric Perception

6.1 Introduction

Human-centric perception has been a long-standing pursuit for computer vision and machine learning communities. It encompasses massive research tasks and applications including person ReID in surveillance [307, 72, 165, 71, 280], human parsing and pose estimation in the metaverse [270, 232, 262, 141, 140, 166], and pedestrian detection in autonomous driving [40, 148, 249]. Although significant progress has been made, most existing human-centric studies and pipelines are task-specific for better performances, leading to huge costs in representation/network design, pretraining, parameter-tuning, and annotations. To promote real-world deployment, we ask: *whether a general human-centric pretraining model can be developed that can benefit diverse human-centric tasks and be efficiently adapted to downstream tasks?*

Intuitively, we argue that pretraining such general human-centric models is possible for two reasons. First, there are obvious correlations among different human-centric tasks. For example, both human parsing and pose estimation predict the fine-grained parts of human bodies [145, 98] with differences in annotation granularities. Thus, the annotations in one human-centric task may benefit other human-centric tasks

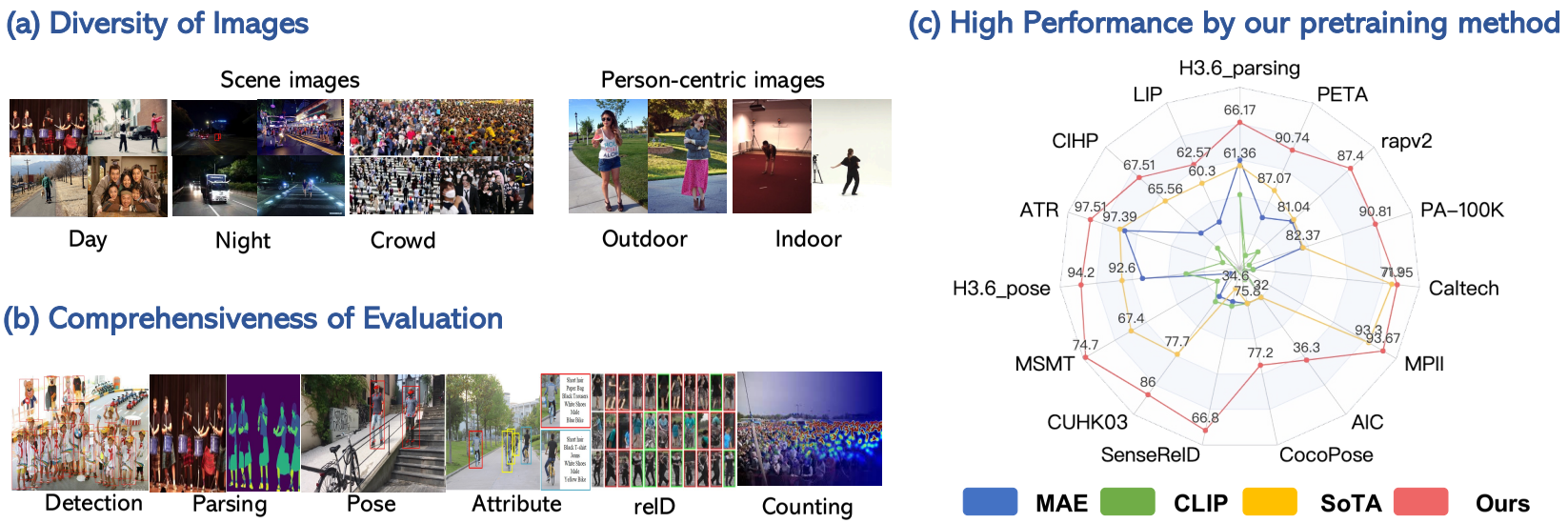


Table 6.1: (a-b) Overview of our proposed HumanBench. HumanBench includes diverse images, including scene images and person-centric images. Our HumanBench also has comprehensive evaluation. Specifically, it evaluates pretraining models on 6 tasks, including pedestrian detection, human parsing, pose estimation, pedestrian attribute recognition, person ReID, and crowd counting. (c) High performances are achieved by our pretraining method on HumanBench. We report 1-heavy occluded MR^{-2} and 1-EPE for Caltech and H3.6pose.

when trained together. Second, recent achievements in foundation models [235, 47, 188, 189, 20, 129] have shown that large-scale deep neural networks (*e.g.*, transformers [53]) have the flexibility to handle diverse input modalities and the capacity to deal with different tasks. For example, Uni-Perceiver [314] and BEITv3 [246] are applicable to multiple vision and language tasks.

Despite the opportunities of processing multiple human-centric tasks with one pretraining model, there are two obstacles for developing general human-centric pretraining models. First, although there are many benchmarks for every single human-centric task, there is still no benchmark to fairly and comprehensively compare various pretraining methods on a common ground for a broad range of human-centric tasks, data distributions, and application scenarios. Second, different from most existing general foundation models trained by unified global vision-language consistencies, pretraining human-centric models are required to learn both global (*e.g.,* person ReID and pedestrian detection) and fine-grained semantic features (*e.g.,* pose estimation and human parsing) of human bodies from diverse annotation granularity simultaneously.

In this chapter, we first build a benchmark, called **HumanBench**, based on existing datasets to enable pretraining and evaluating human-centric representations that can be generalized to various downstream tasks. HumanBench has two appealing properties. **(1) Diversity.** The images in our HumanBench include diverse image properties, ranging from person-centric cropped images to scene images with crowd pedestrians, ranging from indoor scenes to outdoor scenes (Fig. 6.1(a)), and from surveillance to metaverse. **(2) Comprehensiveness.** Humanbench covers comprehensive image-based human-centric tasks in both pretraining datasets and downstream tasks (Fig. 6.1(b)). For pretraining, we include 11 million images from 37 datasets across five representative human-centric tasks, *i.e.,* person ReID, pose estimation, human parsing, pedestrian attribute recognition, and pedestrian detection. For evaluation, HumanBench evaluates the generalization abilities on 12 pretraining datasets, 6 unseen datasets of pretraining tasks, and 2 datasets out of pretraining tasks, ranging from global prediction, *i.e.,* ReID, to local

prediction, *i.e.*, human parsing and pose estimation. Results on our HumanBench (Fig. 6.1(c)) lead to two interesting findings. First, compared with datasets with natural images for general pretrained models, HumanBench is more effective for human-centric perception tasks. Second, as human-centric pretraining requires to learn features of diverse granularity, supervised pretraining methods with proper designs can learn from diverse annotations in HumanBench and perform better than the existing unsupervised pretraining methods, for which details will be shown in Sec. 6.4.3.

Based on HumanBench, we further investigate how to learn a better human-centric supervised pretraining model from diverse datasets with various annotations. However, naive multitask pretraining may easily suffer from the task conflicts [151, 282] or overfitting to pretrained annotations [198, 300], losing the desirable generalization ability of pretraining. Inspired by [250], which suggests adding an MLP projector before the task head can significantly enhance the generalization ability of supervised pretraining, we propose **P**rojector **A**ssis**T**ed **H**ierarchical Pre-training (**PATH**), a projector assisted pretraining method with hierarchical weight sharing to tackle the task conflicts of supervised pretraining from diverse annotations. Specifically, the weights of backbones are shared among all datasets, and the weights of projectors are shared only for datasets of the same tasks, while the weights of the heads are shared only for a single dataset – forming a hierarchical weight-sharing structure. During the pretraining stage, we insert the task-specific projectors before dataset heads but discard them when evaluating models on downstream tasks. With the hierarchical weight-sharing strategy, our pretraining method enforces the backbone to learn the shared knowledge pool, the projector to attend to the task-specific knowledge, and the head to focus on the dataset with specific annotation and data distribution.

In summary, our contributions are two folds: (1) we build HumanBench, a large-scale dataset for human-centric pretraining including diverse images and comprehensive evaluations. (2) To tackle the diversity of input images and annotations of various human-centric datasets, we

propose PATH, a projector-assisted hierarchical weight-sharing method for pretraining the general human-centric representations. We achieve state-of-the-art results by PATH on 15 datasets throughout 6 downstream tasks (Fig. 6.1(c)), on-par results on 2 datasets, and slightly lower results on 2 datasets on HumanBench when using ViT-Base. Experiments with ViT-Large backbone show that our method can further achieve considerable gains over ViT-Base, achieving another 2 new state-of-the-art results and showing the promising scalability of our method. We hope our work can shed light on future research on pretraining human-centric representations, such as unified structures.

6.2 HumanBench

6.2.1 Pretraining Datasets

According to biologists [44], nonverbal communication in daily life includes identity, visual appearance, and posture information. Following this domain knowledge, we select person ReID as the identification task, pedestrian attribute recognition, pedestrian detection, human parsing as the visual appearance task, and pose estimation as the posture task in HumanBench. 37 datasets containing 11,019,187 images [1] are collected for pretraining. Tab. **??** presents the number of datasets and images in each task. For the selected datasets, we leverage their original annotations except for the noisy labeled person ReID dataset, *i.e.*, LUPerson-NL. In LUPerson-NL, we observe that identities with relatively few images are accurate. Therefore, we only select the identities that contain 15 to 200 images in LUPerson-NL, corresponding to 151,595 identities and 5,178,420 images.

To ensure no data leakage and small information redundancy, we further de-duplicate the pretraining dataset from two aspects. First, we

[1] Full list of these 37 datasets are given in Appendix F.

remove all potential duplicates from pretraining datasets that may appear in the evaluation datasets (detailed in Sec. 6.2.2) to enable a meaningful evaluation of generalization. Specifically, we first utilize the Difference Hash [217] to calculate the hash code of images in the evaluation datasets and pretraining datasets. Then, we delete the images in the pretraining datasets that have the same hash code as any image in the evaluation datasets. Second, some images come from some video-based datasets, *e.g.,* AIST++ [133] and UppenAction [292], which contain large information redundancy between consecutive frames. In this case, we select only one image from every 8 consecutive frames to reduce redundancy.

6.2.2 Evaluation Scenarios and Protocols

Evaluation Scenarios. Our benchmark comprehensively quantifies the generalization ability of human-centric representation on 6 human-centric tasks from 19 datasets. Specifically, we establish three evaluation scenarios for HumanBench: (1) *in-dataset evaluation:* we select 12 representative datasets whose training subsets are allocated to the pretraining dataset and evaluation subsets assigned to the evaluation dataset to evaluate the performance of a general pretrained model on diverse seen datasets (meaning similar data distribution for training and evaluation). (2) *out-of-dataset evaluation:* we select 5 datasets that do not appear in pretraining but belong to the seen task for evaluating the ability of the pretrained model on unseen datasets (meaning potentially different data distribution for training and evaluation). (3) *unseen-task evaluation:* we further add 2 datasets for crowd counting to evaluate the generalization ability to unseen tasks. More detailed distributions of these evaluation datasets are presented in Tab. 6.2.

Evaluation Protocols. For each evaluation scenario, we expect a good representation can generalize to specific human-centric tasks without updating the feature extractor or being a good starting point when adapted to any specific human-centric tasks by finetuning. Therefore, we present three evaluation protocols for the experiments in Sec. 6.4.

Task	Datasets	in-dataset evaluations	out-of-dataset evaluations	unseen-task evaluations
ReID	Market1501 [304]	✓		
	MSMT [255]	✓		
	CUHK03 [137]	✓		
	SenseReID [297]		✓	
Pose	COCO [149]	✓		
	Human3.6M [107]	✓		
	AIC [257]	✓		
	MPII [6]		✓	
Parsing	Human3.6M [107]	✓		
	LIP [78]	✓		
	CIHP [77]	✓		
	ATR [146]		✓	
Attribute	PA-100K [158]	✓		
	RAPv2 [130]	✓		
	PETA [46]		✓	
Detecton	CrowdHuman [201]	✓		
	Caltech [50]		✓	
Counting	ShTech PartA [293]			✓
	ShTech PartB [293]			✓

Table 6.2: Summary of datasets for in-dataset evaluations, out-of-dataset evaluations, and unseen-task evaluations.

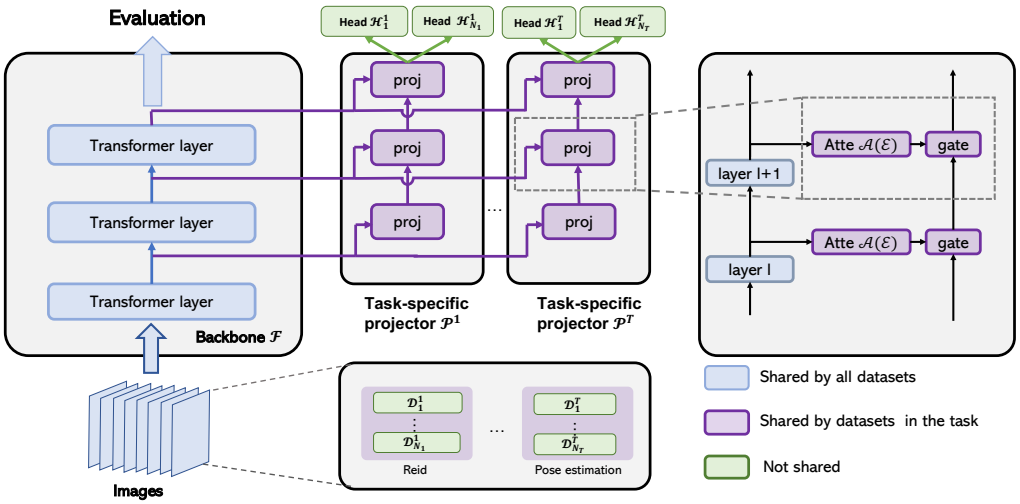


Figure 6.1: Overview of our proposed pretraining method, PATH. Images from various datasets are fed into the same backbone to extract the general features, and then the task-specific projector attends to the task-specific features from the general features. The dataset-specific head is imposed to predict dataset-specific results, which are fed into the loss function for training.

Full Finetuning. Full Finetuning evaluates the generalization ability when pretraining models serve as training starting points. In this case, we load the pretrained backbone and finetune all layers by supervision from downstream tasks.

Head Finetuning. Head Finetuning is very similar to linear probing [87] in self-supervised learning on natural image classification. It evaluates the generalization ability of pretrained models without updating. Therefore, we keep the pretrained backbone frozen and learn simple task heads for downstream datasets.

Partial Finetuning. Partial Fintuning is a setting between head finetuning and full finetuning [86], which finetunes the last several layers while freezing the others. This evaluation protocol can take advantage of both full finetuning and head finetuning, *i.e.*, it can efficiently evaluate the opportunity of pursuing strong but non-linear features.

6.3 Methodology

We now introduce our proposed projector-assisted pretraining method (PATH) with hierarchical weight sharing. Our method is motivated by [250], which reveals that inserting an MLP projector before the objective function can significantly increase the generalization ability of supervised pretraining. To avoid task conflicts among various tasks, we improve this method by inserting task-specific projectors between the backbone and the head of every dataset and designing a new hierarchical weight-sharing strategy. Concretely, the projectors are very lightweight modules composed of attention and gating modules (Sec. 6.3.2). The hierarchical weight-sharing strategy enforces that parameters of backbone, projectors, and heads are shared among all datasets across different tasks, shared among datasets in the same task, and not shared, respectively (Sec. 6.3.1). As such, we expect the backbone to learn general representations of all human-centric tasks, the projector to attend to the task-specific features from the general representations, and the head to supervise the network optimization by the annotations of every dataset. To evaluate the generalization ability of the pretrained backbone, we discard the projectors and heads, using the backbone only.

6.3.1 Hierarchical Weight Sharing

We design a hierarchical weight-sharing strategy to reduce task conflicts among various annotations. Specifically, our model consists of three components: a single backbone shared by all datasets, T task-specific projectors shared by all datasets in the same task, and N dataset-specific heads that are not shared, where $N = N_1 + N_2 + ... + N_T$ is the number of datasets in the pretraining dataset and N_t is the number of datasets in the t-th task.

Backbone. The backbone $\mathcal{F}$ is implemented by a plain vision transformer [53] in the experiments. The parameters of the backbone are shared by all datasets regardless of tasks.

Task-specific projector. Each task-specific projector $\mathcal{P}^t$ consists of sets of attention modules and gating modules, which link with the backbone $\mathcal{F}$, where $t \leq T$ and T is the number of tasks. Since the parameters of the task-specific projector are shared among the datasets with the same task, the attention modules in the task-specific network can be considered as selecting features from the shared backbone network, whilst the shared backbone network learns a compact global feature pool across all datasets.

Dataset-specific head. To tackle the possible data distribution shift in different datasets, we still preserve the dataset-specific head $\mathcal{H}_j^t$, whose parameters are not shared. Here, $\mathcal{H}_j^t$ is the j-th dataset in the t-th task.

Figure 6.1 shows a detailed visualization of our PATH. The detailed pipeline is described as follows.

Step1: Extract the general features of images in the pretraining dataset. Given an image $\mathbf{x}$ sampled from $\mathcal{D}_j^t$ which is the j-th dataset in the t-th task in the pretraining dataset, extract the intermediate and final feature maps $\mathbf{F}$ by the backbone, which will be fed into the projectors.

Step2: Attend the task-specific features by the task-specific projector (Sec. 6.3.2). Given the feature maps $\mathbf{F}$ from the backbone, we attend the task-specific features $\mathbf{p} = \mathcal{P}^t(\mathbf{F})$ by the t-th task-specific projector.

Step3: Calculate the activations by dataset-specific heads, losses by the activations, and optimize the parameters of the backbone, the projector, and the head simultaneously by backward propagation (Sec. 6.3.3).

During the evaluation stage, we discard the projectors and evaluate the generalization ability of the backbone $\mathcal{F}$ using the protocols in Sec. 6.2.2.

6.3.2 Design of Task-specific Projector

The task-specific projector is designed to attend to task-specific features from backbone outputs, by applying an alternating chain of the attention module and gating module to the features in the shared backbone. Given

an image $\mathbf{x}$ sampled from the j-th dataset in the t-th task, *i.e.*, $\mathcal{D}_j^t$ and its intermediate feature maps $\mathbf{f}_l$ in the l-th transformer block, we leverage a squeeze-and-excitation layer [100] to implement the channel attention and a self-attention module [235] to implement spatial attention to generate the attended feature maps $\mathbf{z}_l$. Mathematically, $\mathbf{z}_l = \mathcal{A}^t(\mathcal{E}^t(\mathbf{f}_l))$, where $\mathcal{A}^t$ and $\mathcal{E}^t$ respectively denote standard self-attention blocks [235] and squeeze-and-excitation blocks [100] for the t-th task.

To effectively aggregate features from different backbone layers, we design a gating module to dynamically aggregate features from different layers. Specifically, given the feature map $\mathbf{z}_l$ after the attention module and the gated feature maps $\mathbf{p}_{l-1}$ in the $(l-1)$-th layer, the gating function aggregates features as follows:

$$\mathbf{p}_l = \mu_l \mathbf{z}_l + (1 - \mu_l)\mathbf{p}_{l-1}, \tag{6.1}$$

where $\mathbf{p}_1 = \mathbf{z}_1$, $\mu_l = \sigma(\alpha_l / T)$ is a gate parameterized with a learnable zero-initilalized scalar α_i and temperature T(=0.1), and σ is the sigmoid function. By iteratively computing Eq. 6.1 from $l = 1$ to L, we generate the final feature maps *i.e.*, $\mathbf{p} = \mathbf{p}_L$ for the dataset head.

6.3.3 Dataset-specific Head and Objective Functions

Dataset-specific heads aim at transforming task-specific features into activations for computing losses of every dataset. In general multi-dataset learning with N datasets, the features $\mathbf{P}_i$ after the projector of all images $\mathbf{X}_i$ and labels $\mathbf{Y}_i$, $i = 1, 2, ..., N$ in i-th dataset, the objective function is defined as $\mathcal{L} = \sum_{i=1}^{N} \lambda_i \mathcal{L}_i(\mathbf{Z}_i, \mathbf{Y}_i)$, where $\mathbf{Z}_i$ is the activation generated by the dataset-specific head. This is the linear combination of dataset-specific losses $\mathcal{L}_i$ with task weightings λ_i. In this chapter, we follow some basic head and loss function designs of all pretraining tasks we include. Specifically, we follow the head and loss function designs in VitPose [270] for pose estimation, in TransReID [91] for person ReID, in Segformer [265] for human parsing, in Anchor Detr [249] for pedestrian detection, in Label2Label [136] for pedestrian attribute recognition, and

in DR.VIC [82] for crowd counting. More details of these head and loss designs will be elaborated on below.

Person ReID

Task Head. Following [165], the task head of person ReID is a Synchronized BatchNorm [106]. Mathematically, the activation $\mathbf{Z}_j^t$ is defined as

$$\mathbf{Z}_j^t = \text{BatchNorm}(\mathbf{P}_j^t). \tag{6.2}$$

Objective Function. We use the triplet loss [95] and cross-entropy [296] to supervise the ReID task. Mathematically,

$$\mathcal{L}_{\text{reid}} = \sum_{t=1}^{T}\sum_{j=1}^{N_t}\mathcal{L}_{\text{ce}}(\mathbf{Z}_j^t, \mathbf{Y}_j^t) + \sum_{t=1}^{T}\sum_{j=1}^{N_t}\mathcal{L}_{\text{triplet}}(\mathbf{Z}_j^t), \tag{6.3}$$

where $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss, $\mathbf{Y}_j^t$ is the labels and N_j^t is the number of images in $\mathcal{D}_j^t$. The triplet loss enlarges the distance between negative pairs and minimizes the distance between positive pairs, which can be mathematically defined as

$$\mathcal{L}_{\text{triplet}} = [d_p - d_n + \alpha]_+, \tag{6.4}$$

where d_p and d_n are feature distances of positive and negative pairs. α is the margin of triple loss, and $[\cdot]$ equals $max(\cdot, 0)$.

Pose Estimation

Task Head. Following [270], the task head is lightweight, processes the features after the task-specific features, and localizes the keypoints. We use the structure of classic decoders in [270], which consists of two deconvolution blocks, each of which contains one deconvolution layer followed by layer normalization and ReLU. Following the common setting of previous methods in pose estimation, each block upsamples the feature maps by 2 times. Mathematically, the activation (the localization

heatmaps) can be defined as

$$\mathbf{Z}_j^t = \text{Conv}_{1\times1}(\text{Deconv}(\text{Deconv}(\mathbf{P}_j^t))), \tag{6.5}$$

where $\mathbf{Z}_j^t \in \mathcal{R}^{\frac{H}{4}\times\frac{W}{4}\times N_k}$, H is the height of the image, W is the width of the image, and N_k is the number of keypoints.

Objective Function. We leverage the mean square error (MSE) for pose estimation, *i.e.*,

$$\mathcal{L}_{\text{pose}} = \sum_{t=1}^{T}\sum_{j=1}^{N_t}\text{MSE}(\mathbf{Z}_j^t, \mathbf{Y}_j^t), \tag{6.6}$$

where $\mathbf{Y}_j^t$ is the ground-truth heatmap of keypoints.

Human Parsing

Task Head. We follow the naive head design of [306] for human parsing. Specifically, the naive head first projects the features after the task-specific projectors to the dimension of category number (*e.g.*, 20 in LIP [145]). For this, we adopt a simple 2-layer network with architecture: 1×1 Conv+LayerNorm+ReLU+1×1Conv. After that, we simply bilinearly upsample the output to the full image resolution, followed by a classification layer with pixel-wise cross-entropy loss. Mathematically, the task head can be defined as

$$\mathbf{Z}_j'^t = \text{Conv}_{1\times1}(\text{LayerNorm}(\text{ReLU}(\text{Conv}_{1\times1}(\mathbf{P}_j^t)))), \tag{6.7}$$

$$\mathbf{Z}_j^t = \text{Upsample}(\mathbf{Z}_j'^t), \tag{6.8}$$

where $\mathbf{Z}_j^t$ is upsampled to the size of input images.

Objective Function. Following common implementations in [283], we use the cross-entropy loss to supervise the human parsing. Specifically, the objective function can be defined as

$$\mathcal{L}_{\text{parsing}} = \sum_{t=1}^{T}\sum_{j=1}^{N_t}\text{CE}(\mathbf{Z}_j^t, \mathbf{Y}_j^t), \tag{6.9}$$

where $\mathbf{Y}_j^t \in \mathcal{R}^{H \times W \times N_c}$ is the annotation map whose elements represent the label of the pixel.

Pedestrian Attribute Recognition

Task Head. Following the common implementations in [136], we only use a fully-connected layer followed by a sigmoid function to project the feature to the activation, which can be mathematically defined as

$$\mathbf{Z}_j^t = \text{Sigmoid}(\text{FC}(\mathbf{Y}_j^t)), \tag{6.10}$$

where $\mathbf{Z}_j^t \in \mathcal{R}^{N \times N_c}$ Fc is a fully-connected layer, and N_c is the number of attributes in the dataset.

Objective Function. Our objective function is the binary cross-entropy loss between the activation and the ground-truth label, which can be mathematically defined as

$$\mathcal{L}_{\text{attribute}} = \sum_{t=1}^{T} \sum_{j=1}^{N_t} \text{BCE}(\mathbf{Z}_j^t, \mathbf{Y}_j^t). \tag{6.11}$$

Pedestrian Detection

Task Head. Following Anchor Detr [249], the task head consists of 9 transformer decoder layers, *i.e.*, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_9\}$. The every transformer decoder layer $\mathcal{D}_i$ includes a cross-attention layer, a self-attention layer, and a feedforward network. Therefore, features processed by the decoder $\mathcal{D}_l$ are defined as

$$\mathbf{P}_l = \mathcal{D}_l(\mathbf{Q}_{l-1}^t, \mathbf{Q}_p^t, \mathbf{P}_j^t, \mathbf{P}_p), \tag{6.12}$$

where $\mathbf{P}_p = proj(\mathcal{A}_\mathbf{P})$, $proj$ is a linear projection, and $\mathcal{A}_\mathbf{P}$ is the coordinates of all tokens in the task-specific feature $\mathbf{P}_j^t$. Similarly, $\mathbf{Q}_p^t = proj(\mathcal{A}_\mathbf{Q})$ refers to a linear projection of the coordinates of learnable anchor points initialized with a uniform distribution following [249].

Counting Head
Upsample(scale_factor=2)
Conv{k=(3,3),c=64,s=1}-BN-ReLU
Conv{k=(3,3),c=32,s=1}-BN-ReLU
Upsample(scale_factor=2)
Conv{k=(3,3),c=16,s=1}-BN-ReLU
Conv{k=(3,3),c=1,s=1}-ReLU

Table 6.3: Detailed architecture of counting head.

Objective Function. Given the features $\mathbf{P}_L$ after the decoder, we use the classification loss, GIoU loss and bounding box loss to supervise the pedestrian detection, *i.e.*,

$$\begin{aligned}\mathcal{L}_{peddet} =& \lambda_{cls}\mathcal{L}_{cls}(\mathbf{Z}_{cls}, \mathbf{Y}_{cls}) + \lambda_{iou}\mathcal{L}_{iou}(\mathbf{Z}_{bbox}, \mathbf{Y}_{bbox}) \\ &+ \lambda_{L1}\mathcal{L}_{L1}(\mathbf{Z}_{bbox}, \mathbf{Y}_{bbox}),\end{aligned} \tag{6.13}$$

where $\mathcal{L}_{cls}$ is the classificatin loss, λ_{iou} is the GIoU loss, λ_{L1} is L1 loss of the bounding boxes, and $\mathbf{Y}_{cls}$, $\mathbf{Y}_{bbox}$ are annotations of classes and bounding boxes. Here, $\mathbf{Z}_{cls} = f_{cls}(\mathbf{P}_L)$, $\mathbf{Z}_{bbox} = f_{bbox}(\mathbf{P}_L)$ are linearly projections of $\mathbf{P}_L$, f_{cls} and f_{bbox} are two fully connected layers.

Crowd Counting

Task Head. Table 6.3 details the configurations of counting head for regressing the density map. In this table, "Conv{k(3,3),c64,s1}-BN-R" represents the convolutional operation with kernel size of 3×3, output channels of 64, and stride size of 1. The "BN" and "ReLU" mean that the Batch Normalization and ReLU layer are added to this convolutional layer. Specifically, we denote the task head of counting using layers in Table 6.3 as $\mathcal{H}_{\text{count}}$, *i.e.*,

$$\mathbf{Z}_j^t = \mathcal{H}_{\text{count}}(\mathbf{P}_j^t). \tag{6.14}$$

Objective Function. We leverage the MSE between the activation and the ground-truth heatmap to supervise the learning of crowd counting, *i.e.*,

$$\mathcal{L}_{\text{counting}} = \text{MSE}(\mathbf{Z}_j^t, \mathbf{Y}_j^t), \tag{6.15}$$

where $\mathbf{Y}_j^t$ is the ground-truth heatmap of crowd counting.

6.3.4 Technical Details

Replacing all Batchnorm with Layernorm in pose and parsing decoders. Generally, the original feature normalization method in pose estimation and human parsing tasks is batch normalization with CNN backbone, which renders the model to learn powerful feature distribution from the statistics of batch inputs when trained on a single domain. However, in HumanBench, each task has different datasets, which may have domain gaps, resulting in inaccurate normalization statistics when the dataset-specific head is fed with features from the task-share projector. To reduce the inaccurate statistics, we replace the Normalization method from BatchNorm[106] to LayerNorm[9] and experimentally find that it can improve feature representation.

Sharing Positional Embedding among All Datasets. In HumanBench, the input image size of different tasks varies largely, resulting in different numbers of patch embeddings and positional embeddings after projecting an image to patch embedding. As a result, different tasks cannot share positional embeddings when the model is trained in a distributed manner. To tackle this problem, we parameterize positional embeddings as 224×224 in all tasks and interpolate their size according to each dataset's actual input image size during the pretraining stage.

6.4 Experiment

6.4.1 Experimental Setup

The backbone used for experiments is the plain ViT-base. It has 12 transformer blocks with the dimension of patch embedding 768 and 12 attention heads. In the pre-train stage, each GPU is responsible for one dataset independently for training in a distributed manner. We use Adafactor [202] optimizer with base learning rate of 5e-4 and weight decay of 0.05. We linearly warmup the learning rate from 1e-7 to 5e-4 for the first

1500 iteration steps. Step learning rate decay of 0.5 is used in 50%, 75%, 95% iterations. For the ViT-Base encoder, we set a layer-wise learning rate decay of 0.75 for 12 transformer blocks and the model is trained for 80k iterations.

6.4.2 Experimental Results

As detailed in Sec. 6.2.2, we implement in-dataset evaluation, out-of-dataset evaluation, and unseen-task evaluation on **HumanBench**. Both in-dataset evaluation and out-of-dataset evaluation include 5 human-centric tasks, *i.e.,* person ReID, pose estimation, human parsing, pedestrian attribute recognition, and pedestrian detection. The unseen downstream task which is not in the pretraining tasks, *i.e,* crowd counting, evaluates the generalization ability to unseen tasks. The compared methods are the state-of-the-art methods of each task and two popular pretraining models, *i.e.,* MAE [86] and CLIP [187]. MAE is a newly proposed vision self-supervised pretraining method. Pretrained on ImageNet-1K, MAE achieves excellent results for many visual tasks. CLIP learns generic and transferable representations from a dataset of 400 million (image, text) pairs. We summarize our experimental results with 3 evaluation scenarios and 3 evaluation protocols in Tab. 6.4.

In-dataset Evaluation. In-dataset evaluation quantifies the ability of the pretraining method when it is evaluated on the data with similar data distribution and pretrained tasks. As shown in Tab. 6.4, compared with SoTA methods used in different papers for their specific tasks, our HumanBench with full finetuning achieves better performance on 8 datasets. Specifically, for human parsing, we improve the current state-of-the-art results by **+2.5%** mIOU, **+1.1%** mIOU and **+1.2%** mIOU on Human3.6M, LIP and CIHP, respectively. We also improve the person ReID by **+4.9%** mAP on CUHK03 datasets. We notice our results are lower than PASS [315] on Market1501 and MSMT, probably because PASS uses techniques, *i.e.,* part models [242, 216], that are time-consuming (120 hours using 8 A100

		Human Parsing				Person ReID				Pedestrian Detection	
		Human3.6M	LIP	CIHP	ATR	Market1501	MSMT	CUHK03	SenseReID	CrowdHuman	Caltech (↓)
	SoTA	62.5 [99]	60.3 [154]	65.6 [154]	97.4 [154]	86.8 [91]	61.0 [91]	76.4 [111]	34.6 [297]	92.1 [303]	46.6[84]
	SoTA †	-	-	-	-	93.0 [315]	71.8 [315]	77.7 [134]	-	92.5 [303]	28.8 [84]
ViT-B	MAE	62.0	57.2	62.9	97.4	79.2	51.5	65.8	43.8	89.6	48.1
	CLIP	58.2	53.4	61.7	97.0	78.6	53.6	66.9	42.5	82.1	-
	PATH (w/o FT)	63.9	56.3	63.9	-	88.6	66.3	77.2	-	89.1	-
	PATH (FT)	**65.0**	**61.4**	**66.8**	**97.5**	**89.5**	**69.1**	**82.6**	47.7	90.6	30.1
	PATH (Head FT)	64.1	59.9	63.3	97.1	-	-	-	-	90.0	31.1
	PATH (Partial FT)	63.7	60.0	63.1	97.2	88.7	66.1	79.5	**48.2**	**90.9**	**28.3**
ViT-L	PATH (w/o FT)	65.0	**62.9**	67.1	-	91.6	72.7	83.7	-	89.4	-
	PATH (Partial FT)	**66.2**	62.6	**67.5**	**97.4**	**91.8**	**74.7**	**86.0**	**60.0**	**90.8**	**28.7**

		Pose Estimation				Pedestrian Attribute Recognition			Counting (unseen task)	
		COCO	Human3.6M (↓)	AIC	MPII	PA-100K	RAPv2	PETA	ShTech PartA (↓)	ShTech PartB (↓)
	SoTA	75.8 [270]	7.4 [213]	-	92.3 [273]	83.5 [110]	81.0 [110]	87.1 [110]	94.3 [203]	11.0 [203]
	SoTA †	77.1 [270]	-	32.0 [270]	93.3 [270]	-	-	-	-	-
ViT-B	MAE	75.8	8.2	31.8	90.1	82.3	80.8	84.6	102.1	15.5
	CLIP	74.4	9.9	31.1	88.1	76.1	77.0	81.2	117.9	16.3
	PATH (w/o FT)	75.0	6.9	31.1	-	-	-	-	-	-
	PATH (FT)	**76.3**	6.2	**35.0**	**93.3**	85.0	81.2	88.0	**91.7**	**10.8**
	PATH (Head FT)	75.2	**6.1**	31.6	92.7	77.4	72.4	79.0	-	-
	PATH (Partial FT)	76.0	**6.1**	33.3	93.0	**86.9**	**83.1**	**89.8**	-	14.0
ViT-L	PATH (w/o FT)	74.7	7.1	25.6	-	-	-	-	-	-
	PATH (Partial FT)	**77.1**	**5.8**	**36.3**	**93.7**	**90.8**	**87.4**	**90.7**	-	-

Table 6.4: Experimental results of our PATH and recent state-of-the-art methods (SoTA in the table) on 6 human-centric tasks. The results include 12 *in-dataset evaluations*, 5 *out-of-dataset evaluations* (columns w. gray) and 2 *unseen task evaluations* on the unseen counting task. Following the most commonly-used metrics, for human parsing tasks, we report pACC for ATR, mIoU for others. † indicates that the results are obtained with additional information, multitask learning, or stronger models. We highlight the **best** using ViT-Base and ViT-Large backbone, respectively. We also highlight these best results in red if they outperform SoTAs.

GPUs) but specifically effective for ReID. Besides, we improve pose estimation by **+3.0%** AP, **-1.2%** $MR^{-2}(\downarrow)$ on AIC and Human3.6m, respectively. Furthermore, we improve pedestrian attribute recognition **+1.5%** mA and **+0.2%** mA on PA-100K and RAPv2 datasets, respectively.

To evaluate the generalization of different methods when all backbone parameters or most of the backbone parameters are frozen, we further evaluate our HumanBench with head finetuning and partial finetuning with 100% of the downstream data. We observe that our method with only head finetuning can be on par with and even surpasses the SoTAs in 12 seen datasets, such as **-1.2%** heavy occluded $MR^{-2}(\downarrow)$ **+1.6%** mIOU on Human3.6m pose estimation and human parsing tasks. Our HumanBench with partial finetuning performs better than full finetuning in 2 Pedestrian Attribute Recognition datasets (PA-100K and RapV2) of 12 seen datasets, possibly because these two datasets have fewer data.

We also use ViT-Large to verify the model scalability of our method PATH on HumanBench in Tab. 6.4. Results show that the results with a large backbone under partial finetuning can further achieve considerable gains over the best ViT-Base results, showing the promising scalability of our proposed pretraining method PATH on HumanBench.

Out-of-dataset Evaluation. To quantify the generalization ability of pretrained models on tasks with potentially different data distribution but the same task in the pretraining dataset, we implement out-of-dataset evaluations on 5 datasets, *i.e.,* ATR, SenseReID, Caltech, MPII, PETA, one dataset for each pretraining task. As shown in Tab. 6.4, our pretraining method PATH performs better than previous methods in 4 of 5 unseen datasets and comparable in the remaining one. To be concrete, our method improves by **+0.1%**pACC, **+4.4%**Top1 accuracy, **-0.5%** heavy occluded $MR^{-2}(\downarrow)$ and **+2.7%** mA on ATR (human parsing), SenseReID (person ReID), Caltech (pedestrian detection) and PETA datasets (pedestrian attribute recognition), respectively.

These significant and consistent performance gains across different datasets verify the generalization ability of our pretrained model to tasks

with potentially different data distributions. We also observe the results when we only finetune the last two layers are already on par or even better than the results by full finetuning. Especially, the results of SenseReID, Caltech and PETA by partial finetuing are better than that of full finetuning by **+0.5%**Top1 accuracy, **-1.8%**heavy occluded $MR^{-2}(\downarrow)$ and **+1.8%**mA, showing the good generalization of our pretrained models and its easy deployment in the real world. Similar to the results in the *out-of-dataset evaluation*, partial finetuing performs better than full finetuning when the dataset is small in Caltech (4250 images) and PETA (9500 images). Therefore, partial finetuning can be a choice when the downstream dataset has few samples.

Unseen-task Evaluations. To evaluate the generalization ability to unseen tasks, we construct an *unseen task evaluation*, in which the task is not involved in the pretraining tasks, *i.e,* crowd counting. As presented in Tab. 6.4, our pretrained model achieves significant performance gains than the MAE pretrained model by **-10.4%** MSE$(\downarrow)$ and **-4.7%** MSE$(\downarrow)$. Our HumanBench improves previous SoTAs that are specially designed for crowd counting by **-2.6%** MSE$(\downarrow)$ and **-0.2%** MSE$(\downarrow)$ on ShTech PartA and PartB datasets, respectively. These consistent improvements validate the generalization ability of our learned representations.

Comparsion with MAE and CLIP Models. We also compare our pretrained method with other popular pretrained models, *i.e.,* MAE and CLIP, on our proposed HumanBench. In Tab. 6.4, we find our pretraining method performs considerably better than CLIP and MAE under the full finetuning evaluation protocol on all tasks. Interestingly, the performance of CLIP[2] is lower than MAE, which shows that more data on natural images and languages may not naturally benefit a variety of human-centric tasks, which empirically validates the importance of our HumanBench for further research on human-centric pretraining.

[2]We carefully tune the learning rate, drop path rate, and weight decay of CLIP pretrained ViT-B and report the best results we have ever achieved.

		(a)	(b)	(c)	(d)
	Shared Pos. Embedding				✓
	Projector Share Type	A	S	T	T
Detection	Caltech †	60.6	57.5	60.2	60.9
Attribute	PA100K	84.4	84.6	84.6	84.4
	PETA	87.6	87.2	87.9	87.5
Pose	MPII	92.0	92.4	92.5	92.4
Parsing	LIP	59.7	59.7	60.5	61.0
ReID	Market1501	86.2	86.5	87.1	87.6
	MSMT	65.8	66.0	66.1	66.8
	On average	**76.6**	**76.3**	**76.9**	**77.2**

Table 6.5: Ablation results. "A", "S", and "T" respectively denote all shared, specific, and task-shared projectors. † indicates the results are reported as 1-heavy occluded MR^{-2} for averaging.

6.4.3 Ablation Study

Due to the significant computation cost with the large-scale full datasets, as summarized in Table 1 in Appendix F, we sample a subset containing a similar number of images as ImageNet-1K ($\sim$ 1.28 M) from the full training set. We pretrain our models on this subset to verify the effectiveness of our designs by default in this section, and implement 4 in-dataset evaluations (PA-100K, LIP, Market1501, MSMT) and 3 out-of-dataset evaluations (Caltech, PETA, MPII) full dataset finetuning.

Effectiveness of hierarchical weight sharing. To verify the effectiveness of our hierarchical weight sharing, we adapt the three projector share strategies: (1) all shared projector (**A**): sharing the projector parameters across all the tasks and datasets (Table 6.5 (a)); (2) task-shared projector (**T**): sharing the projector parameters across all datasets in a single task, while maintaining an independent projector for each task (Table 6.5 (c)); and (3) specific projector (**S**): maintaining an independent projector for each dataset (Table 6.5 (b)). The results show that the task-shared projector is better than the other two. We speculate that the

Pretraining data		ImageNet-1K	Our subset		
Method		MAE	MAE	MOCOv3	Ours
Detection	Caltech †	51.9	58.2	57.5	60.9
Attribute	PA100K	82.3	83.5	82.9	84.4
	PETA	84.6	85.3	84.3	87.5
Pose	MPII	90.1	91.3	90.4	92.4
Parsing	LIP	57.2	60.1	58.6	61.0
ReID	Market1501	79.2	84.6	86.8	87.6
	MSMT	51.5	64.5	67.2	66.8
On average		**71.0**	**75.4**	**75.4**	**77.2**

Table 6.6: Comparison with self-supervised pretraining methods on ImageNet and the subset of our HumanBench. † indicates the results are repoted as 1-heavy occluded MR^{-2} for averaging.

projector is the core component to map the general human-centric features to task-specific features. Therefore, all datasets in the same task are supposed to share the same mapping functionality while different tasks should operate differently due to the existing task gaps.

Effectiveness of shared positional embedding. Experiments (c) and (d) in Tab. 6.5 ablate whether positional embeddings are shared or not across the different tasks. The results show that shared positional embedding helps to learn general human-centric representations and leads to **+0.3%** improvement on average when the five tasks are considered. Since the backbone is shared, we conjecture that independent positional embedding would cause inconsistency between different tasks, create barriers in learning shared backbone across tasks, and result in difficulties in learning the model.

Comparison with self-supervised pretraining methods. As shown in Table 6.6, we first ablate the effectiveness of our dataset on downstream tasks. With almost the same number of images, the MAE pretrained on our subset for 800 epochs surpasses ImageNet pretrained MAE by +4.4%, which shows that by combining the diverse human-centric data

across various human-centric tasks, our dataset is more suitable to learn human-centric features. Second, pretrained on our subset, our supervised pretraining method, i.e. PATH, performs better than both MAE (800 epochs) and MOCOv3 (800 epochs) by +1.8%. Different from MAE and MOCOv3 which ignore the general properties of the human body and the potential association between the data in different tasks, our PATH is designed to capture the potential complementary knowledge between different tasks, leading to learning more general human-centric representations to improve the performance on various human-centric tasks.

6.5 Conclusion

In this chapter, we investigate the opportunities and challenges in pretraining on various human-centric tasks and propose a new HumanBench with the existing publicly available datasets. Based on HumanBench, we design a projector-assisted pretraining with hierarchical weight sharing (PATH) to learn human-centric information from annotations with different granularities. We hope our HumanBench can facilitate future works such as unified network structure design and multi-task/ supervised/ self-supervised learning methods on a broad variety of human-centric tasks.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The field of large-scale visual pretraining witnessed significant progress in recent years. However, there are still a number of obstacles that need to be overcome to make the development of visual large-scale pretraining more effective. Given the growing interest in developing large foundation models, it is crucial to address these challenges. In this thesis, we make a focused effort to tackle three of the most pressing challenges in this field, and proposed novel solutions. Our major contributions can be summarized as follows:

- **Relative Contrastive Loss (RCL):** Chapter 3 introduces a relative contrastive loss for unsupervised learning that treats query-key pairs as relatively positive based on semantic criteria derived from online hierarchical clustering. The representations learned with this loss capture diverse semantic criteria, improving sample relationships. Extensive results across self-supervised, semi-supervised, and transfer learning settings demonstrate the effectiveness of the proposed loss, although optimal criteria design is yet to be explored.
- **Unified Visual Contrastive Learning (VCL):** Chapter 4 presents UniVCL, a unifying framework that combines state-of-the-art methods. The GCN predictor is introduced to unify predictor layer designs in self-supervised learning methods, emphasizing the importance of aggregating neighboring information in the feature space.

The effectiveness of graph augmentations in vision contrastive learning is confirmed. Future work includes exploring non-contrastive frameworks with graph self-supervised learning and validating effectiveness in other vision tasks.

- **Improved Supervised Pretraining by adding an MLP projector (SL-MLP):** Chapter 5 studies the transferability gap between supervised and unsupervised pretraining, finding that the MLP projector is crucial for good transferability. Adding an MLP projector to supervised pretraining narrows the gap and improves performance. Extensive experiments on diverse networks and tasks support this finding, while future work could explore other straightforward designs in the objective function.
- **Supervised Pretraining with Hierarchical Weight Sharing (PATH):** Chapter 6 investigates pretraining on human-centric tasks and introduces HumanBench, a benchmark dataset. The proposed PATH method utilizes hierarchical weight sharing to learn human-centric information from annotations of different granularities. HumanBench aims to facilitate research in network structure design and multi-task learning for various human-centric tasks.

By conducting thorough experiments and analysis, we have provided compelling evidence of the efficacy of our proposed methods in tackling the challenges associated with neural network design and training. These contributions establish a solid groundwork for future research in this area and contribute to the advancement of state-of-the-art techniques in neural network design and training.

7.2 Limitations of current visual pretraining

While this thesis aims at designing visual pretraining, they still have the following challenges when deployed in the real world:

- **Dataset Bias:** Visual pretraining methods heavily rely on large-scale datasets, which often suffer from biases in terms of demographics, representation, and environmental factors. This can lead

to biased models and poor generalization in real-world scenarios.

- **Lack of Domain Adaptation:** Pretrained models may not generalize well to new domains or environments that differ significantly from the training data. The models may fail to capture important nuances and variations present in real-world data.
- **Limited Robustness:** Pretrained models are susceptible to perturbations and adversarial attacks. They may fail to perform reliably when faced with real-world challenges such as occlusions, varying lighting conditions, or image distortions.
- **Lack of Fine-grained Control:** Visual pretraining methods often focus on general object recognition and may not provide fine-grained control over specific tasks or attributes required in real-world applications. Fine-tuning or additional training may be necessary to achieve desired performance.
- **Computational Resources:** Training large-scale visual models requires substantial computational resources, limiting their accessibility and scalability in real-world deployment scenarios, especially in resource-constrained environments.
- Lack of Interpretability: Pretrained models often lack interpretability, making it challenging to understand their decision-making process or diagnose potential errors or biases, which is crucial for real-world applications where transparency and trust are important.

Addressing these limitations is crucial to ensure the reliable and effective deployment of visual pretraining methods in real-world scenarios.

7.3 Future Work

The research presented in this thesis opens up several avenues for future work. In the following, we outline some of the most promising directions for future research.

- **Relative Contrastive Loss by Improved Clustering.** The performance of unsupervised pretraining largely relies on the accuracy of clustering. Further investigation into the better clustering algorithms could continuously achieve improved performance on unsupervised pertraining.
- **Investigation of MLP Projector in Different Domains.** Extending the analysis of the Multilayer Perceptron (MLP) projector's impact on transferability, future work can focus on evaluating its effectiveness in various domains beyond image classification. Investigating its performance in tasks such as object detection, semantic segmentation, or video understanding can provide a comprehensive understanding of the MLP projector's role in improving transferability across different visual tasks.
- **Multitask Pretraining Methods for a Wider Range of Human-Centric Tasks.** While the current version of PATH encompasses five human-centric tasks, there are several additional tasks that remain unaddressed, including action recognition, 3D human reconstruction, and crowd counting, among others. The development of the next iteration of UniHCP to incorporate these tasks would represent a noteworthy advancement towards achieving comprehensive human-centric perception.

To conclude, the outcomes presented in this thesis provide valuable insights into neural network design and training, opening up exciting possibilities for further advancements. As we strive for practical and efficient approaches in large-scale visual pretraining, we foresee the expansion of this field enabling progress in artificial intelligence and contributing to a brighter future for everyone.

Appendix A

Mathematical Analysis of Relative Contrastive Loss

A.1 Detailed Mathmatical Analysis of Relative Contrastive Loss

We start by denoting two different images $\mathbf{x}$ and $\mathbf{x}'$. Given a criteria $\mathcal{M}_i$, we define their label as $\mathcal{Y}_i(\mathbf{x})$ and $\mathcal{Y}_i(\mathbf{x}')$, respectively. Inspired by BYOL [80] and Simsiam [36], the predictor layer aims to predict the expectation of projections $\mathbf{z}$ under transformation $\mathcal{T}_i$, *i.e.,* $\mathbb{E}_{\mathcal{T}_i}(\mathbf{z})$, where $\mathcal{T}_i$ is semantic-invariant on $\mathcal{M}_i$, *i.e.,* $\mathcal{Y}_i(\mathcal{T}_i(\mathbf{x})) = \mathcal{Y}_i(\mathbf{x})$.

To analyze the formulation of our relative contrastive loss given two images $\mathbf{x}$ and $\mathbf{x}'$, we start with its individual component in Eq. (2) of the main paper, *i.e.,*

$$\mathcal{L}_{RCL}\left(\mathbf{x}, \mathbf{x}', \theta; \{\mathcal{M}_i\}_{i=1}^{H}\right) = \sum_{i=1}^{H} \alpha_i \mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i), \tag{A.1}$$

where α_i is the trade-off parameter among different criteria. The loss $\mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)$ for criterion $\mathcal{M}_i$ can be defined as

$$\mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i) = -\log\left[\frac{\mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] \times \exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')\right]}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}\right]. \tag{A.2}$$

When $\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')$,

$$\mathcal{L}^+(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i) = -\log \left[\frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}'/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}'/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{s}_k/\tau\right)} \right], \tag{A.3}$$

which pulls the query predictions $\mathbf{q}_{\mathcal{M}_i}$ and projection $\mathbf{z}'$ together.

When $\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')$,

$$\mathcal{L}^-(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i) = -\log \left[\frac{1}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{z}'/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^\top \mathbf{s}_k/\tau\right)} \right], \tag{A.4}$$

which pushs the query predictions $\mathbf{q}_{\mathcal{M}_i}$ and projection $\mathbf{z}'$ apart.

Given a query-key pair $(\mathbf{z}, \mathbf{z}')$ and a set of semantic criteria $\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$, if $\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')$ for $i < h$ and $\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')$ for $i \geq h$, the relative contrastive loss becomes

$$\mathcal{L}_{RCL}\left(\mathbf{x}, \mathbf{x}', \theta; \{\mathcal{M}_i\}_{i=1}^{H}\right) = \sum_{i=1}^{h-1} \alpha_i \mathcal{L}^-(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i) + \sum_{i=h}^{H} \alpha_i \mathcal{L}^+(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i), \tag{A.5}$$

where the positive-negative relation of $(\mathbf{z}, \mathbf{z}')$ is relative and depends on the particular semantic criterion $\mathcal{M}_i$.

A.2 Derivative of Gradients of Relative Contrastive Loss

We calculate the gradient of relative contrastive loss $\mathcal{L}^+(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)$ when $\mathcal{Y}_i(\mathbf{x}) = \mathcal{Y}_i(\mathbf{x}')$ and $\mathcal{L}^-(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)$ when $\mathcal{Y}_i(\mathbf{x}) \neq \mathcal{Y}_i(\mathbf{x}')$, respectively.

Specifically,

$$\begin{aligned}
\frac{\partial \mathcal{L}^{+}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= -\frac{\partial}{\partial \mathbf{z}}\left[\frac{\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'}{\tau} - \log\left[\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)\right]\right] \\
&= -\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{z}'}{\tau} + \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right)\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{z}'}{\tau} + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{s}_k}{\tau}}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}, \\
\frac{\partial \mathcal{L}^{-}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= -\frac{\partial}{\partial \mathbf{z}}\left[-\log\left[\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)\right]\right] \\
&= \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right)\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{z}'}{\tau} + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{s}_k}{\tau}}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}.
\end{aligned} \tag{A.6}$$

Denote

$$\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}, \tag{A.7}$$

$$\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'\right)/\tau) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}, \tag{A.8}$$

where $\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right)$ and $\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right)$ are always non-negative and $\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) + \sum_{k=1}^{K}\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) = 1$. Therefore, $\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right)$ can viewed as a valid probability of assigning the query prediction $\mathbf{q}_{\mathcal{M}_i}$ to the label of projection $\mathbf{z}'$ and the label of negative samples $\mathbf{s}_k$, respectively.

After substituting Eq. A.7 and Eq. A.8 into Eq. A.6, we get

$$\begin{aligned}
\frac{\partial \mathcal{L}^{+}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= \left[\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) - 1\right]\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{z}'}{\tau} + \sum_{k=1}^{K}\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right)\frac{\mathbf{s}_k}{\tau}, \\
\frac{\partial \mathcal{L}^{-}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= \left[\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right)\right]\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\frac{\mathbf{z}'}{\tau} + \sum_{k=1}^{K}\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right)\frac{\mathbf{s}_k}{\tau}.
\end{aligned} \tag{A.9}$$

Then ,we get the gradient of $\mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)$ in Eq. A.2 as Eq. 4 in the main text, *i.e.*,

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{x}', \theta; \mathcal{M}_i)}{\partial \mathbf{q}_{\mathcal{M}_i}} \\ &= \left[\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right] \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\mathbf{z}'}{\tau} + \sum_{k=1}^{K} \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) \frac{\mathbf{s}_k}{\tau}.\end{aligned} \tag{A.10}$$

Finally, the gradient of relative contrastive loss $\mathcal{L}_{RCL}$ is (discard negative samples $\{\mathbf{s}_k\}_{k=1}^{K}$ in the support set $\mathcal{S}$)

$$\frac{\partial \mathcal{L}_{RCL}}{\partial \mathbf{z}} = \sum_{i=1}^{H} \alpha_i \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right) \frac{\mathbf{z}'}{\tau}. \tag{A.11}$$

Appendix B

Visualization of Relative Contrastive Loss

The relative contrastive loss considers the positive-negative relation depending on a set of criteria $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$. According to Eq. A.11, we define the attractor $\mathcal{A}(\mathbf{z}, \mathbf{z}')$ and repellor $\mathcal{R}(\mathbf{z}, \mathbf{z}')$ to describe the relativeness between the features of a given query-key pair $(\mathbf{z}, \mathbf{z}')$. Without the loss of generality, we add only one predictor $\mathcal{P}(*, \theta_p)$ instead of multiple predictors $\{\mathcal{P}(*, \theta_p^i)\}_{i=1}^H$ after query projection in our experiments for visualization, and set the weight $\alpha_i = \frac{1}{H}$. The number of criteria H is set to be 3. For ease of our visualization, we only visualize the pull-push dynamics between query prediction $\mathbf{q}$ and the key projection $\mathbf{z}'$, *i.e.*,

$$\frac{\partial \mathcal{L}_{RCL}}{\partial \mathbf{q}} = \sum_{i=1}^{H} \alpha_i \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] \right) \frac{\mathbf{z}'}{\tau}, \tag{B.1}$$

Concretely, the attractor $\mathcal{A}(\mathbf{q}, \mathbf{z}')$ and the repellor $\mathcal{R}(\mathbf{q}, \mathbf{z}')$ can be defined as

$$\mathcal{A}(\mathbf{q}, \mathbf{z}') = \sum_{i=1}^{H} \alpha_i \mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) \tag{B.2}$$

$$\mathcal{R}(\mathbf{q}, \mathbf{z}') = \sum_{i=1}^{H} \alpha_i \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] \tag{B.3}$$

Figure B.1(a) shows a query image (the first image in the column and row), two images that share the same label with the query image

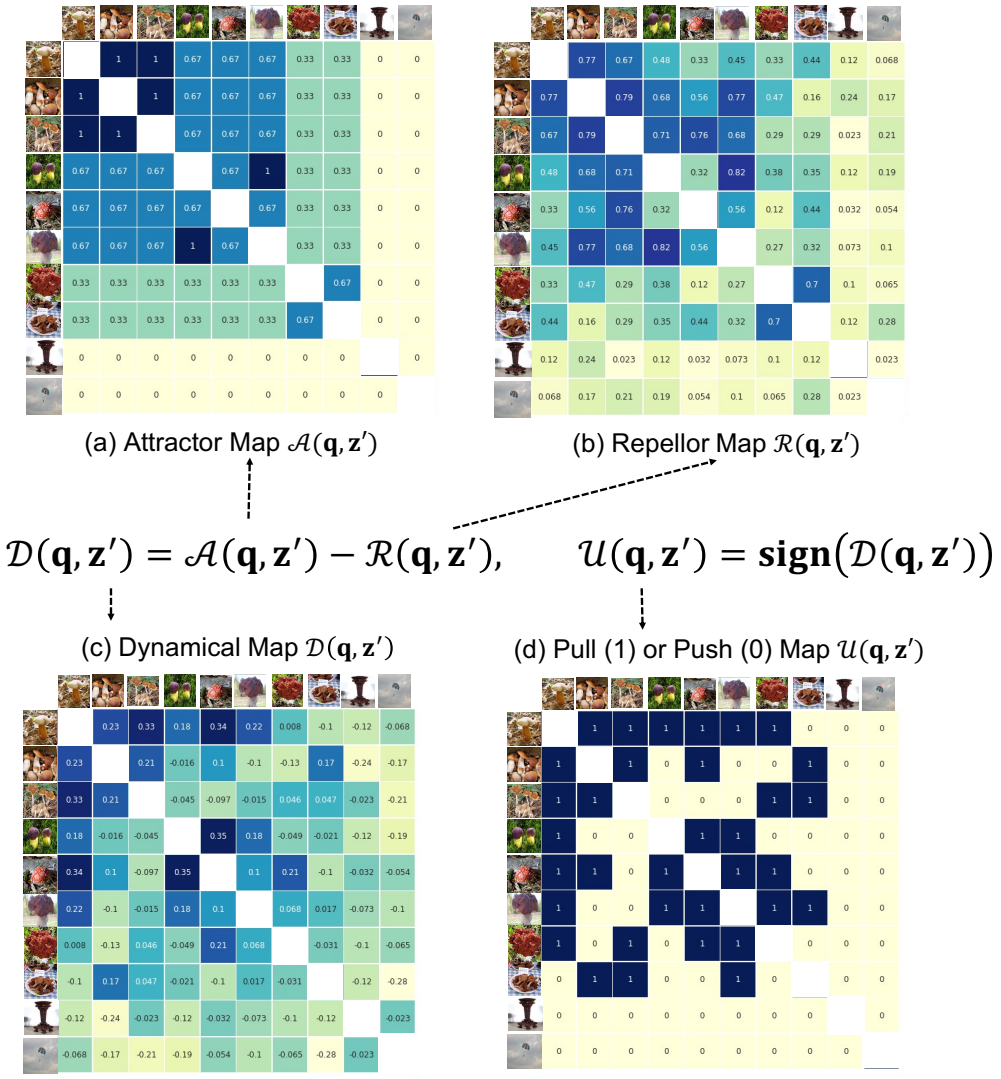


Figure B.1: Visualization of relative contrastive loss. (a) *Attractor Map* $\mathcal{A}(\mathbf{q}, \mathbf{z}')$ in Eq. B.2: Attractive map denotes the attractive force of relative contrastive loss that pulls query-key pair $(\mathbf{q}, \mathbf{z}')$ together. (b) *Repellor Map* $\mathcal{R}(\mathbf{q}, \mathbf{z}')$ in Eq. B.3: Repellor map denotes the repulsive force of relative contrastive loss that pushes query-key pair $(\mathbf{q}, \mathbf{z}')$ apart. (c) *Dynamical Map* $\mathcal{D}(\mathbf{q}, \mathbf{z}') = \mathcal{A}(\mathbf{q}, \mathbf{z}') - \mathcal{R}(\mathbf{q}, \mathbf{z}')$: the difference of the attractor map and the repellor map. Positive value means the query-key pair $(\mathbf{q}, \mathbf{z}')$ should be pulled together, the negative value means the query-key pair $(\mathbf{q}, \mathbf{z}')$ should be pushed apart. The absolute value of *dynamical map* means the strength of force. (d) *Pull or Push Map* $\mathcal{U}(\mathbf{q}, \mathbf{z}') = \text{sign}\,(\mathcal{D}(\mathbf{q}, \mathbf{z}'))$: Pull or Push map denotes the final attractive or repulsive force between a query-key pair $(\mathbf{q}, \mathbf{z}')$. 0 denotes pushing two features apart and 1 denotes pulling two feature together.

in the hierarchical label bank at all levels $h = 1, 2, 3$, three images that share the same label in the hierarchical label bank at level $h = 2, 3$, two images that only share the same label in hierarchical label bank at level $h = 3$, and two images that are not labeled the same with query image in the hierarchical label bank at any level. The results in Figure B.1(c) show that the final decision on pull (greater than 0) and push (smaller than 0) is continuous, different from the designs in [87, 33, 56, 31, 80], that are discrete. Besides, the continuous values reflect relative semantic and visual similarities among samples.

Appendix C

Label Propagation in Relative Contrastive Loss

Label Propagation [159] is a widely-adopted method of computing the possibility that two samples/clusters belong to the same class. Given n units $\mathcal{U} = \{\mathcal{U}_i\}_{i=1}^n$, *i.e.,* clusters or samples to be split/merged, we first estimate its pairwise similarities by the dot product of the unit prototypes $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n)$, *i.e.,* features for single images or cluster feature centers. Mathematically, it can be formulated as

$$\mathbf{A} = \mathbf{U}^\top \mathbf{U}. \tag{C.1}$$

Following [159], we can obtain the normalized affinity matrix $\hat{\mathbf{A}}$ by

$$\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}, \tag{C.2}$$

where $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{A}_{ij}$. We denote the predicted probabilities of samples/clusters as $\mathbf{P}^t = (\mathbf{p}_1^t, \mathbf{p}_2^t, ..., \mathbf{p}_n^t) \in \mathbb{R}^{n\times k}$ after t-th propagation (defined in Eq. C.4), where k is the number of classes (clusters) that n units may belong to, $\mathbf{p}_*^t = (p_{*,1}^t, p_{*,2}^t, ..., p_{*,k}^t)$ and $p_{*,k'}^t$ denotes the probability of the sample belong to the k'-th class. For the i-th unit, we would like to propagate the class predictions from other units j as

$$\mathbf{p}_i^{t+1} = \gamma \sum_{j\neq i} \hat{\mathbf{A}}_{ij}\mathbf{p}_j^t + (1-\gamma)\mathbf{p}_i^0 = \gamma \hat{\mathbf{A}}_i \mathbf{P}^t + (1-\gamma)\mathbf{p}_i^0, \tag{C.3}$$

where γ is a propagation strength parameter, $\mathbf{P}^0 = (\mathbf{p}_1^0, \mathbf{p}_2^0, ..., \mathbf{p}_n^0)$, $\mathbf{p}_i^0$ is the initial label prediction of the i-th unit that we will specifically define in the following cluster split and cluster merge.

Intuitively, if the i-th sample and the j-th sample are similar with a high affinity $\hat{\mathbf{A}}(i,j)$, the prediction $\mathbf{p}_j^t$ of the jth sample would have a larger weight to be propagated to the prediction $\mathbf{p}_i^{t+1}$ of the i-th sample. Propagating the predictions between all samples in parallel can be formulated as

$$\mathbf{P}^{t+1} = \gamma \hat{\mathbf{A}} \mathbf{P}^t + (1-\gamma)\mathbf{P}^0, \tag{C.4}$$

which is an iterative algorithm. The closed solution $\mathbf{P}^\infty$ after conducting Eq. C.4 for multiple times until convergence is

$$\mathbf{P}^\infty = (\mathbf{I} - \gamma \hat{\mathbf{A}})^{-1} \mathbf{P}^\mathbf{0}. \tag{C.5}$$

For each unit to be split or merged, we estimate its class prediction $\mathbf{p}_i^\infty = (p_{i,0}^\infty, p_{i,1}^\infty, ..., p_{i,k}^\infty)$ by propagating the neighboring information with Eq. C.5, which is used to merge the i-th unit to j-th cluster when $p_{i,j}^\infty > \sigma_m$. Here σ_m is the manually designed threshold for cluster merge.

Initialize $\mathbf{P}_0$ in Cluster Split. As described in Cluster Split part in Sec. 4.3 in the main text, we split a cluster $\mathcal{C}_i^{h+1}$ into m clusters, and uses the clusters at h-th level at its split units, *i.e.*, $\mathcal{U}_i^{h+1} = \{\mathcal{C}_j^h | \mathcal{C}_j^h \subset \mathcal{C}_i^{h+1}, j = 1, 2, ..., k^h\}$, where k^h is the number of clusters at h-th level. We re-denote $\mathcal{U}_i^{h+1} = \{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_n\}$, where $n = |\mathcal{U}_i^{h+1}|$. We first select m the most dissimilar split units in $\mathcal{U}_i^{h+1}$ as the prototypes of each class. Then, we initialize $(\mathbf{p}_{jk}^0)$ as 1 if $\mathcal{O}_j$ is selected as the prototype of the k-th class, and as 0 otherwise, *i.e.*,

$$\mathbf{p}_{jk}^0 = \begin{cases} 1, & \text{if } \mathcal{O}_j \text{ is selected as the prototype of the } k\text{-th class,} \\ 0, & \text{otherwise,} \end{cases} \tag{C.6}$$

Initialize $\mathbf{P}_0$ in Cluster Merge. We treat every cluster in the merge units $\mathcal{V}_i^{h+1}$ in Sec. 4.3 Cluster merge in the main text as an individual class, and then use label propagation to determine to merge two clusters if their prediction belonging to the same class is larger than σ_m. Specifically, we

initialize $\mathbf{P}^0$ as

$$\mathbf{P}^0 = \mathbf{I}_{n' \times n'}, \tag{C.7}$$

where $n' = |\mathcal{V}_i^{h+1}|$ is number of merge units in $\mathcal{V}_i^{h+1}$.

Effectiveness of Label Propagation. Label propagation serves as a cornerstone in cluster split and cluster merge for estimating the possibility that two samples/clusters labeled the same. To evaluate the effectiveness of label propagation in hierarchical clustering, we replace the label propagation by typical implementation, *i.e.,* feature similarity, in hierarchical clustering to justify whether two units belong to the same class. Average linkage based hierarchical clustering [175] determines merge and split by pairwise similarity only, thus can not consider the neighboring information in the data distribution. The detailed implementation of hierarchical clustering by average linkage is specifically described in supplementary materials. Comparing Exp. 2 with Exp. 3 and comparing Exp. 4 with Exp. 5 in S-Table 3.4, we find the accuracy with label propagation is about 6% higher than that clustered by average linkage if we set the hierarchy of clustering to 3. Comparing Exp 3, 5 and Exp 1, 2, 4 in S-Table 3.4, we find different trends when implementing the label propagation and average linkage, *i.e.,* the accuracy increases as the number of hierarchies increases for label propagation (Exp 1, 2, 4) but obviously drops for average linkage (Exp 3, 5). We attribute this to the failure of average linkage based clustering, and therefore the criteria by hierarchical clustering with label propagation determine query-key pair positive and negative incorrectly. This analysis shows the potential of designing more appropriate criteria as the future work when implementing relative contrastive loss in the feature.

Appendix D

Visualization of Feature Mixture and Feature Distribution

D.1 Visualization of Feature Mixtureness

We provide an intuitive understanding of the relation between Feature Mixtureness and the feature distribution distance by manually generating two sets of features with different distribution distance. We use red and blue to represent class centers from pre-D and eval-D, respectively. The visualization results are illustrated in Fig. D.1. From (a) to (c), when the distribution distance between pre-D and eval-D increases, Feature Mixtureness decreases accordingly. When we fix the variance of features in pre-D and gradually enlarge the variance of features in eval-D (from (d) to (f)), Feature Mixtureness will decrease as well. Based on the observations above, we conclude that our Feature Mixtureness can empirically measure the feature distribution distance between pre-D and eval-D.

D.2 Visualization of Feature Distribution

In this section, we provide an illustration to establish an intuition about how intra-class variation and Feature Mixtureness evolve during different pretraining epochs.

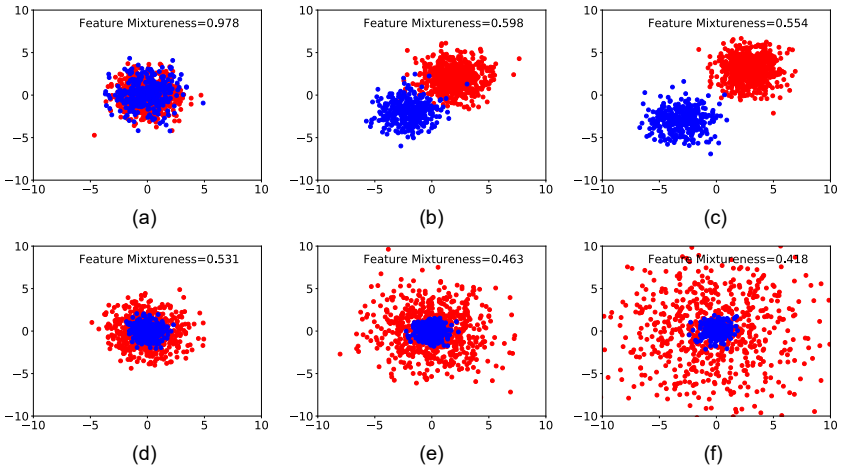


Figure D.1: Visualization of Feature Mixtureness with different manually generated feature distribution. Red and blue represent pre-D and eval-D class centers, respectively.

D.2.1 Intra-class Variation on pre-D

We visualize the feature distribution using samples from 10 randomly selected classes in pre-D in Fig. D.2 to illustrate the evaluation results of the intra-class variation on pre-D. Different colors represent different classes. In SL, the intra-class variation will continuously decrease to a small value with more training epochs. In contrast, the intra-class variance of SL-MLP and Byol retains even though we pretrain the networks at large pretraining epochs. This visualization graphically validates that the MLP projector can enlarge the intra-class variation of features in pre-D.

D.2.2 Feature Mixtureness between pre-D and eval-D

We randomly select features from 5 classes in pre-D and 5 classes in eval-D, and then visualize them by t-SNE in Fig. D.3. Cold colors represent features from pre-D and warm colors represent features from eval-D. At the early pretraining stage, all methods show high Feature Mixtureness as they cannot well classify images in pre-D. When the training epoch is becoming larger, SL shows lower Feature Mixtureness, which indicates a larger feature distribution distance between pre-D and eval-D. Instead, SL-MLP and Byol remain large Feature Mixtureness when the training

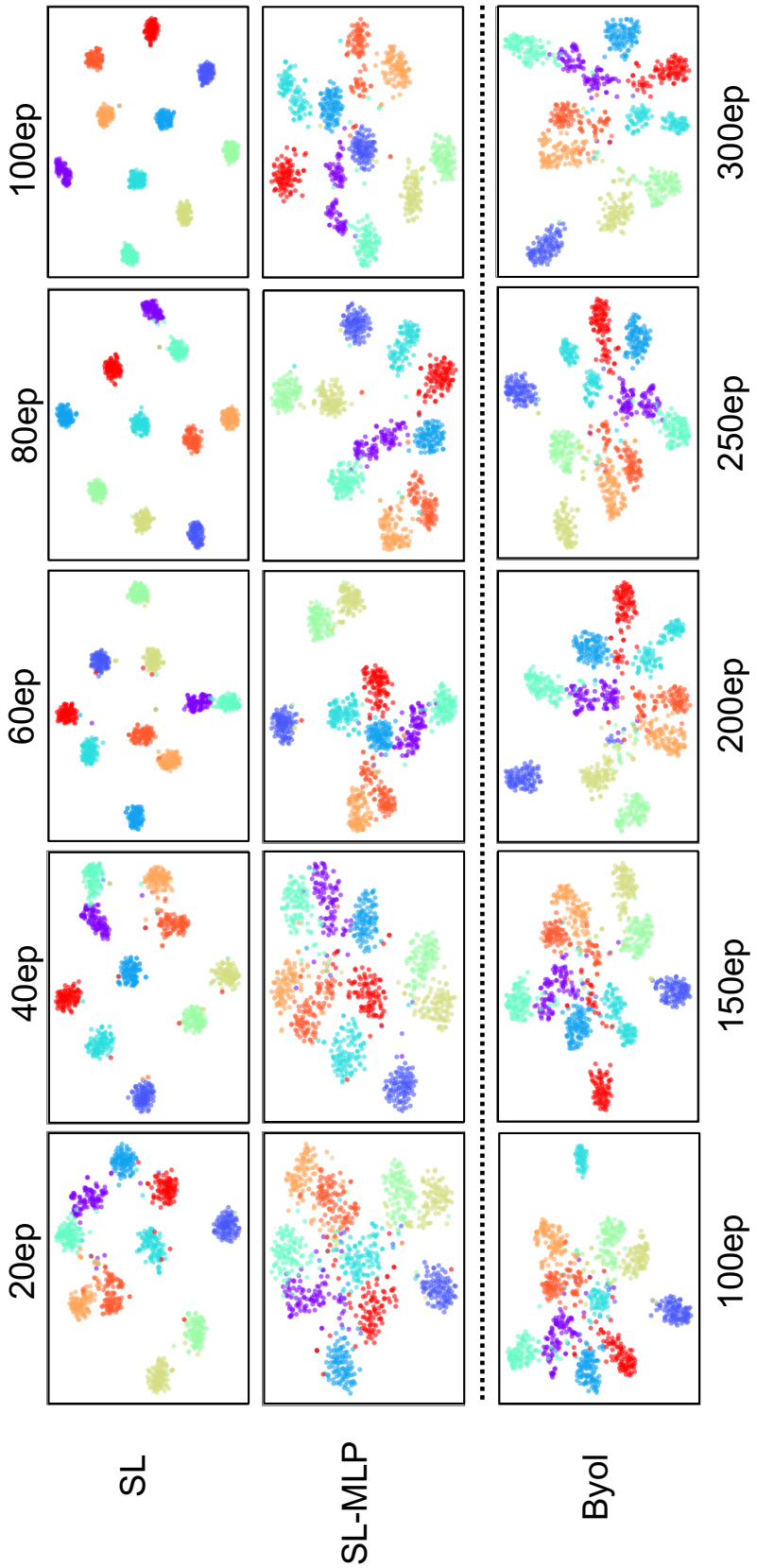


Figure D.2: Evolution of intra-class variation of features in pre-D with different epochs. Different colors denote different classes. The intra-class variation of SL will be very small when the pretraining epoch is large enough. Instead, the intra-class variation of SL-MLP and Byol still retains even though the model is pretrained by large epochs.

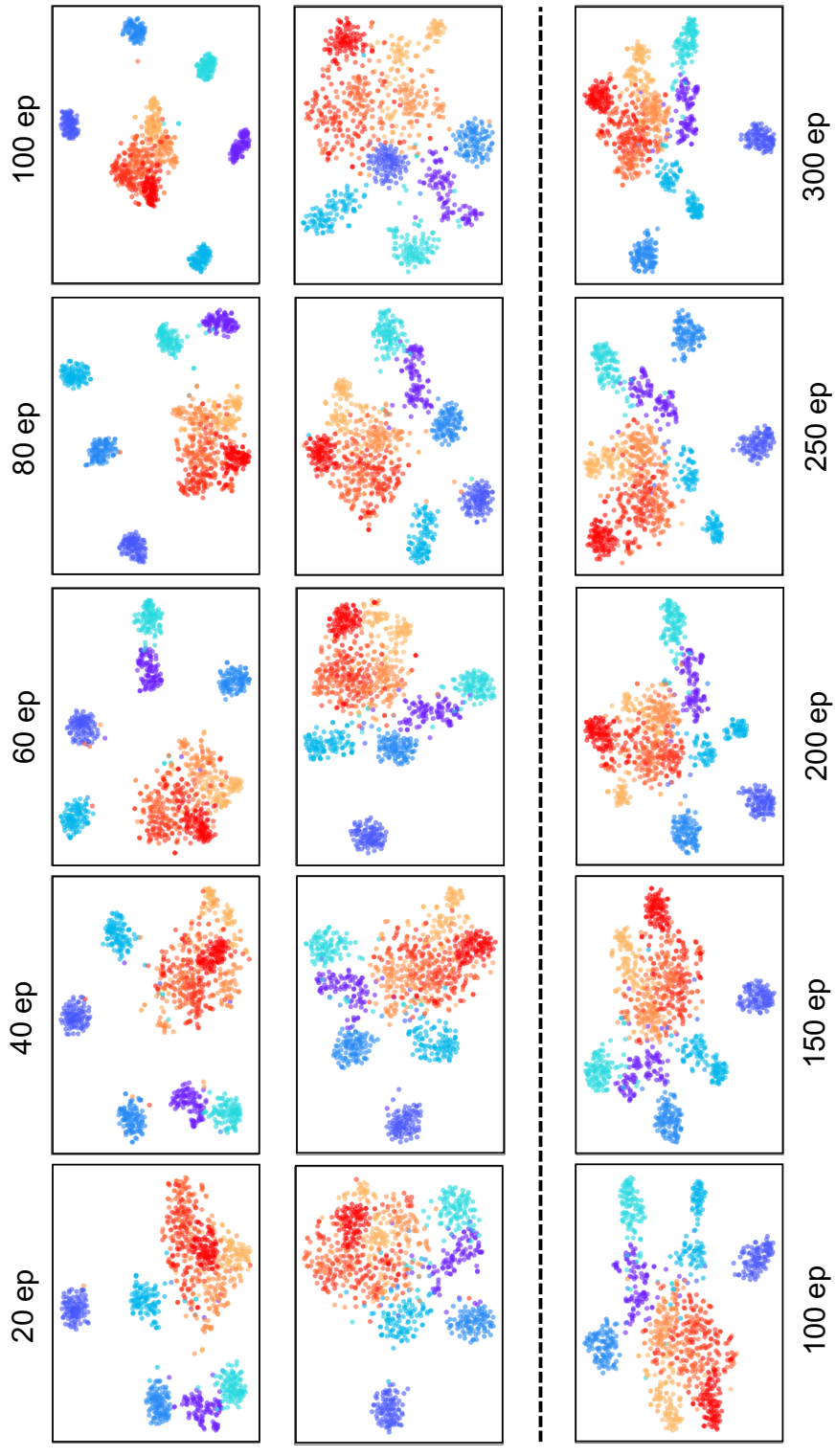


Figure D.3: Evolution of Feature Mixtureness between features from pre-D and from eval-D. Cold colors denote features from 5 classes that are randomly selected from pre-D, and warm colors denote features from 5 classes that are randomly selected from eval-D. Feature Mixtureness of SL continuously decrease during pretraining. Alternatively, SL-MLP and Byol keeps a relatively high Feature Mixtureness at large pretraining epochs.

epoch is becoming larger, which shows that the feature distribution distance between pre-D and eval-D is not enlarged by Byol and SL-MLP.

Appendix E

More Investigation of the Influences of MLP on Transferability

In this section, we provide the detailed analysis about how each component of the MLP projector influences the intra-class variation (represented by discriminative ratio ϕ^{pre}) on pre-D, Feature Mixtureness Π between pre-D and eval-D, and feature redundancy $\mathcal{R}$. Based on SL which does not include MLP, we ablate the structure of the MLP projector by adding the input fully connected layer, the output fully connected layer, the batch normalization layer and the ReLU layer incrementally. The input fully connected layer and the output fully connected layer are both set to have hidden units of 2048 and output dimensions of 2048 to keep same output feature dimensions as SL. All experiments are pretrained over 100 epochs. Testing results of the discriminative ratio on pre-D, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ are illustrated in Tab. E.1.

E.1 Visualization of intra-class variation

We randomly select features from 10 classes in pre-D and visualize their intra-class variation in Fig. E.1. Different colors denote features from different classes. We specify the components in the MLP projector below each visualization image. Comparing (a) with (b), we can see that adding a fully connected layer can slightly enlarge intra-class variation, which

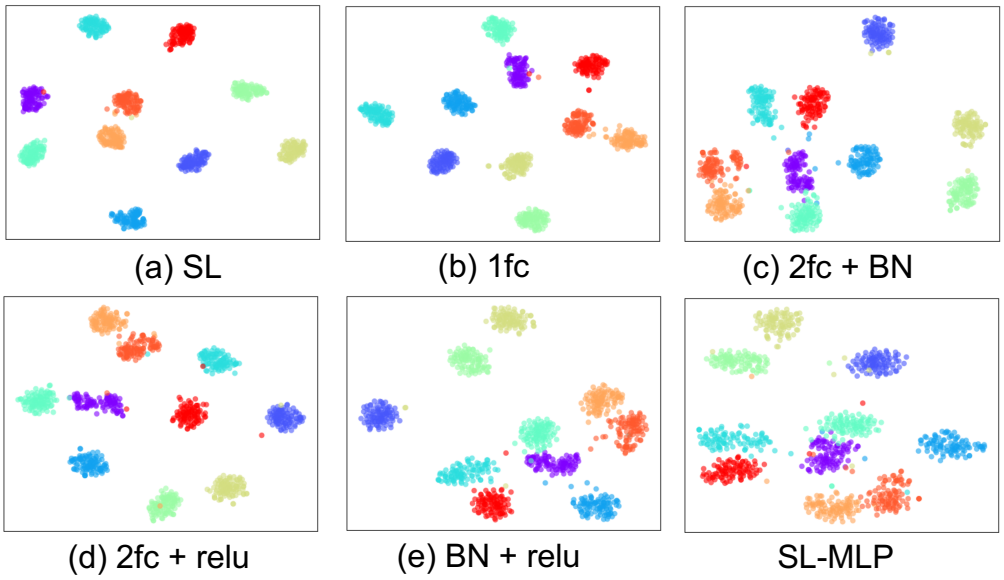


Figure E.1: Visualization of intra-class variation by different components. We randomly select 10 classes in pre-D. Different colors denote different classes. Comparing (a) wth (b), we can see the fully-connected layer can slightly help enlarge the intra-class variation. Comparing (a-b) and (d-e), we can observe the batch normalization layer and the ReLU layer can significantly enlarge the intra-class variation in the feature space. In general, all components in the MLP layer is beneficial to enlarge intra-class variation, which proves their effectiveness in enhancing transferaiblity of pretraining models.

indicates that linear transformation helps transferability marginally. Instead, comparing (a-b) with (c-e), we can observe that the batch normalization layer and the ReLU layer are important components in the MLP projector, which can significantly enlarge the intra-class variation in the feature space of pre-D. In general, comparing SL-MLP with (a-e), we can conclude that all components in MLP projector help enlarge the intra-class variation of features in pre-D while the batch normalization layer and the ReLU layer play the most important roles.

E.2 Visualization of Feature Mixtureness

We randomly select features from 5 classes in pre-D and 5 classes in eval-D to visualize Feature Mixtureness with different MLP components. The

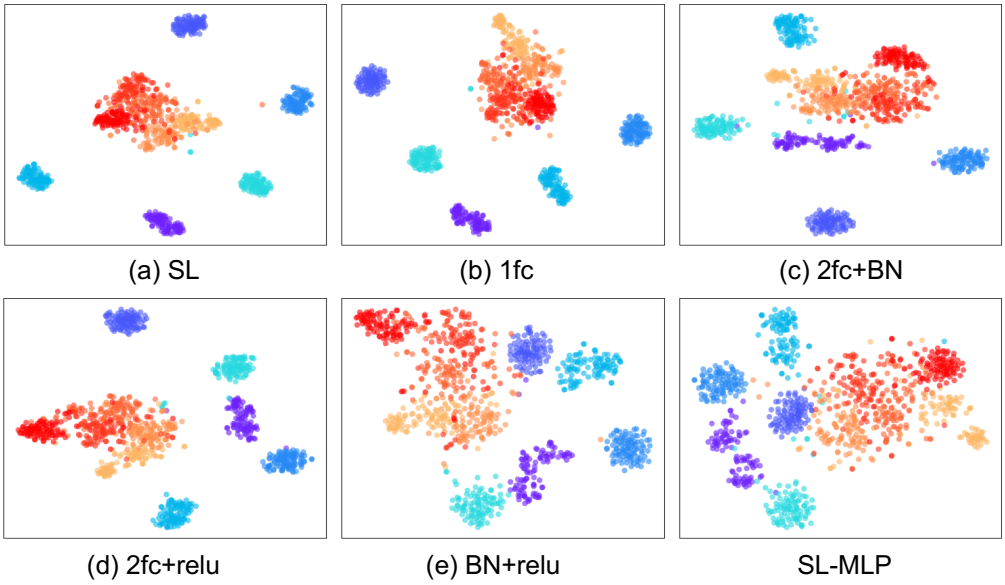


Figure E.2: Visualization of Feature Mixtureness of features pretrained by different MLP components. Different colors denote different classes. Points with cold colors denote the features from pre-D, and points with warm colors denote the features from eval-D. Comparing (c-d) with (a-b), we can see that adding BN and ReLU can increase Feature Mixtureness between pre-D and eval-D. Comparing (e) with (a-d), we can conclude that BN and ReLU play the main roles in the MLP projector as (e) shows larger Feature Mixtureness. An MLP projector with all components achieves the largest Feature Mixtureness.

results are summarized in Fig. E.2. The features with cold colors come from pre-D, the features with warm colors come from eval-D. Comparing (a) and (b), we can see adding a fully connected layer can hardly increase Feature Mixtureness between pre-D and eval-D. Comparing (c-d) with (b), we can conclude that the batch normalization layer and the ReLU layer can increase Feature Mixtureness between pre-D and eval-D. Comparing (b-d) with (e), we can summarize that the batch normalization and the ReLU layer are the most important components. A batch normalization layer with a ReLU layer can significantly increase Feature Mixtureness between pre-D and eval-D, which has already been similar to Feature Mixtureness when the MLP projector has the complete architectural.

E.3 Quantitative Analyse of MLP components

With the discriminative ratio ϕ^{pre}, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ defined in main text Sec. 4.2, we quantitatively examine the effect of different components in the MLP projector. The results are presented in Tab. E.1. Firstly, the fully connected layer has little influence on three metrics. Comparing (a) and (b), when adding a fully connected layer, the model shows slight improvement on Feature Mixtureness and feature redundancy, and slight decrease of the discriminative ratio on pre-D. Second, non-linear layer brings considerable improvements. Comparing (b) to (d), we can summarize that incrementally adding a ReLU, a batch normalization layer can increase Feature Mixtureness, reduce discriminative ratio, which could improve transferability of the pretrained model. Specifically, the ReLU layer brings a little improvement on feature redundancy. Comparing (a,b) with (c,e), we can conclude that BN not only reduces the discriminative ratio on pre-D, but also increases Feature Mixtureness. BN has a significant influence on future redundancy, which reduces feature redundancy by 50% (from 0.0671 to 0.0369). Last but not least, the combination of all components achieves the best transferability with the lowest feature redundancy, the highest Feature Mixtureness and a relatively large intra-class variation.

Exp	Components				Top-1	$D^{pre}_{inter}/D^{pre}_{intra}$	$\Pi(\uparrow)$	$\mathcal{R}(\downarrow)$
	Input FC	BN	ReLU	Output FC				
(a)					55.9	2.034	0.515	0.0776
(b)	✓				56.6	1.505	0.679	0.0671
(c)	✓	✓		✓	61.0	1.269	0.870	0.0369
(d)	✓		✓	✓	60.1	1.362	0.804	0.0654
(e)		✓	✓		60.5	1.045	0.846	0.0369
SL-MLP	✓	✓	✓	✓	62.5	1.124	0.871	0.0351

Table E.1: Quantitative analysis of structural design of inserted MLP, including discriminative ratio on pre-D, Feature Mixtureness Π and feature redundancy $\mathcal{R}$. (b-e) denote experiments in which different components are added on the SL baseline (a). When incrementally adding components of the MLP into SL, the distriminative ratio on pre-D and feature redundancy will decrease while the Feature Mixtureness will increase.

Appendix F

More Details of HumanBench

In the main text, we briefly introduce the number of images and tasks in the pretraining dataset of HumanBench. To evaluate HumanBench, we introduce the evaluation scenario and protocols. In this section, we present detailed information on the pretraining dataset and evaluation dataset and discuss the ethical issues of these datasets.

F.1 Dataset Statistics of HumanBench

HumanBench collects 37 publicly available datasets of 5 human-centric tasks, including person ReID, human parsing, pose estimation, pedestrian detection, and pedestrian attribute. More details can be seen in Table F.1. The existing distribution of datasets includes large numbers of human-centric cropped images in ReID, video frames in person pose estimation, and human parsing. In particular, we select a single frame from every 8 video frames to avoid information reduction. Except for using training images in all datasets, we also use all/partial test images in some datasets. Specifically, for the person Reid task, we use all test images in LaST and partial test images in the PRCC dataset; for the human parsing task, we only use train images and publicly released images in DeepFashion($\sim$half of the dataset reported in [161]). For the pedestrian detection dataset, we remove the images in which there is no person. For the pose estimation datasets, we only use train images. We only use partial test images in the UAV-Human dataset for the pedestrian attribute recognition dataset and do not contain test images in other pedestrian attribute recognition datasets. All the images in the pretraining dataset

have been de-duplicated with the testing datasets to be a meaningful benchmark of our HumanBench.

F.2 Discussion of Ethical Issues

The usage of HumanBench might bring several risks, such as privacy, and problematic content. We discuss these risks and their mitigation strategies as follows.

Copyright. All images in this paper and dataset are collected by publicly available. We claim the dataset:

- Copy and redistribute the material in any medium or format.
- Remix, transform and build on the material for any commercial purpose.

Referring to OmniBenchmark [295], MS-COCO [149], Kinetics-700 [26], we only present the lists of URLs and their corresponding metainformation to our HumanBench.

F.3 Details of HumanBench-Subset

Due to the high computational cost when we pretrain the model on the full dataset, we select 17 subsets from 37 full datasets for ablation study, which contains 1,270,186 images as a similar number with ImageNet-1K($\sim$1.28M). Table F.1 summarizes the statistics of the HumanBench-Subset. For the person ReID task, we select widely-used Market1501 and CUHK03 datasets and the clothes-changing ReID dataset PRCC, forming 38,197 images. We select widely used Human3.6M, LIP, CIHP, VIP datasets, and one clothes parsing dataset, *i.e.*, ModaNet, with 192,124 images for the human parsing task. We select widely-used COCO, AIC, and PoseTrack datasets with 748,812 images for the pose estimation task. For the attribute task, we select PA-100K, RAPv2, and Market1501-Attribute datasets with a total of 170,879. Due to the significant resource cost, we

only selected one widely used dataset CrowdHuman for the pedestrian detection task.

Partition	Task	Name	Num of samples	Task	Name	Num of samples
Full	ReID	Market1501 [304]	12,936	Detection	WIDER Pedestrian [163]	57,999
		CUHK03 [137]	7,365	Pose	COCO [149]	262,465
		MSMT [255]	30,248		AIC [257]	378,374
		LaST [204]	71,248		PoseTrack [5]	107,973
		PRCC [272]	17,896		JRDB [237]	310,035
		DGMarket [308]	128,306		MHP [131]	41,128
		LUPerson-NL [66]	5,178,420		UppenAction [292]	163,839
	Parsing	Human3.6M [107]	62,668		Halpe [61]	41,712
		LIP [78]	30,462		3dpw [238]	74,620
		CIHP [77]	28,280		MPI-INF-3DHP [169]	1,031,701
		VIP [312]	18,469		Human3.6M [107]	312,187
		Paper Doll [271]	1,035,825		AIST++ [133]	1,015,257
		DeepFashion [161]	191,961	Attribute	PA100K [158]	90,000
		ModaNet [305]	52,245		RAPv2 [130]	67,943
	Detection	CrowdHuman [201]	15,000		HARDHC [142]	28,336
		WiderPerson [291]	9,000		UVA-Human [135]	16,183
		COCO-person [149]	64,115		Parse27k [212]	27,482
		EuroCity Persons [19]	21,795		Market1501-Attribute [304]	12,936
		CityPersons [290]	2,778	**Total**	**11,019,187**	
Subset	ReID	Market1501 [304]	12,936	Pose	COCO [149]	262,465
		CUHK03 [137]	7,365		AIC [257]	378,374
		PRCC [272]	17,896		PoseTrack [5]	107,973
	Parsing	Human3.6M	62,668	Detection	CrowdHuman [201]	15,000
		LIP [78]	30,462	Attribute	PA100K [158]	90,000
		CIHP [77]	28,280		RAPv2 [130]	67,943
		VIP [312]	18,469		Market1501-Attribute [304]	12,936
		ModaNet [305]	52,245	**Total**	**1,165,012**	

Table F.1: Dataset statistics of pretraining datasets

Appendix G

Detailed Implementations of PATH

During pretraining, we collect in total of 39 datasets from person ReID, human parsing, pose estimation, pedestrian attribute recognition, and pedestrian detection. To pretrain the model in a distributed manner, we only train a dataset in each GPU. We pretrain our model using 64 V100-32G GPUs. In the following, we present the task-agnostic parameters and task-specific parameters.

G.1 Task-agnostic Hyperparameters

Table G.1 illustrates the learning hyper-parameters utilized in our pre-training stage. Specifically, we train our model for 80000 iterations in total. During pretraining, we use **STEP** learning rate decay strategy with a warm-up from $1e^{-7}$ to $5e^{-4}$ during 1500 iterations. we multiply the learning rate $5e^{-4}$ by 0.5, 0.2 and 0.1 at the 40000-th, 60000-th and 76000-th iteration, respectively. The backbone multiplier and the positional multiplier are the ratios of the actual learning rate of the backbone and the positional embedding, respectively, which are all set as 1.0.

G.2 Task-specific Hyperparameters

Table G.2 presents the task-specific hyper-parameters of each dataset, including batch size per GPU, the number of GPUs, sample weights,

lr_schedule	type	Step
	base_lr	1.00E-07
	warmup_steps	1500
	warmup_lr	5.00E-04
	lr_mults	[0.5, 0.2, 0.1]
	lr_steps	[40000, 60000, 76000]
	max_iter	80000
	backbone_multiplier	1.0
	pos_embed_multiplier	1.0
optimizer	type	Adafactor_dev
	beta1	0.9
	clip_beta2	0.999
	clip_threshold	0.5
	decay_rate	-0.8
	scale_parameter	FALSE
	relative_step	FALSE
	weight_decay	0.05
layer_decay	num_layers	12
	layer_decay_rate	0.75

Table G.1: Detailed description of task-agnostic hyper-parameters in the pretraining stage.

and loss weights. Specifically, the dataset weights are related to sample weights and the number of GPUs:

$$\text{loss weight} = \text{sample weight} \times \text{images per GPU} \times \text{number of GPUs}. \tag{G.1}$$

The loss weights of the pose estimation are larger than other tasks because the loss functions used in pose estimation are MSE loss between the predicted heatmaps of keypoints and the heatmaps of the ground truth whose value is very small. For tasks other than pose estimation, the difference between different datasets among different tasks are relatively small.

G.3 Data Augmentation

We apply augmentation techniques to human-centric images, ranging from scene images in pedestrian detection to cropped images in person ReID. Here, we list the augmentations below for different tasks.

Person ReID. For person ReID, we use the same augmentation as in [165]. Specifically, we use the random horizontal flip and random erasing for pretraining. Finally, we resize the input image to size 256×128.

Pose Estimation. For pose estimation, we use the same augmentation as ViTPose[270]. Specifically, we use random horizontal flip, half-body transform, and random scale rotation for pretraining. Finally, we resize the input image to size 256×192.

Human Parsing. For human parsing, we use the same augmentation as in [77]. Specifically, we use random crop, random image rotation, and photometric distortion augmentation for pretraining. Particularly, for the human parsing dataset, we also use horizontal random flip augmentation, *e.g.,* Human3.6M, LIP, CIHP, LIP, VIP. Finally, we resize the input image to size 480×480.

Pedestrian Attribute Recognition. For pedestrian attribute recognition, we use the same augmentation as in [136]. Specifically, we use random crop and random horizontal flip augmentation for pretraining. Finally, we resize the input image to size 256×192.

Pedestrian Detection. For pedestrian detection, we use the same augmentation as in [303]. Specifically, we use random horizontal flips and random crop augmentation for pretraining. Finally, we random resize the input image with the longest side bound of 1333 and the shortest side bound of 800 while keeping the height and width ratio.

Crowd Counting. For the crowd counting dataset, we use random horizontal flip, random scaling ($0.5\times \sim 2\times$), and random cropping augmentation for pretraining.

G.4 Details of Implementations in Evaluation

For full finetuning, we carefully tune the learning rate $\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$, the weight decay $\{0.05, 0.1, 0.3\}$, drop path rate $\{0.1, 0.3, 0.5\}$, the backbone multiplier $\{0.1, 0.3, 0.5\}$, and report the best performance. We will provide the exact hyperparameters in our released repository after acceptance. For head finetuning and partial finetuning, we specifically set the weight decay as 0, which empirically proved very important in our experiments.

Task	Dataset	Batch Size Per GPU	GPU	Sample Weight	Loss Weight
ReID	Market1501+MSMT+CUHK03	112	1	5	560
	DGMarket+LaST+PRCC	96	1	0.1	9.6
	LUPerson-NL	192	2	1	384
Pose	COCO	224	2	8000	3584000
	AIC	224	2	6000	2688000
	PoseTrack	224	1	6000	1344000
	JRDB	224	1	4000	896000
	MHP	96	1	4000	384000
	UppenAction	128	1	4000	512000
	MPI-INF-3DHP	128	1	4000	512000
	Halpe	64	1	2000	128000
	3dhp	128	1	2000	256000
	Human3.6M	128	1	2000	256000
	AIST++	128	1	2000	256000
Parsing	Human3.6M	26	3	20	1560
	LIP	18	2	20	720
	CIHP	24	2	20	960
	VIP	16	1	20	320
	Paper Doll	24	2	15	720
	DeepFashion	32	2	15	960
	ModaNet	32	1	15	480
Attribute	rap2+pa100k	128	1	0.1	12.8
	HARDHC+UAV-Human+Parse27k+Market1501-Attribute	116	1	0.1	11.6
Detection	CrowdHuman	2	16	10	320
	WidePerson+COCO-person+EuroCity Persons+CityPersons	2	16	10	320

Table G.2: Detailed Implementation about Task-specific Hyper-parameters

Appendix H

Visualization of Task-Specific Features

To visualize the features attended by the task-specific projectors, we plot the heatmap of L2-normalization of the channels of the attended features. The red color in Figure H.1, H.2, H.3 show the important region, which leads to three conclusions. First, the highlighted regions in the pose estimation and the human parsing locates at the joints of human bodies, which shows that these two tasks are very similar. Second, the heatmap for pedestrian detection includes the whole person, which is consistent with the goal of pedestrian detection to detect all people. Third, for the pedestrian attribute recognition, we can see that the heatmap highlights the attributes, *e.g.*, gloves, bags. These highlighted regions instead of the whole body are also consistent with the goal of pedestrian attribute recognition to recognize attributes.

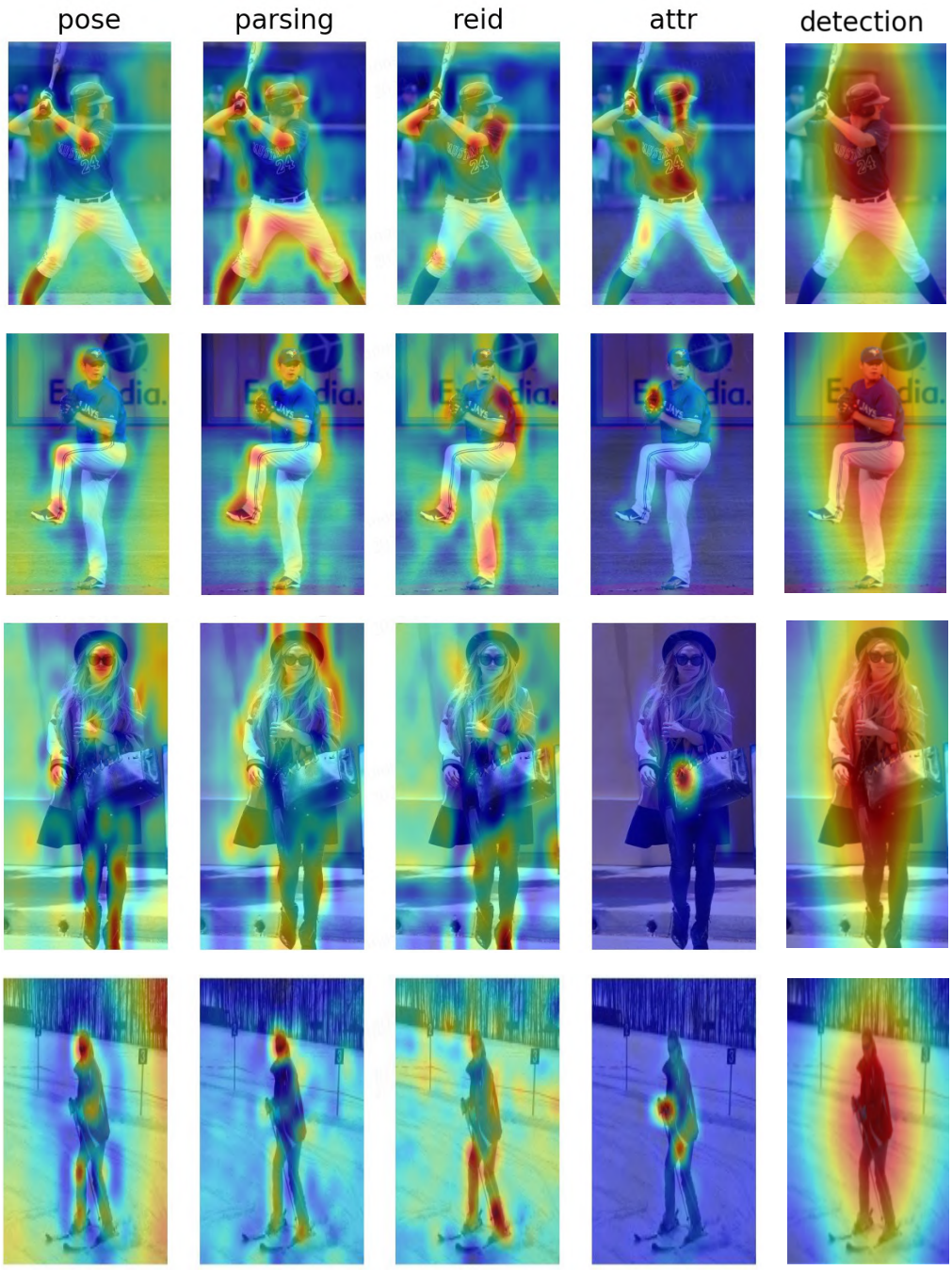


Figure H.1: Visualization of features after the task-specific projectors.

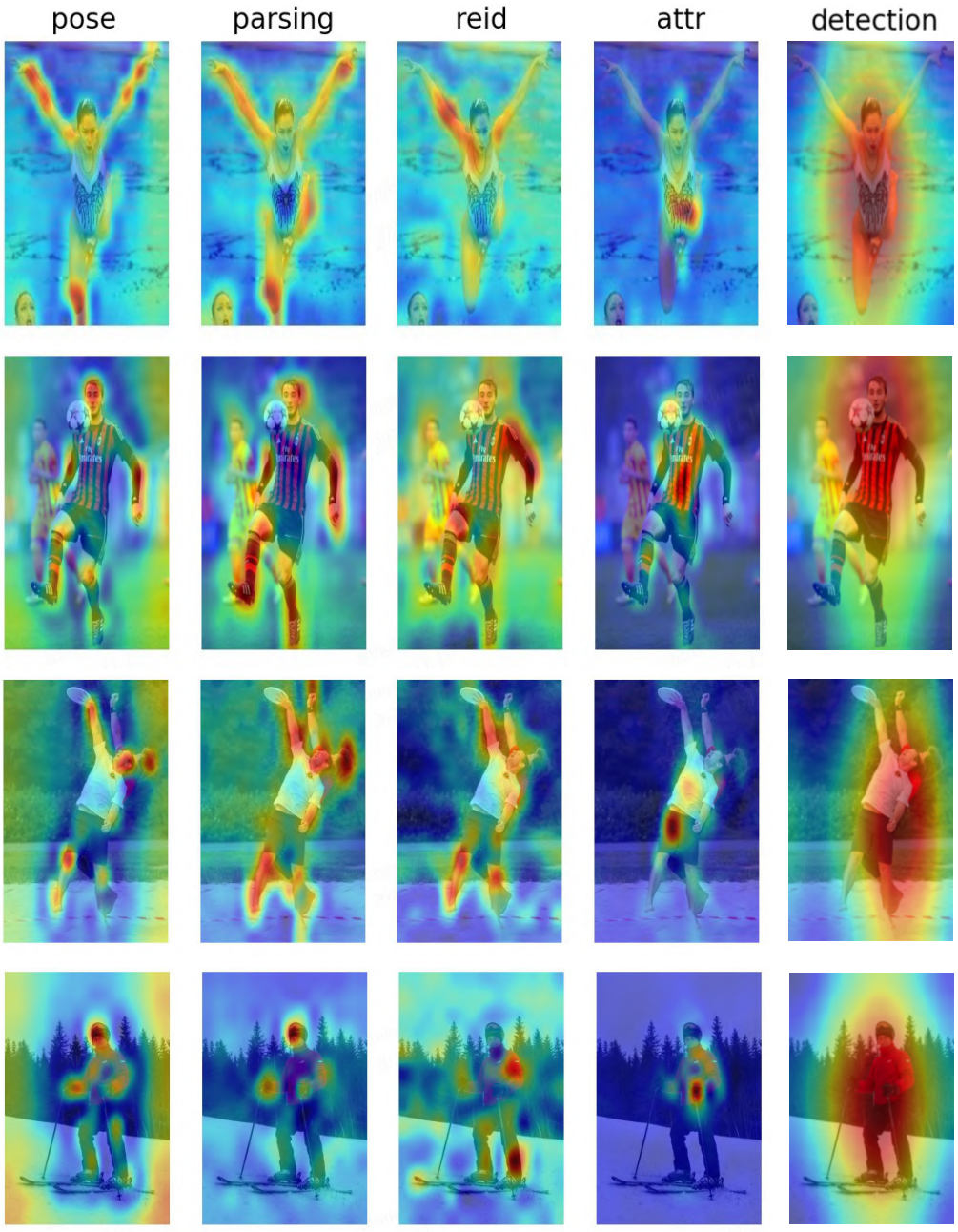


Figure H.2: Visualization of features after the task-specific projectors.

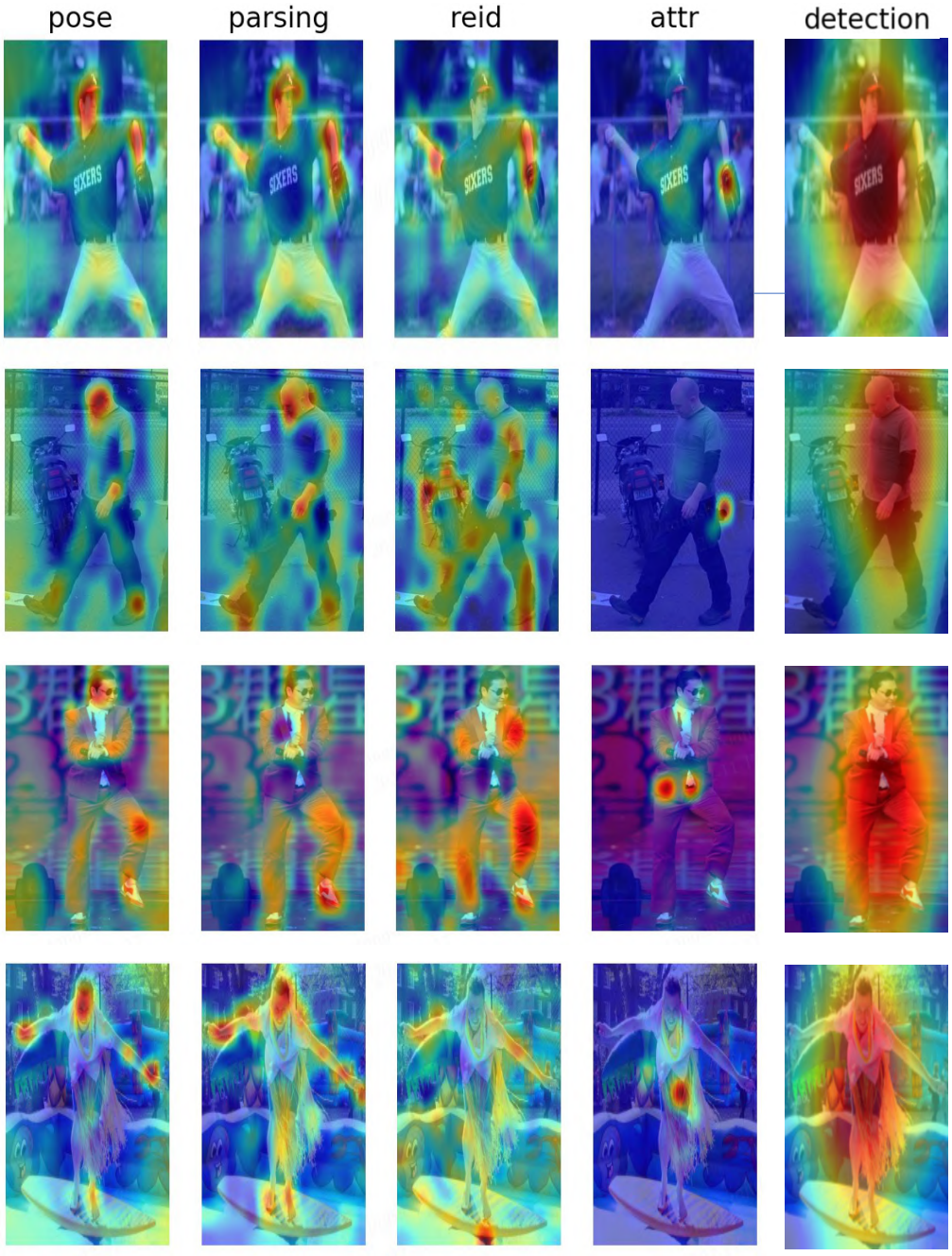


Figure H.3: Visualization of features after the task-specific projectors.

Bibliography

[1] A. Achille, G. Paolini, G. Mbeng, and S. Soatto. "The information complexity of learning tasks, their structure and their distance". In: *Information and Inference: A Journal of the IMA* 10.1 (2021), pp. 51–72.

[2] P. Agrawal, J. Carreira, and J. Malik. "Learning to see by moving". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 37–45.

[3] P. Agrawal, R. Girshick, and J. Malik. "Analyzing the performance of multilayer neural networks for object recognition". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*. Springer. 2014, pp. 329–344.

[4] J.-B. Alayrac, J. Carreira, and A. Zisserman. "The visual centrifuge: Model-free layered video representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2457–2466.

[5] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele. "Posetrack: A benchmark for human pose estimation and tracking". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5167–5176.

[6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. "2d human pose estimation: New benchmark and state of the art analysis". In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014, pp. 3686–3693.

[7] S. Atito, M. Awais, and J. Kittler. "Sit: Self-supervised vision transformer". In: *arXiv preprint arXiv:2104.03602* (2021).

[8] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. "From generic to specific deep representations for visual recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 36–45.

[9] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[10] S. Balakrishnama and A. Ganapathiraju. "Linear discriminant analysis-a brief tutorial". In: *Institute for Signal and information Processing* 18.1998 (1998), pp. 1–8.

[11] H. Bao, L. Dong, S. Piao, and F. Wei. "Beit: Bert pre-training of image transformers". In: *arXiv preprint arXiv:2106.08254* (2021).

[12] J. Baxter. "A model of inductive bias learning". In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198.

[13] E. F. Beckenbach, R. Bellman, and R. E. Bellman. *An introduction to inequalities*. Tech. rep. Mathematical Association of America Washington, DC, 1961.

[14] S. Ben-David and R. Schuller. "Exploiting task relatedness for multiple task learning". In: *Learning theory and kernel machines*. Springer, 2003, pp. 567–580.

[15] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. "Birdsnap: Large-scale fine-grained visual categorization of birds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2011–2018.

[16] H. Bilen and A. Vedaldi. "Universal representations: The missing link between faces, text, planktons, and cat breeds". In: *arXiv preprint arXiv:1701.07275* (2017).

[17] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. "Learning Bounds for Domain Adaptation". In: *Advances in Neural Information Processing Systems* 20 (2007), pp. 129–136.

[18] L. Bossard, M. Guillaumin, and L. Van Gool. "Food-101–mining discriminative components with random forests". In: *European conference on computer vision*. Springer. 2014, pp. 446–461.

[19] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila. "The eurocity persons dataset: A novel benchmark for object detection". In: *arXiv preprint arXiv:1805.07193* (2018).

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[21] J. Cao, X. Lin, S. Guo, L. Liu, T. Liu, and B. Wang. "Bipartite graph embedding via mutual information maximization". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 635–643.

[22] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149.

[23] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. "Unsupervised pre-training of image features on non-curated data". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2959–2968.

[24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *arXiv preprint arXiv:2006.09882* (2020).

[25] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. "Emerging properties in self-supervised vision transformers". In: *arXiv preprint arXiv:2104.14294* (2021).

[26] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. "A short note on the kinetics-700 human action dataset". In: *arXiv preprint arXiv:1907.06987* (2019).

[27] R. Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[28] D. Chakrabarty and S. Khanna. "Better and simpler error analysis of the Sinkhorn–Knopp algorithm for matrix scaling". In: *Mathematical Programming* (2020).

[29] D. Chen, G. Hua, F. Wen, and J. Sun. "Supervised transformer network for efficient face detection". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer. 2016, pp. 122–138.

[30] K. Chen, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung. "Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7546–7554.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[33] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. "Big self-supervised models are strong semi-supervised learners". In: *arXiv preprint arXiv:2006.10029* (2020).

[34] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang. "Context autoencoder for self-supervised representation learning". In: *arXiv preprint arXiv:2202.03026* (2022).

[35] X. Chen, H. Fan, R. Girshick, and K. He. "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297* (2020).

[36] X. Chen and K. He. "Exploring simple siamese representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15750–15758.

[37] X. Chen, S. Xie, and K. He. "An empirical study of training self-supervised visual transformers". In: *arXiv e-prints* (2021), arXiv–2104.

[38] X. Chen, W. Chen, T. Chen, Y. Yuan, C. Gong, K. Chen, and Z. Wang. "Self-pu: Self boosted and calibrated positive-unlabeled training". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1510–1519.

[39] G. Cheng, J. Han, and X. Lu. "Remote sensing image scene classification: Benchmark and state of the art". In: *Proceedings of the IEEE* 105.10 (2017), pp. 1865–1883.

[40] X. Chu, A. Zheng, X. Zhang, and J. Sun. "Detection in crowded scenes: One proposal, multiple predictions". In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 12214–12223.

[41] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. "Describing textures in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3606–3613.

[42] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al. "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)". In: *arXiv preprint arXiv:1902.03368* (2019).

[43] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.

[44] C. Darwin and P. Prodger. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[46] Y. Deng, P. Luo, C. C. Loy, and X. Tang. "Pedestrian attribute recognition at far distance". In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 789–792.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[48] C. Doersch, A. Gupta, and A. A. Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[49] C. Doersch and A. Zisserman. "Multi-task self-supervised visual learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2051–2060.

[50] P. Dollar, C. Wojek, B. Schiele, and P. Perona. "Pedestrian detection: An evaluation of the state of the art". In: *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2011), pp. 743–761.

[51] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo. "Peco: Perceptual codebook for bert pre-training of vision transformers". In: *arXiv preprint arXiv:2111.12710* (2021).

[52] M. Donoser and H. Bischof. "Diffusion processes for retrieval revisited". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 1320–1327.

[53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[54] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. "Discriminative unsupervised feature learning with exemplar convolutional neural networks". In: *IEEE transactions on pattern analysis and machine intelligence* (2015).

[55] P. Druzhkov and V. Kustikova. "A survey of deep learning methods and software tools for image classification and object detection". In: *Pattern Recognition and Image Analysis* 26 (2016), pp. 9–15.

[56] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations". In: *arXiv preprint arXiv:2104.14548* (2021).

[57] K. Dwivedi and G. Roig. "Representation similarity analysis for efficient task taxonomy & transfer learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12387–12396.

[58] M. Ereshefsky. *The poverty of the Linnaean hierarchy: A philosophical study of biological taxonomy*. Cambridge University Press, 2000.

[59] L. Ericsson, H. Gouk, and T. M. Hospedales. "How Well Do Self-Supervised Models Transfer?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5414–5423.

[60] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[61] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu. "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time". In: *arXiv preprint arXiv:2211.03375* (2022).

[62] Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu. "Seed: Self-supervised distillation for visual representation". In: *arXiv preprint arXiv:2101.04731* (2021).

[63] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *2004 conference on computer vision and pattern recognition workshop*. IEEE. 2004, pp. 178–178.

[64] Y. Feng, J. Jiang, M. Tang, R. Jin, and Y. Gao. "Rethinking supervised pre-training for better downstream transferring". In: *arXiv preprint arXiv:2110.06014* (2021).

[65] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn. "Efficiently Identifying Task Groupings for Multi-Task Learning". In: *arXiv preprint arXiv:2109.04617* (2021).

[66] D. Fu, D. Chen, J. Bao, H. Yang, L. Yuan, L. Zhang, H. Li, and D. Chen. "Unsupervised pre-training for person re-identification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 14750–14759.

[67] Y. Gao, X. Yu, and H. Zhang. "Graph clustering using triangle-aware measures in large networks". In: *Information Sciences* 584 (2022), pp. 618–632.

[68] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille. "Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative

dimensionality reduction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3205–3214.

[69] L. Gatys, A. S. Ecker, and M. Bethge. "Texture synthesis using convolutional neural networks". In: *Advances in neural information processing systems* 28 (2015).

[70] L. A. Gatys, A. S. Ecker, and M. Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[71] Y. Ge, D. Chen, and H. Li. "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification". In: *arXiv preprint arXiv:2001.01526* (2020).

[72] Y. Ge, F. Zhu, D. Chen, R. Zhao, et al. "Self-paced contrastive learning with hybrid memory for domain adaptive object re-id". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11309–11321.

[73] G. Ghiasi, B. Zoph, E. D. Cubuk, Q. V. Le, and T.-Y. Lin. "Multi-Task Self-Training for Learning General Representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8856–8865.

[74] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord. "Learning representations by predicting bags of visual words". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6928–6938.

[75] S. Gidaris, P. Singh, and N. Komodakis. "Unsupervised representation learning by predicting image rotations". In: *arXiv preprint arXiv:1803.07728* (2018).

[76] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[77] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin. "Instance-level human parsing via part grouping network". In: *Proceedings*

of the European conference on computer vision (ECCV). 2018, pp. 770–785.

[78] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin. "Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 932–940.

[79] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. "Scaling and benchmarking self-supervised visual representation learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6391–6400.

[80] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. "Bootstrap your own latent: A new approach to self-supervised learning". In: *arXiv preprint arXiv:2006.07733* (2020).

[81] Y. Guo, Y. Li, L. Wang, and T. Rosing. "Depthwise convolution is all you need for learning multiple visual domains". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8368–8375.

[82] T. Han, L. Bai, J. Gao, Q. Wang, and W. Ouyang. "DR. VIC: Decomposition and Reasoning for Video Individual Counting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3083–3092.

[83] T. Han, W. Xie, and A. Zisserman. "Self-supervised co-training for video representation learning". In: *arXiv preprint arXiv:2010.09709* (2020).

[84] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao. "Generalizable pedestrian detection: The elephant in the room". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11328–11337.

[85] K. Hassani and A. H. Khasahmadi. "Contrastive multi-view representation learning on graphs". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4116–4126.

[86] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. "Masked autoencoders are scalable vision learners". In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 16000–16009.

[87] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.

[88] K. He, R. Girshick, and P. Dollár. "Rethinking imagenet pre-training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4918–4927.

[89] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[90] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[91] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang. "Transreid: Transformer-based object re-identification". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 15013–15022.

[92] P. Helber, B. Bischke, A. Dengel, and D. Borth. "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12.7 (2019), pp. 2217–2226.

[93] O. Henaff. "Data-efficient image recognition with contrastive predictive coding". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4182–4192.

[94] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. "Data-efficient image recognition with contrastive predictive coding". In: *arXiv preprint arXiv:1905.09272* (2019).

[95] A. Hermans, L. Beyer, and B. Leibe. "In defense of the triplet loss for person re-identification". In: *arXiv preprint arXiv:1703.07737* (2017).

[96] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. "Learning deep representations

by mutual information estimation and maximization". In: *arXiv preprint arXiv:1808.06670* (2018).

[97] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[98] F. Hong, L. Pan, Z. Cai, and Z. Liu. "Versatile Multi-Modal Pre-Training for Human-Centric Perception". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16156–16166.

[99] F. Hong, L. Pan, Z. Cai, and Z. Liu. "Versatile Multi-Modal Pre-Training for Human-Centric Perception". In: *arXiv preprint arXiv:2203.13815* (2022).

[100] J. Hu, L. Shen, and G. Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[101] Q. Hu, X. Wang, W. Hu, and G.-J. Qi. "Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1074–1083.

[102] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. "Strategies for pre-training graph neural networks". In: *arXiv preprint arXiv:1905.12265* (2019).

[103] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun. "Gpt-gnn: Generative pre-training of graph neural networks". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1857–1867.

[104] L. Huang, S. You, M. Zheng, F. Wang, C. Qian, and T. Yamasaki. "Green hierarchical vision transformer for masked image modeling". In: *arXiv preprint arXiv:2205.13515* (2022).

[105] M. Huh, P. Agrawal, and A. A. Efros. "What makes ImageNet good for transfer learning?" In: *arXiv preprint arXiv:1608.08614* (2016).

[106] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[107] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (July 2014), pp. 1325–1339.

[108] A. Islam, C.-F. Chen, R. Panda, L. Karlinsky, R. Radke, and R. Feris. "A Broad Study on the Transferability of Visual Representations with Contrastive Learning". In: *arXiv preprint arXiv:2103.13517* (2021).

[109] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. "A survey on contrastive self-supervised learning". In: *Technologies* (2021).

[110] J. Jia, N. Gao, F. He, X. Chen, and K. Huang. "Learning Disentangled Attribute Representations for Robust Pedestrian Attribute Recognition". In: (2022).

[111] X. Jin, C. Lan, W. Zeng, G. Wei, and Z. Chen. "Semantics-aligned representation learning for person re-identification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11173–11180.

[112] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. "Supervised contrastive learning". In: *arXiv preprint arXiv:2004.11362* (2020).

[113] D. Kim, D. Cho, D. Yoo, and I. S. Kweon. "Learning image representations by completing damaged jigsaw puzzles". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 793–802.

[114] T. N. Kipf and M. Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[115] T. N. Kipf and M. Welling. "Variational graph auto-encoders". In: *arXiv preprint arXiv:1611.07308* (2016).

[116] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. "Skip-thought vectors". In: *Advances in neural information processing systems* 28 (2015).

[117] J. Klicpera, S. Weißenberger, and S. Günnemann. "Diffusion improves graph learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[118] P. A. Knight. "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* (2008).

[119] A. Kolesnikov, X. Zhai, and L. Beyer. "Revisiting self-supervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 1920–1929.

[120] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash. "Mean shift for self-supervised learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10326–10335.

[121] S. Kornblith, J. Shlens, and Q. V. Le. "Do better imagenet models transfer better?" In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2661–2671.

[122] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. "3d object representations for fine-grained categorization". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2013, pp. 554–561.

[123] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[124] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[125] A. Kumar and H. Daume III. "Learning task grouping and overlap in multi-task learning". In: *arXiv preprint arXiv:1206.6417* (2012).

[126] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266 (2015), pp. 1332–1338.

[127] G. Larsson, M. Maire, and G. Shakhnarovich. "Learning representations for automatic colorization". In: *European conference on computer vision*. Springer. 2016, pp. 577–593.

[128] D.-H. Lee et al. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. 2013, p. 896.

[129] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. "Gshard: Scaling giant models with conditional computation and automatic sharding". In: *arXiv preprint arXiv:2006.16668* (2020).

[130] D. Li, Z. Zhang, X. Chen, and K. Huang. "A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios". In: *IEEE transactions on image processing* 28.4 (2018), pp. 1575–1590.

[131] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, T. Sim, S. Yan, and J. Feng. "Multiple-human parsing in the wild". In: *arXiv preprint arXiv:1705.07206* (2017).

[132] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. Hoi. "Prototypical contrastive learning of unsupervised representations". In: *arXiv preprint arXiv:2005.04966* (2020).

[133] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. "Ai choreographer: Music conditioned 3d dance generation with aist++". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13401–13412.

[134] S. Li, D. Chen, B. Liu, N. Yu, and R. Zhao. "Memory-based neighbourhood embedding for visual recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6102–6111.

[135] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang, and Z. Li. "Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 16266–16275.

[136] W. Li, Z. Cao, J. Feng, J. Zhou, and J. Lu. "Label2Label: A Language Modeling Framework for Multi-Attribute Learning". In: *European Conference on Computer Vision*. Springer. 2022, pp. 562–579.

[137] W. Li, R. Zhao, T. Xiao, and X. Wang. "Deepreid: Deep filter pairing neural network for person re-identification". In: *Proceedings of*

the IEEE conference on computer vision and pattern recognition. 2014, pp. 152–159.

[138] W.-H. Li and H. Bilen. "Knowledge distillation for multi-task learning". In: *European Conference on Computer Vision*. Springer. 2020, pp. 163–176.

[139] X. Li, W. Wang, L. Yang, and J. Yang. "Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality". In: *arXiv preprint arXiv:2205.10063* (2022).

[140] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4804–4814.

[141] Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, and E. Zhou. "Tokenpose: Learning keypoint tokens for human pose estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11313–11322.

[142] Y. Li, C. Huang, C. C. Loy, and X. Tang. "Human attribute recognition by deep hierarchical contexts". In: *European conference on computer vision*. Springer. 2016, pp. 684–700.

[143] Z. Li, X. Shen, Y. Jiao, X. Pan, P. Zou, X. Meng, C. Yao, and J. Bu. "Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications". In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE. 2020, pp. 1677–1688.

[144] Z. Li, A. Ravichandran, C. Fowlkes, M. Polito, R. Bhotika, and S. Soatto. "Representation Consolidation for Training Expert Students". In: *arXiv preprint arXiv:2107.08039* (2021).

[145] X. Liang, K. Gong, X. Shen, and L. Lin. "Look into person: Joint body parsing & pose estimation network and a new benchmark". In: *IEEE transactions on pattern analysis and machine intelligence* 41.4 (2018), pp. 871–885.

[146] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan. "Human parsing with contextualized convolutional neural network". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1386–1394.

[147] V. Likhosherstov, A. Arnab, K. Choromanski, M. Lucic, Y. Tay, A. Weller, and M. Dehghani. "PolyViT: Co-training Vision Transformers on Images, Videos and Audio". In: *arXiv preprint arXiv:2111.12993* (2021).

[148] M. Lin, C. Li, X. Bu, M. Sun, C. Lin, J. Yan, W. Ouyang, and Z. Deng. "DETR for crowd pedestrian detection". In: *arXiv preprint arXiv:2012.06785* (2020).

[149] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[150] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu. "Negative margin matters: Understanding margin in few-shot classification". In: *European Conference on Computer Vision*. Springer. 2020, pp. 438–455.

[151] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu. "Conflict-averse gradient descent for multi-task learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18878–18890.

[152] F. Liu, C. Shen, and G. Lin. "Deep convolutional neural fields for depth estimation from a single image". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5162–5170.

[153] H. Liu, M. Long, J. Wang, and M. I. Jordan. "Towards understanding the transferability of deep representations". In: *arXiv preprint arXiv:1909.12031* (2019).

[154] K. Liu, O. Choi, J. Wang, and W. Hwang. "CDGNet: Class Distribution Guided Network for Human Parsing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4473–4482.

[155] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt. "Constrained graph variational autoencoders for molecule design". In: *Advances in neural information processing systems* 31 (2018).

[156] S. Liu, E. Johns, and A. J. Davison. "End-to-end multi-task learning with attention". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1871–1880.

[157] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. "Self-supervised learning: Generative or contrastive". In: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[158] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang. "Hydraplus-net: Attentive deep features for pedestrian analysis". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 350–359.

[159] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang. "Learning to propagate labels: Transductive propagation network for few-shot learning". In: *arXiv preprint arXiv:1805.10002* (2018).

[160] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[161] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1096–1104.

[162] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[163] C. C. Loy, D. Lin, W. Ouyang, Y. Xiong, S. Yang, Q. Huang, D. Zhou, W. Xia, Q. Li, P. Luo, et al. "Wider face and pedestrian challenge 2018: Methods and results". In: *arXiv preprint arXiv:1902.06854* (2019).

[164] Y. Lu, D. Allegra, M. Anthimopoulos, F. Stanco, G. M. Farinella, and S. Mougiakakou. "A multi-task learning approach for meal assessment". In: *Proceedings of the joint workshop on multimedia for cooking and eating activities and multimedia assisted dietary management*. 2018, pp. 46–52.

[165] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang. "Bag of tricks and a strong baseline for deep person re-identification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 0–0.

[166] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. "Pose guided person image generation". In: *Advances in neural information processing systems* 30 (2017).

[167] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. "Fine-grained visual classification of aircraft". In: *arXiv preprint arXiv:1306.5151* (2013).

[168] S. Meftah, N. Semmar, M.-A. Tahiri, Y. Tamaazousti, H. Essafi, and F. Sadat. "Multi-task supervised pretraining for neural domain adaptation". In: *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*. 2020, pp. 61–71.

[169] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. "Monocular 3d human pose estimation in the wild using improved cnn supervision". In: *2017 international conference on 3D vision (3DV)*. IEEE. 2017, pp. 506–516.

[170] G. A. Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[171] I. Misra and L. v. d. Maaten. "Self-supervised learning of pretext-invariant representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6707–6717.

[172] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. "Cross-stitch networks for multi-task learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3994–4003.

[173] S. P. Mohanty, D. P. Hughes, and M. Salathé. "Using deep learning for image-based plant disease detection". In: *Frontiers in plant science* 7 (2016), p. 1419.

[174] R. Mormont, P. Geurts, and R. Marée. "Comparison of deep transfer learning strategies for digital pathology". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 2262–2271.

[175] F. Murtagh and P. Contreras. "Algorithms for hierarchical clustering: an overview". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.1 (2012), pp. 86–97.

[176] Y. Netzer, T. Wang, A. Coates, A. Bissacco, and A. Y. Ng. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011* (2011), pp. 722–729.

[177] M.-E. Nilsback and A. Zisserman. "Automated flower classification over a large number of classes". In: *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE. 2008, pp. 722–729.

[178] M. Noroozi and P. Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles". In: *European conference on computer vision*. Springer. 2016, pp. 69–84.

[179] M. Noroozi, H. Pirsiavash, and P. Favaro. "Representation learning by learning to count". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5898–5906.

[180] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, et al. "DeepWeeds: A multiclass weed species image dataset for deep learning". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[181] A. v. d. Oord, Y. Li, and O. Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[182] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. "Cats and dogs". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3498–3505.

[183] P. Parmar and B. T. Morris. "What and how well you performed? a multitask learning approach to action quality assessment". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 304–313.

[184] J. Pennington, R. Socher, and C. D. Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[185] H. Pham, Z. Dai, Q. Xie, and Q. V. Le. "Meta pseudo labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11557–11568.

[186] G.-J. Qi, L. Zhang, F. Lin, and X. Wang. "Learning generalized transformation equivariant representations via autoencoding transformations". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[187] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[188] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. "Improving language understanding by generative pre-training". In: (2018).

[189] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[190] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 3347–3357.

[191] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8821–8831.

[192] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. "Learning multiple visual domains with residual adapters". In: *arXiv preprint arXiv:1705.08045* (2017).

[193] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka. "Contrastive learning with hard negative samples". In: *arXiv preprint arXiv:2010.04592* (2020).

[194] A. Rodriguez and A. Laio. "Clustering by fast search and find of density peaks". In: *science* 344.6191 (2014), pp. 1492–1496.

[195] J. T. Rolfe. "Discrete variational autoencoders". In: *arXiv preprint arXiv:1609.02200* (2016).

[196] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. "Imagenet

large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[197] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[198] M. B. Sariyildiz, Y. Kalantidis, D. Larlus, and K. Alahari. "Concept generalization in visual representation learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9629–9639.

[199] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[200] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[201] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun. "Crowdhuman: A benchmark for detecting human in a crowd". In: *arXiv preprint arXiv:1805.00123* (2018).

[202] N. Shazeer and M. Stern. "Adafactor: Adaptive learning rates with sublinear memory cost". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4596–4604.

[203] W. Shu, J. Wan, K. C. Tan, S. Kwong, and A. B. Chan. "Crowd Counting in the Frequency Domain". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19618–19627.

[204] X. Shu, X. Wang, X. Zang, S. Zhang, Y. Chen, G. Li, and Q. Tian. "Large-Scale Spatio-Temporal Person Re-identification: Algorithms and Benchmark". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[205] D. L. Silver and K. P. Bennett. "Guest editor's introduction: special issue on inductive transfer learning". In: *Machine Learning* 73.3 (2008), p. 215.

[206] D. L. Silver, Q. Yang, and L. Li. "Lifelong machine learning systems: Beyond learning algorithms". In: *2013 AAAI spring symposium series*. 2013.

[207] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[208] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *arXiv preprint arXiv:2001.07685* (2020).

[209] K. Soroush Abbasi, A. Tejankar, and H. Pirsiavash. "Mean Shift for Self-Supervised Learning". In: *International Conference on Computer Vision (ICCV)*. 2021.

[210] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese. "Which Tasks Should Be Learned Together in Multi-task Learning?" In: *arXiv preprint arXiv:1905.07553* (2019).

[211] A. Subramonian. "Motif-driven contrastive learning of graph representations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 18. 2021, pp. 15980–15981.

[212] P. Sudowe, H. Spitzer, and B. Leibe. "Person Attribute Recognition with a Jointly-trained Holistic CNN Model". In: *ICCV'15 ChaLearn Looking at People Workshop*. 2015.

[213] K. Sun, B. Xiao, D. Liu, and J. Wang. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5693–5703.

[214] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. "High-resolution representations for labeling pixels and regions". In: *arXiv preprint arXiv:1904.04514* (2019).

[215] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He. "Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism". In: *Proceedings of the Web Conference 2021*. 2021, pp. 2081–2091.

[216] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 480–496.

[217] D. Suryawanshi et al. "Image Recognition: Detection of nearly duplicate images". PhD thesis. California State University Channel Islands, 2018.

[218] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[219] M. Tan and Q. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6105–6114.

[220] S. Tang, D. Chen, L. Bai, K. Liu, Y. Ge, and W. Ouyang. "Mutual crf-gnn for few-shot learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2329–2339.

[221] A. Tao, K. Sapra, and B. Catanzaro. "Hierarchical multi-scale attention for semantic segmentation". In: *arXiv preprint arXiv:2005.10821* (2020).

[222] C. Tao, H. Wang, X. Zhu, J. Dong, S. Song, G. Huang, and J. Dai. "Exploring the Equivalence of Siamese Self-Supervised Learning via A Unified Gradient Framework". In: *arXiv preprint arXiv:2112.05141* (2021).

[223] D. Thanou, X. Dong, D. Kressner, and P. Frossard. "Learning heat diffusion graphs". In: *IEEE Transactions on Signal and Information Processing over Networks* 3.3 (2017), pp. 484–499.

[224] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. "YFCC100M: The new data in multimedia research". In: *Communications of the ACM* 59.2 (2016), pp. 64–73.

[225] Y. Tian, C. Suzuki, T. Clanuwat, M. Bober-Irizar, A. Lamb, and A. Kitamoto. "Kaokore: A pre-modern japanese art facial expression dataset". In: *arXiv preprint arXiv:2002.08595* (2020).

[226] Y. Tian, D. Krishnan, and P. Isola. "Contrastive multiview coding". In: *arXiv preprint arXiv:1906.05849* (2019).

[227] Y. Tian, D. Krishnan, and P. Isola. "Contrastive multiview coding". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer. 2020, pp. 776–794.

[228] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. "What makes for good views for contrastive learning?" In: *arXiv preprint arXiv:2005.10243* (2020).

[229] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. "Rethinking few-shot image classification: a good embedding is all you need?" In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 266–282.

[230] Y. Tian, X. Chen, and S. Ganguli. "Understanding self-supervised learning dynamics without contrastive pairs". In: *arXiv preprint arXiv:2102.06810* (2021).

[231] A. T. Tran, C. V. Nguyen, and T. Hassner. "Transferability and hardness of supervised classification tasks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1395–1405.

[232] H. Tu, C. Wang, and W. Zeng. "Voxelpose: Towards multi-camera 3d human pose estimation in wild environment". In: *European Conference on Computer Vision*. Springer. 2020, pp. 197–212.

[233] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. "The inaturalist species classification and detection dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8769–8778.

[234] S. Vandenhende, S. Georgoulis, B. De Brabandere, and L. Van Gool. "Branched multi-task networks: deciding what layers to share". In: *arXiv preprint arXiv:1904.02920* (2019).

[235] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[236] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[237] E. Vendrow, D. T. Le, and H. Rezatofighi. "JRDB-Pose: A Large-scale Dataset for Multi-Person Pose Estimation and Tracking". In: *arXiv preprint arXiv:2210.11940* (2022).

[238] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. "Recovering accurate 3d human pose in the wild using imus and a moving camera". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 601–617.

[239] C. Wang and Z. Liu. "Learning graph representation by aggregating subgraphs via mutual information maximization". In: *arXiv preprint arXiv:2103.13125* (2021).

[240] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang. "Mgae: Marginalized graph autoencoder for graph clustering". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 889–898.

[241] F. Wang, T. Kong, R. Zhang, H. Liu, and H. Li. "Self-Supervised Learning by Estimating Twin Class Distributions". In: *arXiv preprint arXiv:2110.07402* (2021).

[242] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. "Learning discriminative features with multiple granularities for person re-identification". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 274–282.

[243] H. Wang, S. Ge, Z. Lipton, and E. P. Xing. "Learning Robust Global Representations by Penalizing Local Predictive Power". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 10506–10518.

[244] H. Wang, H. Gouk, E. Frank, B. Pfahringer, and M. Mayo. "A Comparison of Machine Learning Methods for Cross-Domain Few-Shot Learning". In: *Australasian Joint Conference on Artificial Intelligence*. Springer. 2020, pp. 445–457.

[245] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions". In: *Proceedings of the*

IEEE/CVF international conference on computer vision. 2021, pp. 568–578.

[246] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, et al. "Image as a foreign language: Beit pretraining for all vision and vision-language tasks". In: *arXiv preprint arXiv:2208.10442* (2022).

[247] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2097–2106.

[248] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos. "Towards universal object detection by domain attention". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7289–7298.

[249] Y. Wang, X. Zhang, T. Yang, and J. Sun. "Anchor detr: Query design for transformer-based detector". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 3. 2022, pp. 2567–2575.

[250] Y. Wang, S. Tang, F. Zhu, L. Bai, R. Zhao, D. Qi, and W. Ouyang. "Revisiting the transferability of supervised pretraining: an mlp perspective". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9183–9193.

[251] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell. "Characterizing and avoiding negative transfer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11293–11302.

[252] Z. Wang, Y. Tsvetkov, O. Firat, and Y. Cao. "Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models". In: *arXiv preprint arXiv:2010.05874* (2020).

[253] C. Wei, H. Wang, W. Shen, and A. Yuille. "Co2: Consistent contrast for unsupervised visual representation learning". In: *arXiv preprint arXiv:2010.02217* (2020).

[254] L. Wei, L. Xie, J. He, J. Chang, X. Zhang, W. Zhou, H. Li, and Q. Tian. "Can Semantic Labels Assist Self-Supervised Visual Representation Learning?" In: *arXiv preprint arXiv:2011.08621* (2020).

[255] L. Wei, S. Zhang, W. Gao, and Q. Tian. "Person transfer gan to bridge domain gap for person re-identification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 79–88.

[256] H.-H. Wu and S. Wu. "Various proofs of the Cauchy-Schwarz inequality". In: *Octogon mathematical magazine* 17.1 (2009), pp. 221–229.

[257] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, et al. "Large-scale datasets for going deeper in image understanding". In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2019, pp. 1480–1485.

[258] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[259] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. "Unsupervised feature learning via non-parametric instance discrimination". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3733–3742.

[260] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[261] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. "Graph wavenet for deep spatial-temporal graph modeling". In: *arXiv preprint arXiv:1906.00121* (2019).

[262] B. Xiao, H. Wu, and Y. Wei. "Simple baselines for human pose estimation and tracking". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 466–481.

[263] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. "Sun database: Large-scale scene recognition from abbey to zoo". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3485–3492.

[264] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, Z. Li, and P. Luo. "Detco: Unsupervised contrastive learning for object detection". In: *arXiv preprint arXiv:2102.04803* (2021).

[265] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[266] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. "Unsupervised data augmentation for consistency training". In: *arXiv preprint arXiv:1904.12848* (2019).

[267] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu. "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16684–16693.

[268] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. "Simmim: A simple framework for masked image modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9653–9663.

[269] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[270] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. "ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation". In: *arXiv preprint arXiv:2204.12484* (2022).

[271] K. Yamaguchi, M. Hadi Kiapour, and T. L. Berg. "Paper doll parsing: Retrieving similar styles to parse clothing items". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3519–3526.

[272] Q. Yang, A. Wu, and W.-S. Zheng. "Person re-identification by contour sketch under moderate clothing change". In: *IEEE transactions on pattern analysis and machine intelligence* 43.6 (2019), pp. 2029–2046.

[273] S. Yang, Z. Quan, M. Nie, and W. Yang. "Transpose: Keypoint localization via transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11802–11812.

[274] Y. Yang, A. Eriguchi, A. Muzio, P. Tadepalli, S. Lee, and H. Hassan. "Improving Multilingual Translation by Representation and Gradient Regularization". In: *arXiv preprint arXiv:2109.04778* (2021).

[275] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems* 27 (2014), pp. 3320–3328.

[276] Y. You, I. Gitman, and B. Ginsburg. "Large batch training of convolutional networks". In: *arXiv preprint arXiv:1708.03888* (2017).

[277] Y. You, I. Gitman, and B. Ginsburg. "Scaling sgd batch size to 32k for imagenet training". In: *arXiv preprint arXiv:1708.03888* 6 (2017), p. 12.

[278] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. "Graph contrastive learning with augmentations". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5812–5823.

[279] Y. You, T. Chen, Z. Wang, and Y. Shen. "When does self-supervision help graph convolutional networks?" In: *international conference on machine learning*. PMLR. 2020, pp. 10871–10880.

[280] S. Yu, F. Zhu, D. Chen, R. Zhao, H. Chen, S. Tang, J. Zhu, and Y. Qiao. "Multiple Domain Experts Collaborative Learning: Multi-Source Domain Generalization For Person Re-Identification". In: *arXiv preprint arXiv:2105.12355* (2021).

[281] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. "Gradient surgery for multi-task learning". In: *arXiv preprint arXiv:2001.06782* (2020).

[282] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. "Gradient surgery for multi-task learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5824–5836.

[283] Y. Yuan, F. Rao, H. Lang, W. Lin, C. Zhang, X. Chen, and J. Wang. "HRFormer: High-Resolution Transformer for Dense Prediction. arXiv 2021". In: *arXiv preprint arXiv:2110.09408* ().

[284] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. "Taskonomy: Disentangling task transfer learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3712–3722.

[285] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. "Barlow twins: Self-supervised learning via redundancy reduction". In: *arXiv preprint arXiv:2103.03230* (2021).

[286] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. "S4l: Self-supervised semi-supervised learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1476–1485.

[287] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy. "Online deep clustering for unsupervised representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6688–6697.

[288] L. Zhang, G.-J. Qi, L. Wang, and J. Luo. "Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2547–2555.

[289] R. Zhang, P. Isola, and A. A. Efros. "Colorful image colorization". In: *European conference on computer vision*. Springer. 2016, pp. 649–666.

[290] S. Zhang, R. Benenson, and B. Schiele. "Citypersons: A diverse dataset for pedestrian detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3213–3221.

[291] S. Zhang, Y. Xie, J. Wan, H. Xia, S. Z. Li, and G. Guo. "Widerperson: A diverse dataset for dense pedestrian detection in the wild". In: *IEEE Transactions on Multimedia* 22.2 (2019), pp. 380–393.

[292] W. Zhang, M. Zhu, and K. G. Derpanis. "From actemes to action: A strongly-supervised representation for detailed action understanding". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2248–2255.

[293] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. "Single-image crowd counting via multi-column convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 589–597.

[294] Y. Zhang and Q. Yang. "An overview of multi-task learning". In: *National Science Review* 5.1 (2018), pp. 30–43.

[295] Y. Zhang, Z. Yin, J. Shao, and Z. Liu. "Benchmarking omni-vision representation through the lens of visual realms". In: *European Conference on Computer Vision*. Springer. 2022, pp. 594–611.

[296] Z. Zhang and M. Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels". In: *Advances in neural information processing systems* 31 (2018).

[297] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. "Spindle net: Person re-identification with human body region guided feature decomposition and fusion". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1077–1085.

[298] N. Zhao, Z. Wu, R. W. Lau, and S. Lin. "Distilling Localization for Self-Supervised Representation Learning". In: *arXiv preprint arXiv:2004.06638* (2020).

[299] N. Zhao, Z. Wu, R. W. Lau, and S. Lin. "What makes instance discrimination good for transfer learning?" In: *arXiv preprint arXiv:2006.06606* (2020).

[300] N. Zhao, Z. Wu, R. W. Lau, and S. Lin. "What makes instance discrimination good for transfer learning?" In: *arXiv preprint arXiv:2006.06606* (2020).

[301] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face recognition: A literature survey". In: *ACM computing surveys (CSUR)* 35.4 (2003), pp. 399–458.

[302] X. Zhao, S. Schulter, G. Sharma, Y.-H. Tsai, M. Chandraker, and Y. Wu. "Object detection with a unified label space from multiple datasets". In: *European Conference on Computer Vision*. Springer. 2020, pp. 178–193.

[303] A. Zheng, Y. Zhang, X. Zhang, X. Qi, and J. Sun. "Progressive End-to-End Object Detection in Crowded Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 857–866.

[304] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. "Scalable person re-identification: A benchmark". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1116–1124.

[305] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu. "Modanet: A large-scale street fashion dataset with polygon annotations". In:

Proceedings of the 26th ACM international conference on Multimedia. 2018, pp. 1670–1678.

[306] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[307] Y. Zheng, S. Tang, G. Teng, Y. Ge, K. Liu, J. Qin, D. Qi, and D. Chen. "Online pseudo label generation by hierarchical cluster dynamics for adaptive person re-identification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8371–8381.

[308] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz. "Joint discriminative and generative learning for person re-identification". In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2138–2147.

[309] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. "Places: An image database for deep scene understanding". In: *arXiv preprint arXiv:1610.02055* (2016).

[310] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. "Graph neural networks: A review of methods and applications". In: *AI open* 1 (2020), pp. 57–81.

[311] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. "ibot: Image bert pre-training with online tokenizer". In: *arXiv preprint arXiv:2111.07832* (2021).

[312] Q. Zhou, X. Liang, K. Gong, and L. Lin. "Adaptive temporal encoding network for video instance-level human parsing". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1527–1535.

[313] X. Zhou, V. Koltun, and P. Krähenbühl. "Simple multi-dataset detection". In: *arXiv preprint arXiv:2102.13086* (2021).

[314] J. Zhu, X. Zhu, W. Wang, X. Wang, H. Li, X. Wang, and J. Dai. "Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs". In: *arXiv preprint arXiv:2206.04674* (2022).

[315] K. Zhu, H. Guo, T. Yan, Y. Zhu, J. Wang, and M. Tang. "PASS: Part-Aware Self-Supervised Pre-Training for Person Re-Identification". In: *European Conference on Computer Vision*. Springer. 2022, pp. 198–214.

[316] Q. Zhu, C. Yang, Y. Xu, H. Wang, C. Zhang, and J. Han. "Transfer learning of graph neural networks with ego-graph information maximization". In: *Advances in Neural Information Processing Systems* 34 (2021).

[317] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang. "Deep graph contrastive representation learning". In: *arXiv preprint arXiv:2006.04131* (2020).

[318] C. Zhuang, A. L. Zhai, and D. Yamins. "Local aggregation for unsupervised learning of visual embeddings". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6002–6012.