Vision Transformer Advanced by Exploring Intrinsic Inductive Bias

YUFEI XU

SID: 490343009

THE UNIVERSITY OF
SYDNEY

Supervisor: Prof. Dacheng Tao
Auxiliary Supervisor: Dr. Jing Zhang

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

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

4 December 2023

To my parents.

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Yufei Xu
School of Computer Science
Faculty of Engineering
The University of Sydney
18 July 2023

Authorship Attribution Statement

This thesis was conducted at the University of Sydney, under the supervision of Prof. Dacheng Tao and Dr. Jing Zhang, between 2020 and 2023. The main results presented in this dissertation were first introduced in the following publications:

(1) **Yufei Xu**, Qiming Zhang, Jing Zhang, and Dacheng Tao. "ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias". Conference on Neural Information Processing Systems (NeruIPS) 2021. This paper is presented in Chapter 2. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

(2) **Yufei Xu**, Qiming Zhang, Jing Zhang, and Dacheng Tao. "RegionCL: Exploring Contrastive Region Pairs for Self-Supervised Representation Learning". European Conference on Computer Vision (ECCV) 2022. This paper is presented in Chapter 3. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

(3) **Yufei Xu**, Jing Zhang, Qiming Zhang, and Dacheng Tao. "ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation". Conference on Neural Information Processing Systems (NeruIPS) 2022. This paper is presented in Chapter 4. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

(4) Qiming Zhang, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "ViTAEv2: Vision Transformer Advanced by Exploring Inductive Bias for Image Recognition and Beyond". International Journal of Computer Vision (IJCV), 2022. This paper is presented in Chapter 2. I designed the research for the isotropic structure.

(5) **Yufei Xu**, Jing Zhang, Qiming Zhang, and Dacheng Tao. "ViTPose++: Vision Transformer Foundation Model for Generic Body Pose Estimation". IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2023. This paper is presented

in Chapter 4. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

Other publications I made contributions during my PhD course are listed as follows:

(1) **Yufei Xu**, Jing Zhang, Stephen J Maybank, and Dacheng Tao. "DUT: Learning Video Stabilization by Simply Watching Unstable Videos". IEEE Transactions on Image Processing (TIP) 2022.

(2) **Yufei Xu**, Jing Zhang, and Dacheng Tao. "Out-of-boundary view synthesis towards full-frame video stabilization". International Conference on Computer Vision (ICCV) 2021.

(3) Qiming Zhang, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "VSA: Learning Varied-Size Window Attention in Vision Transformers". European Conference on Computer Vision (ECCV) 2022.

(4) Zhe Chen, Jing Zhang, **Yufei Xu**, and Dacheng Tao. "Transformer-Based Context Condensation for Boosting Feature Pyramids in Object Detection". International Journal of Computer Vision (IJCV) 2023.

(5) Xu Zhang, Wen Wang, Zhe Chen, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "CLAMP: Prompt-Based Contrastive Learning for Connecting Language and Animal Pose". The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) 2023.

(6) Hang Yu, **Yufei Xu**, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. "AP-10K: A Benchmark for Animal Pose Estimation in the Wild". NeurIPS Datasets and Benchmarks Track 2021.

(7) Yuxiang Yang, Junjie Yang, **Yufei Xu**, Jing Zhang, Long Lan, and Dacheng Tao. "APT-36K: A Large-scale Benchmark for Animal Pose Estimation and Tracking". NeurIPS Datasets and Benchmarks Track 2022.

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.

Yufei Xu
School of Computer Science
Faculty of Engineering
The University of Sydney
01 August 2023

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

Dacheng Tao
School of Computer Science
Faculty of Engineering
The University of Sydney
01 August 2023

Acknowledgement

Throughout my Ph.D. studies, I received tremendous help from my supervisors, colleagues, friends, and family. This thesis will never be finished without their assistance.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dacheng Tao, for his meticulous guidance, insightful advice, and kind support throughout my PhD study. His passion for research and keen eye for the frontiers have been a constant source of inspiration for me. I have learned so much from him, not only about the technical aspects of research but also about the importance of asking critical questions and pursuing new ideas. In addition to his academic guidance, I have also benefited greatly from his attitude to research and taste. He has shown me that research should be more than just a means to an end. It should be a journey of discovery and a source of joy.

I am also particularly grateful to my auxiliary supervisor, Dr. Jing Zhang. His countless and kind help, accompany, and encouragement through the years have been a source of inspiration for me and inspire me to continue to pursue better research and face life positively. I am truly grateful for his mentorship and friendship. He has shown me a lot not only about the methods to do good research, but also the attitudes toward life. I will never forget the discussions and revisions throughout the night, as well as the joys brought by finding fantastic solutions.

I would like to thank my friends and colleagues: Mr. Qiming Zhang, Ms. Sihan Ma, Dr. Benteng Ma, Ms. Jizhizi Li, Dr. Sen Zhang, Ms. Haimei Zhao, Mr. Jinlong Fan, Mr. Haoyu He, Mr. Shaopeng Fu, Mr. Chen Chen, Mr. Qian Chen, Ms. Qi Zheng, Dr. Zhen Wang, Mr. Xu Zhang, Mr. Shiye Lei, Mr. Haibo Qiu, Mr. Zhi Hou, Dr. Shanshan Zhao, Dr. Fengxiang He, Dr. Youjian Zhang, Dr. Hao Guan, Dr. Xinqi Zhu, Ms. Yajing Kong, Mr. Yongcheng Jing, Mr. Zhihao Cheng, Dr. Shuo Yang, Dr. Chenwei Ding, Dr. Wei Zhai, Mr. Wen Wang, Mr. Haofei Xu, Ms Huayu Zhang, Dr. Meng Lan, Ms. Ning Wang, Mr. Di Wang, Ms. Yue He, Mr. Cheng Wen, Mr. Xikun Zhang, Mr. Kaining Zhang, Ms. Yutong Cao, Dr. Lianbo

Zhang, Dr. Zeyu Feng, Dr. Baosheng Yu, Dr. Liang Ding, Dr. Chaoyue Wang, Dr. Yu Cao, Dr. Liu Liu, Dr. Yuxiang Yang, Ms. Xiaofei Liu, for supports, company, and discussions during my Ph.D time.

Finally, I would like to express my gratefulness to my parents Mr. Weigan Xu and Ms. Yan Zhang. They have been continually encouraging me to fulfill my dreams and giving me endless love and support throughout my Ph.D time and throughout my life. They make me dare to try anything and keep optimistic about my life.

Abstract

The rapid expansion in the scale of training data has prompted a paradigm shift in the foundational architecture of vision models — moving away from convolutional neural networks (CNNs) towards transformers. In contrast to convolutions, which accentuate spatially adjacent regions, the attention mechanism within transformers can capture both short- and long-range dependencies. This grants them the capacity to embrace a more adaptable modeling approach when confronted with extensive datasets. This heightened flexibility empowers vision transformers with the potential for enhanced feature extraction prowess. However, it comes at a cost. In scenarios where training data is limited in scope, there arises a susceptibility to overfitting and the acquisition of unsuitable feature patterns. This data-hungry aspect curtails the potential utilization of vision transformers in domains where specialized data is scarce, thereby constraining their applications.

This thesis is dedicated to elevating the performance of vision transformers while preserving their adaptability in capturing both short- and long-range dependencies. With this goal in mind, our investigation delves into the factors contributing to the data-hungry nature of vision transformers, *e.g.*, the absence of inherent structural inductive bias. Notably, while convolution operations inherently encode spatial relationships, vision transformers lack this bias, necessitating learning from data.

To address this, our thesis proposes an innovative architectural approach that capitalizes on both the strengths of convolution and attention mechanisms. Central to our approach is the integration of a convolution block alongside the multi-head self-attention module within each transformer block. This fusion of features is subsequently channeled into feed-forward networks, promoting locality-based inductive bias. Additionally, our design incorporates an extra spatial pyramid reduction module, enhancing the extraction of multi-scale features to foster scale-invariance inductive bias. The accumulation of these meticulously devised modules culminates in the formulation of the ViTAE model.

Beyond architectural considerations, the efficacy of a vision transformer is profoundly influenced by its training methodology. Self-supervised learning (SSL) has become a popular training paradigm for vision transformers since they do not rely on expensive labeled data. To enhance the locality in SSL, we introduce RegionCL, *i.e.*, a novel region-level pretext task for contrastive learning by establishing region-level contrastive pairs. Concretely, this operation involves randomly cropping a region (termed the "paste view") from input images and interchanging these regions across different images to construct composite images alongside the retained regions (referred to as the "canvas view"). Unlike conventional approaches using entire images for positive or negative samples, our approach efficiently constructs contrastive pairs with a regional perspective. This is achieved by adhering to two fundamental criteria: (1) each view is positively paired with views augmented from the same image, and (2) each view is negatively paired with views augmented from distinct images.

To substantiate the efficacy of the design above enhancements, we undertake assessments across a spectrum of downstream vision tasks, including classification, detection, segmentation, and pose estimation. Specifically, in pose estimation tasks, we leverage the pre-trained vision transformers as the foundational architecture and embark on an exploration of the attributes of simplicity, scalability, flexibility, and transferability inherent in these meticulously trained backbone networks. Through judicious pre-training procedures, our extensive ViTPose-G model, boasting a ViTAE-G backbone with an impressive parameter count of 1 billion, emerges as a standout performer across diverse pose estimation datasets, including but not limited to MS COCO, MPII, and OCHuamn.

Publication List

(1) **Yufei Xu**, Qiming Zhang, Jing Zhang, and Dacheng Tao. "ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias". *Conference on Neural Information Processing Systems* (NeruIPS) 2021.

(2) **Yufei Xu**, Qiming Zhang, Jing Zhang, and Dacheng Tao. "RegionCL: Exploring Contrastive Region Pairs for Self-Supervised Representation Learning". *European Conference on Computer Vision* (ECCV) 2022.

(3) **Yufei Xu**, Jing Zhang, Qiming Zhang, and Dacheng Tao. "ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation". *Conference on Neural Information Processing Systems* (NeruIPS) 2022.

(4) **Yufei Xu**, Jing Zhang, Qiming Zhang, and Dacheng Tao. "ViTPose++: Vision Transformer Foundation Model for Generic Body Pose Estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI) 2023.

(5) **Yufei Xu**, Jing Zhang, Stephen J Maybank, and Dacheng Tao. "DUT: Learning Video Stabilization by Simply Watching Unstable Videos". *IEEE Transactions on Image Processing* (TIP) 2022.

(6) **Yufei Xu**, Jing Zhang, and Dacheng Tao. "Out-of-boundary view synthesis towards full-frame video stabilization". *International Conference on Computer Vision* (ICCV) 2021.

(7) Qiming Zhang, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "ViTAEv2: Vision Transformer Advanced by Exploring Inductive Bias for Image Recognition and Beyond". *International Journal of Computer Vision* (IJCV), 2022.

(8) Qiming Zhang, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "VSA: Learning Varied-Size Window Attention in Vision Transformers". *European Conference on Computer Vision* (ECCV) 2022.

(9) Zhe Chen, Jing Zhang, **Yufei Xu**, and Dacheng Tao. "Transformer-Based Context Condensation for Boosting Feature Pyramids in Object Detection". *International Journal of Computer Vision* (IJCV) 2023.

(10) Xu Zhang, Wen Wang, Zhe Chen, **Yufei Xu**, Jing Zhang, and Dacheng Tao. "CLAMP: Prompt-Based Contrastive Learning for Connecting Language and Animal Pose". *The IEEE / CVF Computer Vision and Pattern Recognition Conference* (CVPR) 2023.

(11) Hang Yu, **Yufei Xu**, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. "AP-10K: A Benchmark for Animal Pose Estimation in the Wild". *NeurIPS Datasets and Benchmarks Track* 2021.

(12) Yuxiang Yang, Junjie Yang, **Yufei Xu**, Jing Zhang, Long Lan, and Dacheng Tao. "APT-36K: A Large-scale Benchmark for Animal Pose Estimation and Tracking". *NeurIPS Datasets and Benchmarks Track* 2022.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Over recent years, the Transformer architecture [170] has risen to prominence in the realm of natural language processing, showcasing remarkable scalability and adeptness in feature extraction. This achievement has sparked an exploration into adapting the Transformer framework for vision-related tasks. Nevertheless, it's essential to note that Transformers are fundamentally designed to handle 1D inputs, such as words or sentences.

To align the inherently high-dimensional nature of vision inputs with the Transformer architecture, a pivotal innovation emerged by treating the image patches as words and the image itself as one sentence, formulating the vision transformer [45]. The central mechanism involves the introduction of a flatten operation, which serves to reshape the input data. This operation facilitates the transformation of visual information into distinct patches, effectively converting the high-dimensional input into a sequence of 1D representations. This approach, while elegantly simple in design, has proven to be remarkably effective. Vision transformers have exhibited superior performance, leveraging extensive datasets for pre-training.

Despite the superior performance of vision transformers, it's crucial to acknowledge that the attention operation must acquire the inductive bias that convolution operations naturally encode. This requirement, driven by the representation of images as 1D sequences of patches, gives rise to a prominent challenge termed the "data-hungry issue" within vision transformers. In essence, when presented with limited data, vision transformers are prone to overfitting the training dataset and can develop inappropriate inductive biases. This limitation significantly curtails the potential applications of vision transformers. Addressing this challenge has spurred the exploration of diverse strategies. For instance, DeiT [167] employs a distillation token to transfer inductive bias from a pre-trained CNN model, attempting to bridge the gap

between the biases encoded in convolutions and those learned through attention mechanisms. Subsequent endeavors delve into directly integrating locality inductive bias into vision transformers. The introduction of window-based attention, as witnessed in approaches like Swin Transformer [109], enforces a constraint on the maximum attention distance, thereby promoting locality. However, this approach might inadvertently disregard long-range dependencies due to the imposed limitations on window sizes or convolution kernel dimensions. Is there a mutually beneficial approach?

To navigate this challenge and infuse the vital locality inductive bias into vision transformers without undermining their prowess in modeling long-range dependencies, we propose an innovative parallel structure. Specifically, we advocate for the incorporation of a convolution module that operates in tandem with the multi-head self-attention (MHSA) within each transformer block. In this design, the features extracted from both the convolution module and the MHSA are skillfully merged before being channeled into the feed-forward network (FFN). This seemingly straightforward approach capitalizes on the unique strengths of both the convolution and attention mechanisms. The convolution module excels in capturing locality by honing in on nearby relationships, whereas the attention module excels at modeling distant dependencies. By synergistically combining the outputs from these parallel modules, the resultant network acquires the remarkable ability to concurrently capture both short- and long-range dependencies while seamlessly integrating the desired locality inductive bias.

Furthermore, the architecture might impact the upper limit of performance, whereas the training methods dictate the extent to which the network can realize its potential. To explore appropriate training methods for vision transformers, a prevalent approach involves distilling inductive bias from established CNN models [167] or leveraging large-scale data pre-trained vision transformer models. However, while effective, such strategies might inadvertently limit the performance potential of the designed models. Addressing overfitting concerns, DeiT [167] introduces more robust augmentation techniques to enrich training data. Nevertheless, scaling supervised training data remains challenging due to resource-intensive annotation processes.

Given the appeal of utilizing large-scale data without annotations, the avenue of self-supervised learning emerges as a pivotal direction. Here, contrastive pairs are formed from training data, with the InfoNCE loss [125] often employed to drive the network's ability to distinguish positive pairs (objects of the same category) from negative pairs (objects of differing categories). Earlier approaches [162] harnessed multi-view data or data from distinct modalities for constructing these pairs, yet gathering and organizing such paired data can be demanding. Thus, the emphasis shifts toward constructing contrastive pairs from single-modality data. With training lacking ground truth labels, data augmentation techniques become indispensable for creating these contrastive pairs [24, 26]. Positive pairs are crafted from different views of the same image, each subjected to diverse augmentation settings, while negative pairs are composed of any two views from separate images. Subsequently, the networks process these constructed pairs to extract features, with the features from the class token or global pooling operations often utilized to represent the processed view during loss calculation.

Nonetheless, a limitation arises: the conventional feature extraction process discards spatial information, potentially hampering the model's capacity to learn appropriate inductive bias. Consequently, pre-trained models may not be optimally suited for tasks requiring dense predictions, such as detection or segmentation. To mitigate this constraint, strategies like DenseCL [178] and PixPro [191] have been proposed. These methods introduce pixel-level contrastive pairs, enhancing the model's ability to extract local-sensitive features at the expense of global information. Notably, DetCo [186] aims to concurrently boost global and local information during learning by modifying multi-crop methodologies to encourage alignment between local-local, global-global, and local-global pairs, albeit with the trade-off of increased complexity in structure and design.

In pursuit of enhancing the model's capacity to learn locality without compromising its global modeling capabilities, this thesis introduces a pioneering pretext task called RegionCL via region-swapping. This task serves to facilitate the distinction between local and global pairs, ultimately enhancing the model's capacity to discern spatial relationships. Specifically, when provided with two images, a random region is selected and cropped from each original image. These extracted regions are denoted as "paste views," while the remaining segments are

termed "canvas views." Subsequently, a composite image is crafted by seamlessly merging a paste view from one image with the canvas view from another image. This composite image is then fed into the network for feature extraction. A crucial distinction arises in the treatment of these composite images during feature extraction. Instead of treating the composite image as a unified whole, a more nuanced approach is adopted. Specifically, masked pooling is employed to independently extract features from both the canvas and paste views. Consequently, contrastive pairs are constructed based on straightforward criteria: each view is considered positive when paired with views augmented from the same image, and negative when paired with views from any other image. This simple strategy engenders the creation of both local and global contrastive pairs. Local pairs are formed by pairing canvas views with paste views sourced from different images, fostering the modeling of local relationships. Similarly, global pairs emerge by pairing paste views with canvas views and their corresponding source images, facilitating the assimilation of the global context. With these abundant region-global contrastive pairs, RegionCL could help the models better capture local information during the training process, resulting in better performance in dense prediction tasks.

Beyond the realm of pre-training, it's imperative to assess the efficacy of pre-trained vision transformer models in downstream tasks, *e.g.*, classification, detection, segmentation, and pose estimation. This thesis will focus on the pose estimation task, which is one important topic in computer vision and a wide range of real-world applications [216, 101, 87]. This task demands meticulous annotation and the available dataset sizes can be limited, spanning various subjects like humans and diverse animals. To bridge the gap between vision transformers and pose estimation tasks, prior research endeavors [94, 201] have harnessed convolutional networks for initial feature extraction. Subsequently, transformers are employed as post-processors, tasked with modeling relationships between distinct keypoints. This strategic approach capitalizes on the innate inductive bias encoded within CNN models, facilitating the adaptation of transformers to the demands of pose estimation. However, it's crucial to acknowledge that this design entails certain limitations. Given that transformers solely process features derived from the CNN model, their performance might be hindered by the inherent limitations of the base CNN's feature extraction capabilities. Another line of

research [211] endeavors to leverage transformer-only models for both feature extraction and keypoint estimation. Yet, given the constrained nature of pose estimation data, this approach requires intricate architectural innovations to expedite the convergence of vision transformer models. For instance, parallel branch designs may be employed to concurrently address varying feature resolutions, along with hierarchical structures to effectively leverage different levels of information.

We argue that through the acquisition of pertinent inductive bias during the pre-training phase, the structure of models can be streamlined for downstream vision tasks including classification, detection, segmentation, and pose estimation tasks. In light of this, the thesis introduces the ViTPose model for the pose estimation task. Instead of employing convolutional neural networks (CNNs) for feature extraction or relying on intricate transformer architectures, the approach advocates for using plain vision transformers directly for both feature extraction and pose estimation. This is achieved by processing the target image through a pre-trained vision transformer, and then channeling the extracted features into straightforward decoders for keypoint localization. Given that the backbone model effectively captures distinctive features, the decoder's complexity can be notably reduced. In contrast to conventional setups involving stacks of deconvolution layers, a simpler decoder structure with just one projection layer can be employed. This attribute underscores the inherent simplicity of vision transformers when applied to pose estimation tasks. Furthermore, due to the absence of convoluted designs in vision transformer models, they lend themselves to easy scalability. The model can be expanded by stacking additional transformer blocks, rendering it both deeper and wider. Notably, larger models exhibit improved data efficiency when trained with the same data volume. This scalability feature signifies the potential of vision transformers to serve as foundational models for pose estimation tasks. Moreover, when armed with appropriate pre-training methodologies, plain vision transformers can adeptly acquire relevant inductive bias from task-specific data. This demonstrates the flexibility of transformers in adapting to the requirements of various vision tasks.

Overall, the thesis focuses on injecting inductive bias into vision transformers through architectural design (Chapter 2) and training methods (Chapter 3). The effectiveness of the

pre-trained models is demonstrated in classification (Chapter 2 & 3), detection (Chapter 3), segmentation (Chapter 3), and pose estimation tasks (Chapter 4), respectively. The largest model, *i.e.*, ViTAE-G with 1B parameters, obtains the best performance on MS COCO, OCHuman, and MPII pose estimation tasks (Chapter 4).

1.1 Contributions

The main contributions of the thesis are summarized as follows:

- In Chapter 2, we propose a ViTAE model [196] to introduce inductive bias into vision transformers via architectural design. ViTAE has several spatial pyramid reduction modules to downsample and embed the input image into tokens with rich multi-scale context by using multiple convolutions with different dilation rates. In this way, it acquires an intrinsic scale invariance IB and can learn robust feature representation for objects at various scales. Moreover, in each transformer layer, ViTAE has a convolution block in parallel to the multi-head self-attention module, whose features are fused and fed into the feed-forward network. Consequently, it has the intrinsic locality IB and can learn local features and global dependencies collaboratively. Experiments on ImageNet as well as downstream tasks prove the superiority of ViTAE over the baseline transformer and concurrent works. The code is publicly available at https://github.com/ViTAE-Transformer/ViTAE-Transformer.
- In Chapter 3, we introduce RegionCL [197] to emphasise locality IB during the models' learning. RegionCL constructs novel images by stitching paste and canvas views, which are the cropped and remaining regions generated using random cropping. Then, instead of taking the new images as a whole for positive or negative samples, contrastive pairs are efficiently constructed from the regional perspective based on the following simple criteria, i.e., each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. With minor modifications to popular SSL methods, RegionCL exploits those abundant pairs and helps the model distinguish the local features from both canvas

and paste views, therefore learning better visual representations. Experiments on ImageNet, MS COCO, and Cityscapes demonstrate that RegionCL improves MoCov2, DenseCL, and SimSiam by large margins for both CNNs and vision transformers. The code is publicly available at https://github.com/Annbless/RegionCL.

- In Chapter 4, we explore the application of vision transformers in pose estimation tasks, and a baseline model ViTPose [195] is proposed. It demonstrates that with proper inductive bias learned in the pre-training stage, the vision transformer model can adapt to the downstream tasks well and does not require specific designs. Specifically, ViTPose employs plain and non-hierarchical vision transformers as backbones to extract features for a given person instance and a lightweight decoder for pose estimation. It can be scaled up from 100M to 1B parameters by taking advantage of the scalable model capacity and high parallelism of transformers, setting a new Pareto front between throughput and performance. Besides, We also empirically demonstrate the flexibility and transferability of ViTPose. Experimental results show that our basic ViTPose model outperforms representative methods on the challenging MS COCO Keypoint Detection benchmark, while the largest model with ViTAE-G backbone sets a new state-of-the-art, i.e., 80.9 AP on the MS COCO test-dev set. The code and models are available at https://github.com/ViTAE-Transformer/ViTPose.

CHAPTER 2

ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias

---

2.1 Introduction

Transformers [170, 41, 82, 37, 107, 135] have shown a domination trend in NLP studies owing to their strong ability in modeling long-range dependencies by the self-attention mechanism [145, 177, 116]. Such success and good properties of transformers have inspired following many works that apply them in various computer vision tasks [45, 232, 228, 175, 20]. Among them, ViT [45] is the pioneering pure transformer model that embeds images into a sequence

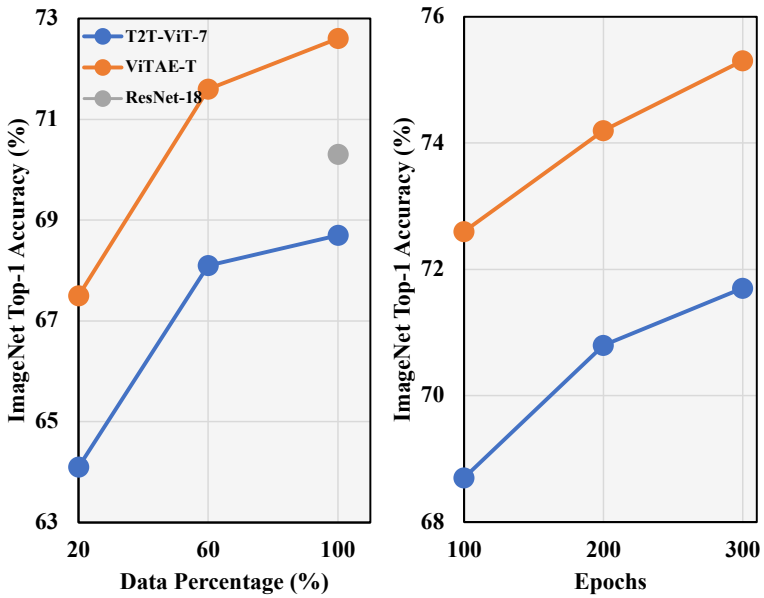


FIGURE 2.1. Comparison of data and training efficiency of T2T-ViT-7 and ViTAE-T on ImageNet.

of visual tokens and models the global dependencies among them with stacked transformer blocks. Although it achieves promising performance on image classification, it requires large-scale training data and a longer training schedule. One important reason is that ViT lacks intrinsic inductive bias (IB) in modeling local visual structures (*e.g.*, edges and corners) and dealing with objects at various scales like convolutions. Alternatively, ViT has to learn such IB implicitly from large-scale data. How to improve the vision transformer's performance with limited data? This thesis will focus on this problem from two perceptive, *i.e.*, structural design and training method. We will first demonstrate the structural improvements by taking inspiration from Convolution Neural Networks (CNNs).

Unlike vision transformers, CNNs naturally equip with the intrinsic IBs of scale-invariance and locality and still serve as prevalent backbones in vision tasks [61, 156, 136, 23, 226]. The success of CNNs inspires us to explore intrinsic IBs in vision transformers. We start by analyzing the above two IBs of CNNs, *i.e.*, locality and scale-invariance. Convolution that computes local correlation among neighbor pixels is good at extracting local features such as edges and corners. Consequently, CNNs can provide plentiful low-level features at the shallow layers [212], which are then aggregated into high-level features progressively by a bulk of sequential convolutions [68, 150, 157]. Moreover, CNNs have a hierarchy structure to extract multi-scale features at different layers [150, 80, 61]. Besides, intra-layer convolutions can also learn features at different scales by varying their kernel sizes and dilation rates [60, 156, 23, 98, 226]. Consequently, scale-invariant feature representation can be obtained via intra- or inter-layer feature fusion. Nevertheless, CNNs are not well suited to model long-range dependencies[1], which is the key advantage of transformers. An interesting question comes up: Can we improve vision transformers by leveraging the good properties of CNNs? Recently, DeiT [167] explores the idea of distilling knowledge from CNNs to transformers to facilitate training and improve the performance. However, it requires an off-the-shelf CNN model as the teacher and consumes extra training cost.

Different from DeiT, we explicitly introduce intrinsic IBs into vision transformers by re-designing the network structures in this paper. Current vision transformers always obtain

[1]Despite projection in transformer can be viewed as 1×1 convolution [29], the term of convolution here refers to those with larger kernels, *e.g.*, 3×3, which are widely used in typical CNNs to extract spatial features.

tokens with single-scale context [45, 209, 175, 187, 109, 151, 168] and learn to adapt to objects at different scales from data. For example, T2T-ViT [209] improves ViT by delicately generating tokens in a soft split manner. Specifically, it uses a series of Tokens-to-Token transformation layers to aggregate single-scale neighboring contextual information and progressively structurizes the image to tokens. Motivated by the success of CNNs in dealing with scale variance, we explore a similar design in transformers, *i.e.*, intra-layer convolutions with different receptive fields [156, 205], to embed multi-scale context into tokens. Such a design allows tokens to carry useful features of objects at various scales, thereby naturally having the intrinsic scale-invariance IB and explicitly facilitating transformers to learn scale-invariant features more efficiently from data. On the other hand, low-level local features are fundamental elements to generate high-level discriminative features. Although transformers can also learn such features at shallow layers from data, they are not skilled as convolutions by design. Recently, [198, 95, 51] stack convolutions and attention layers sequentially and demonstrate that locality is a reasonable compensation of global dependency. However, this serial structure ignores the global context during locality modeling (and vice versa). To avoid such a dilemma, we follow the "divide-and-conquer" idea and propose to model locality and long-range dependencies in parallel and then fuse the features to account for both. In this way, we empower transformers to learn local and long-range features within each block more effectively.

Technically, we propose a new **Vi**sion **T**ransformers **A**dvanced by **E**xploring Intrinsic Inductive Bias (***ViTAE***), which is a combination of two types of basic cells, *i.e.*, reduction cell (RC) and normal cell (NC). RCs are used to downsample and embed the input images into tokens with rich multi-scale context while NCs aim to jointly model locality and global dependencies in the token sequence. Moreover, these two types of cells share a simple basic structure, *i.e.*, paralleled attention module and convolutional layers followed by a feed-forward network (FFN). It is noteworthy that RC has an extra pyramid reduction module with atrous convolutions of different dilation rates to embed multi-scale context into tokens. Following the setting in [209], we stack three reduction cells to reduce the spatial resolution by $1/16$ and a series of NCs to learn discriminative features from data. ViTAE demonstrates its superiority over conventional representative vision transformers across various dimensions, encompassing

data efficiency, training efficiency (as depicted in Figure 2.1), classification accuracy, and generalization on subsequent tasks. Furthermore, we extend our analysis by scaling up the models to an expansive 600 million parameters. This scalability showcases that the inductive bias introduced by ViTAE continues to confer advantages to large-scale vision transformers.

Our contributions are threefold. **First**, we explore two types of intrinsic IB in transformers, *i.e.*, scale invariance and locality, and demonstrate the effectiveness of this idea in improving the feature learning ability of transformers. **Second**, we design a new transformer architecture named ViTAE based on two new reduction and normal cells to intrinsically incorporate the above two IBs. The proposed ViTAE embeds multi-scale context into tokens and learns both local and long-range features effectively. **Third**, ViTAE outperforms representative vision transformers regarding classification accuracy, data efficiency, training efficiency, and generalization on downstream tasks. ViTAE achieves 75.3% and 82.0% top-1 accuracy on ImageNet with 4.8M and 23.6M parameters, respectively.

2.2 Related Work

2.2.1 CNNs with intrinsic IB

CNNs have led to a series of breakthroughs in image classification [80, 212, 61, 223, 188] and downstream computer vision tasks. The convolution operations in CNNs extract local features from the neighbor pixels within the receptive field determined by the kernel size [85]. Following the intuition that local pixels are more likely to be correlated in images [84], CNNs have the intrinsic IB in modeling locality. In addition to the locality, another critical topic in visual tasks is scale-invariance, where multi-scale features are needed to represent the objects at different scales effectively [111, 204]. For example, to effectively learn features of large objects, a large receptive field is needed by either using large convolution kernels [204, 205] or a series of convolution layers in deeper architectures [61, 68, 150, 157]. To construct multi-scale feature representation, the classical idea is using image pyramid [23, 1, 124, 11, 81, 39], where features are hand-crafted or learned from a pyramid of images at different resolutions

respectively [97, 23, 120, 141, 74, 7]. Accordingly, features from the small scale image mainly encode the large objects while features from the large scale image respond more to small objects. In addition to the above inter-layer fusion way, another way is to aggregate multi-scale context by using multiple convolutions with different receptive fields within a single layer, *i.e.*, intra-layer fusion [226, 157, 156, 156, 158]. Either inter-layer fusion or intra-layer fusion empower CNNs an intrinsic IB in modeling scale-invariance. This paper introduces such an IB to vision transformers by following the intra-layer fusion idea and utilizing multiple convolutions with different dilation rates in the reduction cells to encode multi-scale context into each visual token.

2.2.2 Vision transformers with learned IB

ViT [45] is the pioneering work that applies a pure transformer to vision tasks and achieves promising results. However, since ViT lacks intrinsic inductive bias in modeling local visual structures, it indeed learns the IB from amounts of data implicitly. Following works along this direction are to simplify the model structures with fewer intrinsic IBs and directly learn them from large scale data [113, 164, 166, 53, 43, 46, 62] which have achieved promising results and been studied actively. Another direction is to leverage the intrinsic IB from CNNs to facilitate the training of vision transformers, *e.g.*, using less training data or shorter training schedules. For example, DeiT [167] proposes to distill knowledge from CNNs to transformers during training. However, it requires an off-the-shelf CNN model as a teacher, introducing extra computation costs during training. Recently, some works try to introduce the intrinsic IB of CNNs into vision transformers explicitly [55, 130, 51, 95, 38, 198, 181, 208, 19, 109, 33]. For example, [95, 51, 181] stack convolutions and attention layers sequentially, resulting in a serial structure and modeling the locality and global dependency accordingly. [175, 64] design sequential stage-wise structures while [109, 70] apply attention within local windows. However, these serial structure may ignore the global context during locality modeling (and vice versa). [193] establishes connections across different scales at the cost of heavy computation. Instead, we follow the "divide-and-conquer" idea and propose to model locality and global dependencies simultaneously via a parallel structure within each transformer

layer. Conformer [130], the most relevant concurrent work to us, employs a unit to explore inter-block interactions between parallel convolution and transformer blocks. In contrast, in ViTAE, the convolution and attention modules are designed to be complementary to each other within the transformer block. In addition, Conformer is not designed to have inherent scale invariance IB.

2.3 Methodology

2.3.1 Revisit vision transformer

We first give a brief review of vision transformer in this part. To adapt transformers to vision tasks, ViT [45] first splits an image $x \in R^{H\times W\times C}$ into tokens with a reduction ratio of p (*i.e.*, $x_t \in R^{((H\times W)/p^2)\times D}$), where H, W and C denote the height, width, and channel dimensions of the input image, $D = Cp^2$ denotes the token dimension. Then, an extra class token is concatenated to the visual tokens before adding position embeddings in an element-wise manner. The resulting tokens are fed into the following transformer layers. Each transformer layer is composed of two parts, *i.e.*, a multi-head self-attention module (MHSA) and a feed forward network (FFN).

MHSA Multi-head self-attention extends single-head self-attention (SHSA) by using different projection matrices for each head. Specifically, the input tokens x_t are first projected to queries (Q), keys (K) and values (V) using projection matrices, *i.e.*, $Q, K, V = x_tW_Q, x_tQ_K, x_tQ_V$, where $W_{Q/K/V} \in R^{D\times D}$ denotes the projection matrix for query, key, and value, respectively. Then, the self-attention operation is calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D}})V. \tag{2.1}$$

This SHSA module is repeated for h times to formulate the MHSA module, where h is the number of heads. The output features of the h heads are concatenated along the channel dimension and formulate the output of the MHSA module.

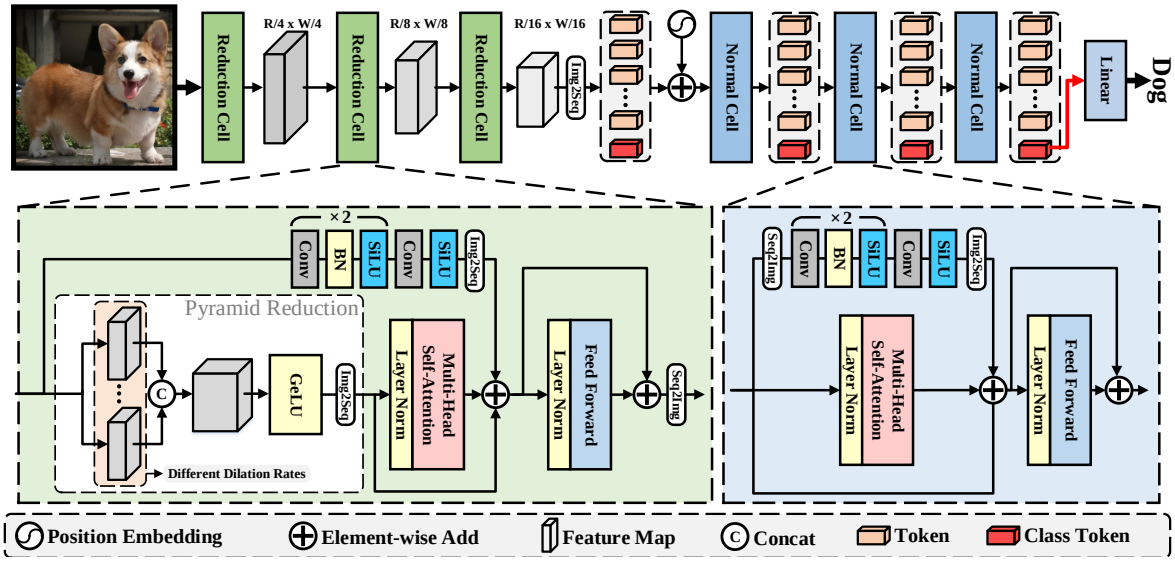


FIGURE 2.2. The structure of the proposed ViTAE. It is constructed by stacking three RCs and several NCs. Both types of cells share a simple basic structure, *i.e.*, an MHSA module and a parallel convolutional module followed by an FFN. In particular, RC has an extra pyramid reduction module using atrous convolutions with different dilation rates to embed multi-scale context into tokens.

FFN FFN is placed on top of the MHSA module and applied to each token identically and separately. It consists of two linear transformations with an activation function in between. Besides, a layer normalization [5] and a shortcut are added before and aside from the MHSA and FFN, respectively.

2.3.2 Overview architecture of ViTAE

ViTAE aims to introduce the intrinsic IB in CNNs to vision transformers. As shown in Figure 2.2, ViTAE is composed of two types of cells, *i.e.*, RCs and NCs. RCs are responsible for embedding multi-scale context and local information into tokens, and NCs are used to further model the locality and long-range dependencies in the tokens. Taken an image $x \in R^{H \times W \times C}$ as input, three RCs are used to gradually downsample x by $4\times$, $2\times$, and $2\times$, respectively. Thereby, the output tokens of the RCs are of size $[H/16, W/16, D]$ where D is the token dimension (64 in our experiments). The output tokens of RCs are then flattened as $R^{HW/256 \times D}$, concatenated with the class token, and added by the sinusoid position encoding.

Next, the tokens are fed into the following NCs, which keep the length of the tokens. Finally, the prediction probability is obtained using a linear classification layer on the class token from the last NC.

2.3.3 Reduction cell

Instead of directly splitting and flatten images into visual tokens based on a linear image patch embedding layer, we devise the reduction cell to embed multi-scale context and local information into visual tokens, which introduces the intrinsic scale-invariance and locality IBs from convolutions. Technically, RC has two parallel branches responsible for modeling locality and long-range dependency, respectively, followed by an FFN for feature transformation. We denote the input feature of the i_{th} RC as $f_i \in R^{H_i \times W_i \times D_i}$. The input of the first RC is the image x. In the global dependencies branch, f_i is firstly fed into a Pyramid Reduction Module (PRM) to extract multi-scale context, *i.e.*,

$$f_i^{ms} \triangleq PRM_i(f_i) = Cat([Conv_{ij}(f_i; s_{ij}, r_i) | s_{ij} \in \mathcal{S}_i, r_i \in \mathcal{R}]), \tag{2.2}$$

where $Conv_{ij}(\cdot)$ indicates the jth convolutional layer in the PRM ($PRM_i(\cdot)$). It uses a dilation rate s_{ij} from the predefined dilation rate set $\mathcal{S}_i$ corresponding to the ith RC. Note that we use stride convolution to reduce the spatial dimension of features by a ratio r_i from the predefined reduction ratio set $\mathcal{R}$. The conv features are concatenated along the channel dimension, *i.e.*, $f_i^{ms} \in R^{(W_i/p) \times (H_i/p) \times (|\mathcal{S}_i|D)}$, where $|\mathcal{S}_i|$ denotes the number of dilation rates in $\mathcal{S}_i$. f_i^{ms} is then processed by an MHSA module to model long-range dependencies, *i.e.*,

$$f_i^g = MHSA_i(Img2Seq(f_i^{ms})), \tag{2.3}$$

where $Img2Seq(\cdot)$ is a simple reshape operation to flatten the feature map to a 1D sequence. In this way, f_i^g embeds the multi-scale context in each token. In addition, we use a Parallel Convolutional Module (PCM) to embed local context within the tokens, which are fused with f_i^g as follows:

$$f_i^{lg} = f_i^g + PCM_i(f_i). \tag{2.4}$$

Here, $PCM_i(\cdot)$ represents the PCM, which is composed of three stacked convolution layers and an $Img2Seq(\cdot)$ operation. It is noteworthy that the parallel convolution branch has the same spatial downsampling ratio as the PRM by using stride convolutions. In this way, the token features can carry both local and multi-scale context, implying that RC acquires the locality IB and scale-invariance IB by design. The fused tokens are then processed by the FFN, reshaped back to feature maps, and fed into the following RC or NC, *i.e.*,

$$f_{i+1} = Seq2Img(FFN_i(f_i^{lg}) + f_i^{lg}), \tag{2.5}$$

where the $Seq2Img(\cdot)$ is a simple reshape operation to reshape a token sequence back to feature maps. $FFN_i(\cdot)$ represents the FFN in the ith RC. In our ViTAE, three RCs are stacked sequentially to gradually reduce the input image's spatial dimension by $4\times$, $2\times$, and $2\times$, respectively. The feature maps generated by the last RC are of a size of $[H/16, W/16, D]$, which are then flattened into visual tokens and fed into the following NCs.

2.3.4 Normal cell

As shown in the bottom right part of Figure 2.2, NCs share a similar structure with the reduction cell except for the absence of the PRM. Due to the relatively small ($\frac{1}{16}\times$) spatial size of feature maps after RCs, it is unnecessary to use PRM in NCs. Given f_3 from the third RC, we first concatenate it with the class token t_{cls}, and then add it to the positional encodings to get the input tokens t for the following NCs. Here we ignore the subscript for clarity since all NCs have an identical architecture but different learnable weights. t_{cls} is randomly initialized at the start of training and fixed during the inference. Similar to the RC, the tokens are fed into the MHSA module, *i.e.*, $t_g = MHSA(t)$. Meanwhile, they are reshaped to 2D feature maps and fed into the PCM, *i.e.*, $t_l = Img2Seq(PCM(Seq2Img(t)))$. Note that the class token is discarded in PCM because it has no spatial connections with other visual tokens. To further reduce the parameters in NCs, we use group convolutions in PCM. The features from MHSA and PCM are then fused via element-wise sum, *i.e.*, $t_{lg} = t_g + t_l$. Finally, t_{lg} are fed into the FFN to get the output features of NC, *i.e.*, $t_{nc} = FFN(t_{lg}) + t_{lg}$. Similar to

TABLE 2.1. Model details of two variants of ViTAE.

Model	Reduction Cell		Normal Cell			Params	Macs
	Dilation	Cells	Heads	Embed	Cells	(M)	(G)
ViTAE-T	$[1, 2, 3, 4] \downarrow$	3	4	256	7	4.8	1.5
ViTAE-S	$[1, 2, 3, 4] \downarrow$	3	6	384	14	23.6	5.6

ViT [45], we apply layer normalization to the class token generated by the last NC and feed it to the classification head to get the final classification result.

2.3.5 Model details

We use two variants of ViTAE in our experiments for a fair comparison of other models with similar model sizes. The details of them are summarized in Table 2.1. In the first RC, the default convolution kernel size is 7×7 with a stride of 4 and dilation rates of $\mathcal{S}_1 = [1, 2, 3, 4]$. In the following two RCs, the convolution kernel size is 3×3 with a stride of 2 and dilation rates of $\mathcal{S}_2 = [1, 2, 3]$ and $\mathcal{S}_3 = [1, 2]$, respectively. Since the spatial dimension of tokens decreases, there is no need to use large kernels and dilation rates. PCM in both RCs and NCs comprises three convolutional layers with a kernel size of 3×3.

2.4 Experiments

2.4.1 Implementation details

We train and test the proposed ViTAE model on the standard ImageNet [80] dataset, which contains about 1.3 million images and covers 1k classes. Unless explicitly stated, the image size during training is set to 224×224. We use the AdamW [110] optimizer with the cosine learning rate scheduler and uses the data augmentation strategy exactly the same as T2T [209] for a fair comparison, regarding the training strategies and the size of models. We use a batch size of 512 for training all our models and set the initial learning rate to be 5e-4. The results of our models can be found in Table 2.2, where all the models are trained for 300 epochs on 8 V100 GPUs. The models are built on PyTorch [128] and TIMM [180].

TABLE 2.2. Comparison of ViTAE and SOTA methods on the ImageNet validation set.

Type	Model	Params (M)	MACs (G)	Input Size	ImageNet Top-1	ImageNet Top-5	Real Top-1
CNN	ResNet-18 [61]	11.7	3.6	224	70.3	86.7	77.3
	ResNet-50 [61]	25.6	7.6	224	76.7	93.3	82.5
	ResNet-101 [61]	44.5	15.2	224	78.3	94.1	83.7
	ResNet-152 [61]	60.2	22.6	224	78.9	94.4	84.1
	EfficientNet-B0 [159]	5.3	0.8	224	77.1	93.3	83.5
	EfficientNet-B4 [159]	19.3	8.4	380	82.9	96.4	88.0
	MobileNetV1 [67]	4.3	0.6	224	72.3	-	-
	MobileNetV2(1.4) [143]	6.9	0.6	224	74.7	-	-
	RegNetY-600M [136]	6.1	1.2	224	75.5	-	-
	RegNetY-4GF [136]	20.6	8.0	224	80.0	-	86.4
	RegNetY-8GF [136]	39.2	16.0	224	81.7	-	87.4
Transformer	DeiT-T [167]	5.7	2.6	224	72.2	91.1	80.6
	DeiT-T⚗ [167]	5.7	2.6	224	74.5	91.9	82.1
	LocalViT-T [95]	5.9	2.6	224	74.8	92.6	-
	LocalViT-T2T [95]	4.3	2.4	224	72.5	-	-
	ConT-Ti [198]	5.8	1.6	224	74.9	-	-
	PiT-Ti [65]	4.9	1.4	224	73.0	-	-
	T2T-ViT-7 [209]	4.3	1.2	224	71.7	90.9	79.7
	ViTAE-T	4.8	1.5	224	75.3	92.7	82.9
	ViTAE-T ↑ 384	4.8	5.7	384	77.2	93.8	84.4
	CeiT-T [208]	6.4	2.4	224	76.4	93.4	83.6
	ConViT-Ti [38]	6.0	2.0	224	73.1	-	-
	CrossViT-Ti [19]	6.9	3.2	224	73.4	-	-
	ViTAE-6M	6.5	2.0	224	77.9	94.1	84.9
	PVT-T [175]	13.2	3.8	224	75.1	-	-
	LocalViT-PVT [95]	13.5	9.6	224	78.2	94.2	-
	ConViT-Ti+ [38]	10.0	4.0	224	76.7	-	-
	PiT-XS [65]	10.6	2.8	224	78.1	-	-
	ConT-M [198]	19.2	6.2	224	80.2	-	-
	ViTAE-13M	13.2	3.4	224	81.0	95.4	86.8
	DeiT-S [167]	22.1	9.8	224	79.9	95.0	85.7
	DeiT-S⚗ [167]	22.1	9.8	224	81.2	95.4	86.8
	PVT-S [175]	24.5	7.6	224	79.8	-	
	Conformer-Ti [130]	23.5	5.2	224	81.3	-	-
	Swin-T [109]	29.0	9.0	224	81.3	-	-
	CeiT-S [208]	24.2	9.0	224	82.0	95.9	87.3
	CvT-13 [181]	20.0	9.0	224	81.6	-	86.7
	ConViT-S [38]	27.0	10.8	224	81.3	-	-
	CrossViT-S [19]	26.7	11.2	224	81.0	-	-
	PiT-S [65]	23.5	4.8	224	80.9	-	-
	TNT-S [55]	23.8	10.4	224	81.3	95.6	-
	Twins-PCPVT-S[32]	24.1	7.4	224	81.2	-	-
	Twins-SVT-S [32]	24.0	5.6	224	81.7	-	-
	T2T-ViT-14 [209]	21.5	5.2	224	81.5	95.7	86.8
	ViTAE-S	23.6	5.6	224	82.0	95.9	87.0
	ViTAE-S ↑ 384	23.6	20.2	384	83.0	96.2	87.5

2.4.2 Comparison with the state-of-the-art

We compare our ViTAE with both CNN models and vision transformers with similar model sizes in Table 2.2. Both Top-1/5 accuracy and real Top-1 accuracy on the ImageNet validation set are reported. We categorize the methods into CNN models, vision transformers with learned IB, and vision transformers with introduced intrinsic IB. Compared with CNN models, our ViTAE-T achieves a 75.3% Top-1 accuracy, which is better than ResNet-18 with more parameters. The real Top-1 accuracy of the ViTAE model is 82.9%, which is comparable to ResNet-50 that has four more times of parameters than ours. Similarly, our ViTAE-S achieves 82.0% Top-1 accuracy with half of the parameters of ResNet-101 and ResNet-152, showing the superiority of learning both local and long-range features from specific structures with corresponding intrinsic IBs by design. Similar phenomena can also be observed when comparing ViTAE-T with MobileNetV1 [67] and MobileNetV2 [143], where ViTAE obtains better performance with fewer parameters. When compared with larger models which are searched according to NAS [159], our ViTAE-S achieves a similar performance when using 384×384 images as input, which further shows the potential of vision transformers with intrinsic IB.

In addition, among the transformers with learned IB, ViT is the first pure transformer model for visual recognition. DeiT shares the same structure with ViT but uses different data augmentation and training strategies to facilitate the learning of transformers. DeiT⚗ denotes using an off-the-shelf CNN model as the teacher model to train DeiT, which introduces the intrinsic IB from CNN to transformer implicitly in a knowledge distillation manner, showing better performance than the vanilla ViT on the ImageNet dataset. It is exciting to see that our ViTAE-T with fewer parameters even outperforms the distilled model DeiT⚗, demonstrating the efficacy of introducing intrinsic IBs in transformers by design. Besides, compared with other transformers with explicit intrinsic IB, our ViTAE with fewer parameters also achieves comparable or better performance. For instance, ViTAE-T achieves comparable performance with LocalVit-T but has 1M fewer parameters, demonstrating the superiority of the proposed RCs and NCs in introducing intrinsic IBs.

TABLE 2.3. Ablation Study of RCs and NCs in our ViTAE. "Pre" indicates the output features of PCM and MHSA are fused before FFN while "Post" indicates a late fusion strategy correspondingly. "BN" indicates whether PCM uses BN or not. "$[1,2,3,4]\downarrow$" denotes using smaller dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1,2,3,4]$, $\mathcal{S}_2 = [1,2,3]$, $\mathcal{S}_3 = [1,2]$.

Reduction Cell		**Normal Cell**			
Dilation ($\mathcal{S}_1 \sim \mathcal{S}_3$)	**PCM**	**Pre**	**Post**	**BN**	**Top-1**
×	×	×	×	×	68.7
×	×	✓	×	×	69.1
×	×	×	✓	×	69.0
×	×	×	✓	✓	68.8
×	×	✓	×	✓	69.9
$[1,2]\times 3$	×	×	×	×	69.5
$[1,2,3]\times 3$	×	×	×	×	69.9
$[1,2,3,4]\times 3$	×	×	×	×	69.2
$[1,2,3,4,5]\times 3$	×	×	×	×	68.9
$[1,2,3,4]\downarrow$	×	×	×	×	69.8
$[1,2,3,4]\downarrow$	✓	×	×	×	71.7
$[1,2,3,4]\downarrow$	✓	✓	×	✓	72.6

2.4.3 Ablation study

We use T2T-ViT [209] as our baseline model in the following ablation study of our ViTAE. As shown in Table 2.3, we investigate the hyper-parameter settings in RCs and NCs by isolating them separately. All the models are trained for 100 epochs on ImageNet and follow the same training setting and data augmentation strategy as described in Section 2.4.1.

We use ✓ and × to denote whether or not the corresponding module is enabled during the experiments. If all columns under the RC and NC are marked × as shown in the first row, the model becomes the standard T2T-ViT model. "Pre" indicates the output features of PCM and MHSA are fused before FFN while "Post" indicates a late fusion strategy correspondingly. "BN" indicates whether PCM uses BN after the convolutional layer or not. "$\times 3$" in the first column denotes that the dilation rate set is the same in the three RCs. "$[1,2,3,4]\downarrow$" denotes using lower dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1,2,3,4]$, $\mathcal{S}_2 = [1,2,3]$, $\mathcal{S}_3 = [1,2]$.

As can be seen, using a pre-fusion strategy and BN achieves the best 69.9% Top-1 accuracy among other settings. It is noteworthy that all the variants of NC outperform the vanilla T2T-ViT, implying the effectiveness of PCM, which introduces the intrinsic locality IB in

transformers. It can also be observed that BN plays an important role in improving the model's performance as it can help to alleviate the scale deviation between convolution's and attention's features. For the RC, we first investigate the impact of using different dilation rates in the PRM, as shown in the first column. As can be seen, using larger dilation rates (*e.g.*, 4 or 5) does not deliver better performance. We suspect that larger dilation rates may lead to plain features in the deeper RCs due to the smaller resolution of feature maps. To validate the hypothesis, we use smaller dilation rates in deeper RCs as denoted by $[1, 2, 3, 4] \downarrow$. As can be seen, it achieves comparable performance as $[1, 2, 3]\times$. However, compared with $[1, 2, 3, 4] \downarrow$, $[1, 2, 3]\times$ increases the amount of parameters from 4.35M to 4.6M. Therefore, we select $[1, 2, 3, 4] \downarrow$ as the default setting. In addition, after using PCM in the RC, it introduces the intrinsic locality IB, and the performance increases to 71.7% Top-1 accuracy. Finally, the combination of RCs and NCs achieves the best accuracy at 72.6%, demonstrating the complementarity between our RCs and NCs.

2.4.4 Data efficiency and training efficiency

To validate the effectiveness of the introduced intrinsic IBs in improving data efficiency and training efficiency, we compare our ViTAE with T2T-ViT at different training settings: (a) training them using 20%, 60%, and 100% ImageNet training set for equivalent 100 epochs on the full ImageNet training set, *e.g.*, we employ 5 times epochs when using 20% data for training compared with using 100% data; and (b) training them using the full ImageNet training set for 100, 200, and 300 epochs respectively. The results are shown in Figure 2.1. As can be seen, ViTAE consistently outperforms the T2T-ViT baseline by a large margin in terms of both data efficiency and training efficiency. For example, ViTAE using only 20% training data achieves comparable performance with T2T-ViT using all data. When 60% training data are used, ViTAE significantly outperforms T2T-ViT using all data by about an absolute 3% accuracy. It is also noteworthy that ViTAE trained for only 100 epochs has outperformed T2T-ViT trained for 300 epochs. After training ViTAE for 300 epochs, its performance is significantly boosted to 75.3% Top-1 accuracy. With the proposed RCs and NCs, the transformer layers in our ViTAE only need to focus on modeling long-range dependencies,

leaving the locality and multi-scale context modeling to its convolution counterparts, *i.e.*, PCM and PRM. Such a "divide-and-conquer" strategy facilitates the training of vision transformers, making it possible to learn more efficiently with less training data and fewer training epochs.

To further validate the data efficiency of ViTAE model, we train the ViTAE model from scratch on the smaller datasets, *i.e.*, Cifar10 and Cifar100. The results are summarized in Table 2.4. It can be viewed that with only $1/7$ number of epochs, the ViTAE-T model achieves better classification performance on Cifar10 dataset, with far fewer parameters (4.8M v.s. 86M), which further confirms ViTAE model's data efficiency.

TABLE 2.4. Results of training from scratch on Cifar10/100.

Model	Params (M)	Top-1 Acc	Epochs	Dataset
DeiT-B	86.0	97.5	7000	Cifar10
ViTAE-T	4.8	97.7	1000	Cifar10
ViTAE-T	4.8	85.0	1000	Cifar100

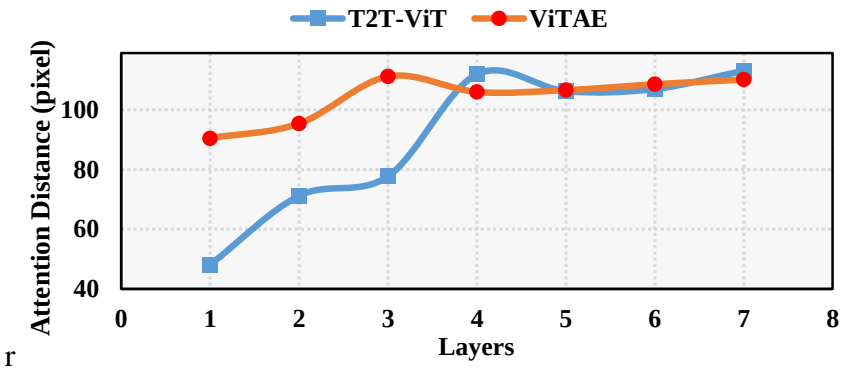
2.4.5 Generalization on downstream tasks

TABLE 2.5. Generalization of ViTAE and SOTA methods on different downstream tasks.

Model	**Params (M)**	**Cifar10**	**Cifar100**	**iNat19**	**Cars**	**Flowers**	**Pets**
Grafit ResNet-50 [169]	25.6	-	-	75.9	92.5	98.2	-
EfficientNet-B5 [159]	30	98.1	91.1	-	-	98.5	-
ViT-B/16 [45]	86.5	98.1	87.1	-	-	89.5	93.8
ViT-L/16 [45]	304.3	97.9	86.4	-	-	89.7	93.6
DeiT-B [167]	86.6	99.1	90.8	77.7	92.1	98.4	-
T2T-ViT-14 [209]	21.5	98.3	88.4	-	-	-	-
ViTAE-T	4.8	97.3	86.0	73.3	89.5	97.5	92.6
ViTAE-S	23.6	98.8	90.8	76.0	91.4	97.8	94.2

We further investigate the generalization of the proposed ViTAE models on downstream tasks by fine-tuning them on the training sets of several fine-grained classification tasks[2], including Flowers [122], Cars [77], Pets [127], and iNaturalist19. We also fine-tune the proposed ViTAE models on Cifar10 [79] and Cifar100 [79]. The results are shown in Table 2.5. It can be seen that ViTAE achieves SOTA performance on most of the datasets using comparable

[2]The performance of ViTAE on dense prediction tasks such as detection, segmentation, pose estimation, can be found in the supplementary material.



r

FIGURE 2.3. The average per-layer attention distance of T2T-ViT-7 and our ViTAE-T.

or fewer parameters. These results demonstrate that the good generalization ability of our ViTAE.

2.4.6 Visual inspection of ViTAE

To further analyze the property of our ViTAE, we first calculate the average attention distance of each layer in ViTAE-T and the baseline T2T-ViT-7 on the ImageNet test set, respectively. The results are shown in Figure 2.3. It can be observed that with the usage of PCM, which focuses on modeling locality, the transformer layers in the proposed NCs can better focus on modeling long-range dependencies, especially in shallow layers. In the deep layers, the average attention distances of ViTAE-T and T2T-ViT-7 are almost the same since modeling long-range dependencies is much more important. These results confirm the effectiveness of the adopted "divide-and-conquer" idea in the proposed ViTAE, *i.e.*, introducing the intrinsic locality IB from convolutions into vision transformers makes it possible that transformer layers only need to be responsible to long-range dependencies, since locality can be well modeled by convolutions in PCM.

Besides, we apply Grad-CAM [144] on the MHSA's output in the last NC to qualitatively inspect ViTAE. The visualization results are provided in Figure 2.4. Compared with the baseline T2T-ViT, our ViTAE covers the single or multiple targets in the images more precisely and attends less to the background. Moreover, ViTAE can better handle the scale variance issue as shown in Figure 2.4(b). Namely, it can precisely cover the birds no matter

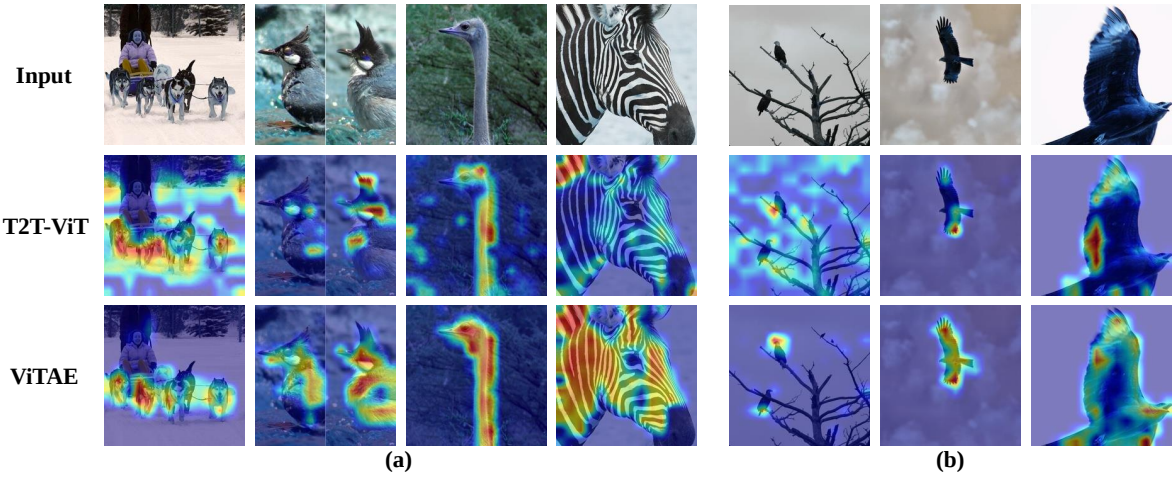


FIGURE 2.4. Visual inspection of T2T-ViT and ViTAE using Grad-CAM [144]. (a) Images containing multiple or single objects and the heatmaps. (b) Images containing the same class of objects at different scales and the heatmaps (Best viewed in color).

they are in small, middle, or large size. Such observations demonstrate that introducing the intrinsic IBs of locality and scale-invariance from convolutions to transformers helps ViTAE learn more discriminate features than the pure transformers.

2.4.7 Analysis of the scaled-up ViTAE models

2.4.7.1 Scaling up ViTAE via self-supervised learning

Besides stacking the proposed RCs and NCs to construct the ViTAE models, we also scale up ViTAE to evaluate the benefit of introducing the inductive bias in vision transformers with large model sizes. Specifically, we follow the setting in ViT [45] to scale up the proposed ViTAE model, *i.e.*, we embed the image into visual tokens and process them using stacked NCs to extract features. The stacking strategy is exactly the same as the strategy adopted in ViT [45], where we use 12 NCs with 768 embedding dimensions to construct the ViTAE base model (*i.e.*, ViTAE-B with 89M parameters), 24 NCs with 1,024 embedding dimensions to construct the ViTAE large model (*i.e.*, ViTAE-L with 311M parameters), and 36 normal cells with 1,248 embedding dimension to construct the ViTAE huge model (*i.e.*, ViTAE-H with 644M parameters). The normal cells are stacked sequentially. However, the scaled-up models

are easy to overfit if only trained using the ImageNet-1K dataset under a fully supervised training setting. Self-supervised learning [56], on the contrary, can eliminate this issue and facilitate the training of scaled-up models.

However, as the encoder only processes the visual tokens, *i.e.*, the remained tokens after removal, the built-in locality property of images have been broken among the visual tokens. To adapt the proposed ViTAE model to the self-supervised task, we simply use convolution with kernel size 1×1 instead of 3×3 to formulate the ViTAE model for pretraining. This simple modification helps us to preserve a similar architecture between the network's pretraining and finetuning stages and helps the convolution branch to learn a meaningful initialization. After the pretraining stage, we convert the kernels of the convolutions from 1×1 to 3×3 by zero-padding to recover the complete ViTAE models, which are further finetuned on the ImageNet-1k training data for 50 epochs.

2.4.7.2 Image classification performance

We evaluate the performance of the scaled-up models on the ImageNet dataset. The scaled-up models are pre-trained for 1600 epochs using MAE [56], taking images from the ImageNet-1K training set. Then, the models are finetuned for 100 (the base model) or 50 (the large and huge model) epochs using the labeled data from the ImageNet-1K training set. It should be noted that the original MAE is trained on the TPU machines with Tensorflow, while our implementation adopts PyTorch as the framework and uses NVIDIA GPU for the training. This implementation difference may cause a slight performance difference in the models' classification accuracy. We compare our methods with T2TViT [209], CvT [208], Swin [109], SwinV2 [108], and ViT [45] with either supervised learning or self-supervised learning like MAE [56], MaskFeat [179], and SimMIM [192] The results are summarized in Table 2.6. It demonstrates that the proposed ViTAE-B model with the introduced inductive bias outperforms the baseline ViT-B model by 0.4 Top-1 classification accuracy. For the ViTAE-L model with 300M parameters, the inductive bias still brings about 0.3 performance gains. After using ImageNet-22K labeled data for finetuning, the classification accuracy on the ImageNet-1K validation set further increases by about 1%. These results show that

TABLE 2.6. The performance of scaled-up ViTAE models on the ImageNet1K dataset. * denotes the results that we re-implement on GPU with PyTorch framework. † indicates that ImageNet22K is used to further finetune the models with 224×224 resolution for 90 epochs. 'Sup' is short for supervised learning.

	#Params	Test size	Method	ImageNet Top-1	Real Top-1
T2TViT-24	65 M	224	Sup	82.3	87.2
ViT-B*	88 M	224	MAE	83.4	89.1
ViTAE-B	89 M	224	MAE	83.8	89.4
ViTAE-B†	89 M	224	MAE	84.8	89.9
Swin-L†	197 M	384	Sup	87.3	90.0
SwinV2-L†	197 M	384	Sup	87.7	-
CoAtNet-4†	275 M	384	Sup	87.9	-
CvT-W24†	277 M	384	Sup	87.7	-
ViT-L*	304 M	224	MAE	85.5	90.1
ViT-L	304 M	224	MaskFeat	85.7	-
ViTAE-L	311 M	224	MAE	86.0	90.3
ViTAE-L†	311 M	224	MAE	87.5	90.8
ViTAE-L†	311 M	384	MAE	88.3	91.1
SwinV2-H	658 M	224	SimMIM	85.7	-
SwinV2-H	658 M	512	SimMIM	87.1	-
ViTAE-H	644 M	224	MAE	86.9	90.6
ViTAE-H	644 M	512	MAE	87.8	91.2
ViTAE-H†	644 M	224	MAE	88.0	90.7
ViTAE-H†	644 M	448	MAE	88.5	90.8

the benefit of introducing inductive bias in vision transformers is scalable to large models and datasets. Notably, our ViTAE-H, trained with only the ImageNet-1K dataset, obtains a classification accuracy of 91.2 on the ImageNet Real dataset [8], which is the highest accuracy we are aware of. It outperforms other methods trained with additional private data, such as EfficientNet [131] and ViT-G [213], where the former obtains 91.1 accuracy using the JFT300M dataset and the latter obtains 90.8 accuracy using the JFT3B dataset.

2.4.7.3 Few-shot learning performance

We further evaluate the data efficiency of scaled-up models by using different percentages of data to finetune the pre-trained models. We use 1%, 10%, and 100% data from the ImageNet-1k training set to finetune the self-supervised pre-trained ViTAE models with different amounts of parameters. We ensure that each model sees the same amount of the

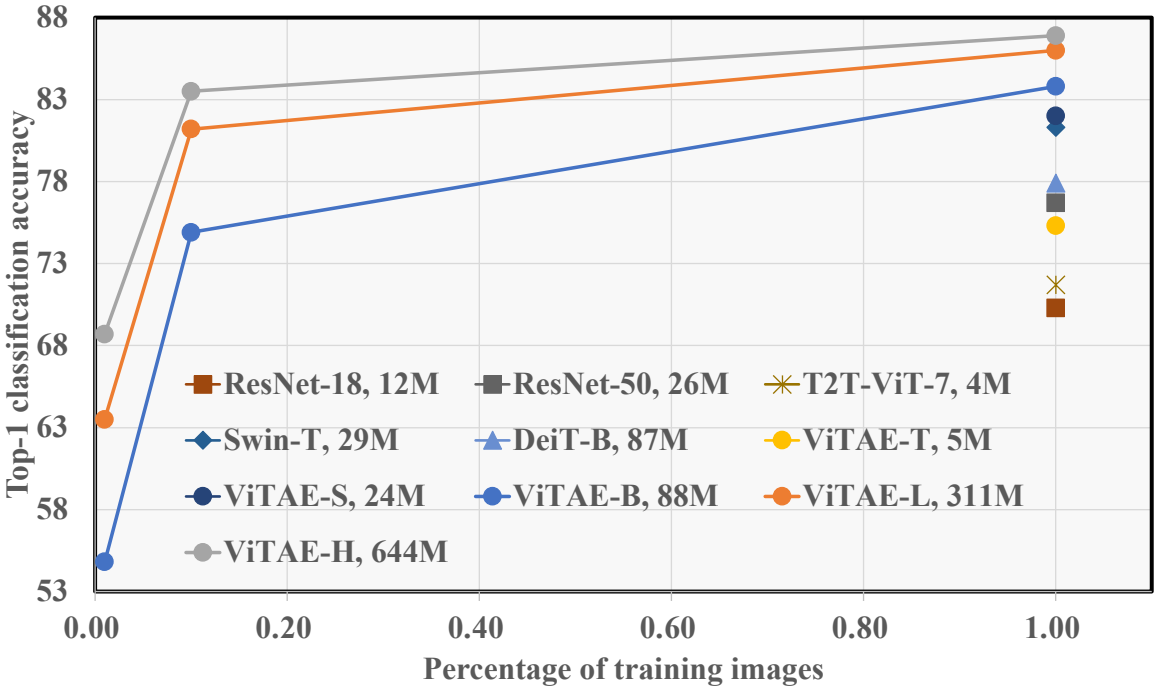


FIGURE 2.5. The few-shot image classification results of scaled-up ViTAE models. With only 10% data for finetuning, the scaled-up ViTAE models obtain comparable or even better performance compared with the small CNNs and transformer models with 100% data for training.

training images under different data settings, *i.e.*, we train the ViTAE-B model for 10,000 epochs using 1% training data, for 1,000 epochs using 10% training data, and for 100 epochs using 1% training data. Similarly, the ViTAE-L and ViTAE-H models are trained for 5,000 epochs using 1% training data and 500 epochs using 10% training data. The smaller models, *i.e.*, with less than 20M parameters, are trained from scratch using 100% ImageNet-1k training data for 300 epochs. As shown in Figure 2.5, the models with more parameters are more data-efficient than those with fewer parameters. For example, the ViTAE-H model with 644M parameters trained with 10% data outperforms the small model with 13.2 parameters trained with 100% data, *i.e.*, 82.4 v.s. 81.0. Such observations mirror the findings in previous studies, including both image classification and language modeling.

TABLE 2.7. The influence of the convolutional kernel size in the scaled up ViTAE models during pretraining.

	Kernel Size	#Params	Accuracy
ViT-L	0	304 M	84.1
ViTAE-L	0	311 M	84.1
ViTAE-L	3	311 M	84.4
ViTAE-L	1	311 M	84.6

2.4.7.4 Ablation study of the convolutional kernel size in the scaled up ViTAE models

We conduct experiments to investigate the influence of the convolutional kernel size in the scaled-up ViTAE models during pretraining. The results are presented in Table 2.7. The kernel size 0 represents that we do not use the convolution branch in ViTAE-L during pretraining, which degenerates to the original ViT-L model. Then, we add the convolution branches during finetuning and initialize the convolutional kernel weight as 0. If we use 1×1 kernels in ViTAE-L during pretraining, we pad them to 3×3 with zero padding during finetuning. We pretrain the models for 400 epochs and further finetune them for 50 epochs on the ImageNet-1K training set. As can be seen, using no convolution branch during pretraining leads to no improvement over the baseline ViT-L since those convolutional kernel weights in ViTAE-L during finetuning are zero-initialized. Directly pretraining and finetuning ViTAE-L with 3×3 convolutional kernels in the convolution branches leads to slightly better performance over the baseline ViT-L, but it is inferior to the proposed setting, *i.e.*, using 1×1 convolutional kernels in the convolution branches during pretraining ViTAE-L while zero-padding them to 3×3 during finetuning. We argue the reason is that most tokens (75%) during pretraining are randomly removed, and the remaining ones have lost spatial information. Therefore, using 3×3 kernels may lead to overfitting while 1×1 convolutions pay little attention to spatial structures and could learn better feature representation.

2.5 Limitation and discussion

In this paper, we explore two types of IBs and incorporate them into transformers through the proposed reduction and normal cells. With the collaboration of these two cells, our ViTAE

model achieves impressive performance on the ImageNet with fast convergence and high data efficiency. Nevertheless, due to computational resource constraints, we have not scaled the ViTAE model and train it on large-size dataset, *e.g.*, ImageNet-21K [80] and JFT-300M [66]. Although it remains unclear by now, we are optimistic about its scale property from the following preliminary evidence. As illustrated in Figure 2.2, our ViTAE model can be viewed as an intra-cell ensemble of complementary transformer layers and convolution layers owing to the skip connection and parallel structure. According to the attention distance analysis shown in Figure 2.3, the ensemble nature enables the transformer layers and convolution layers to focus on what they are good at, *i.e.*, modeling long-range dependencies and locality. Therefore, ViTAE is very likely to learn better feature representation from large-scale data. Besides, we only study two typical IBs in this paper. More kinds of IBs such as constituting viewpoint invariance [142] can be explored in the future study.

2.6 Conclusion

In this paper, we re-design the transformer block by proposing two basic cells (reduction cells and normal cells) to incorporate two types of intrinsic inductive bias (IB) into transformers, *i.e.*, locality and scale-invariance, resulting in a simple yet effective vision transformer architecture named ViTAE. Extensive experiments show that ViTAE outperforms representative vision transformers in various respects including classification accuracy, data efficiency, training efficiency, and generalization ability on downstream tasks. We plan to scale ViTAE to the large or huge model size and train it on large-size datasets in the future study. In addition, other kinds of IBs will also be investigated. We hope that this study will provide valuable insights to the following studies of introducing intrinsic IB into vision transformers and understanding the impact of intrinsic and learned IBs.

TABLE 2.8. ViTAE with different PE.

	Sinusoid	No	Learnable
Top-1	75.3	75.3	75.1

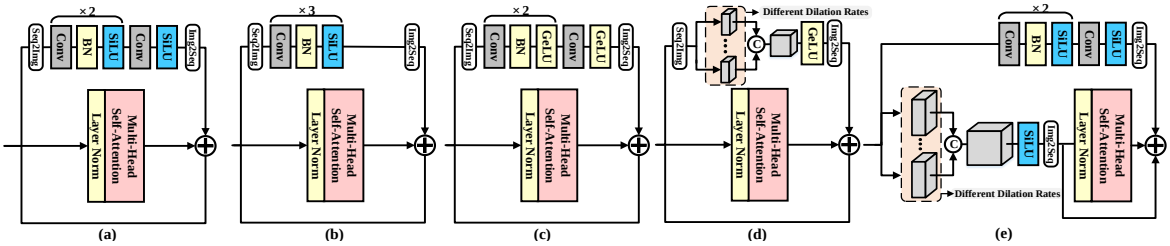


FIGURE 2.6. Different structures used in the ablation study. (a) Origin ViTAE structure. (b) Adding another BN in the PCM. (c) Replacing SiLU with GeLU in the PCM. (d) Replacing PCM with PRM. (e) Replacing GeLU with SiLU in the PRM.

TABLE 2.9. More ablation studies. (a), (b), (c), (d), (e) correspond to different structures in Figure 2.6.

	(a)	(b)	(c)	(d)	(e)
Top-1	72.6	69.6	72.3	71.9	72.4

2.7 Appendix

2.7.1 Analysis of position embedding

As CNN can encode position information with padding [33], we further disable the position embedding and train the ViTAE model without position embedding for 300 epochs. The results are summarized in Table 2.8. It can be seen that removing position embedding (PE) in the ViTAE model does not downgrade its performance. Such phenomena show that the PCM and PRM modules utilized in the ViTAE can aid the model in making sense of location information.

2.7.2 More Ablation Studies

To further analyze the structure of our ViTAE model, we conduct more ablation studies related to the proposed reduction cells and normal cells. As shown in Table 2.9 and Figure 2.6, we first

add another batch normalization [72] in the PCM module to make the PCM module has three exactly same convolution layers. However, such a structure downgrades the performance by 3%. As there is layer normalization [5] before the input to the FFN module, the combination of batch normalization and layer normalization may conflict with each other, resulting in performance degradation. Another design choice we tried is replacing the PCM with PRM modules (Figure 2.6 (d)) to introduce both scale-invariance and locality in the parallel branch. However, such a design shows a small drop in the performance, indicating that not only introducing inductive bias is important in transformers, but also the way in which these inductive biases are introduced also matters. What's more, we test the activation function used in the PCM and PRM modules. By default, we adopt SiLU in PCM, following the design choice in pioneering CNN networks and GeLu in PRM following previous transformers' design. By replacing the SiLU with GeLU (Figure 2.6 (c)), the performance drops a little. Similar phenomena can be observed when all PCM and PRM modules adopt SiLU as activation functions (Figure 2.6 (e)).

2.7.3 More visual results.

We also provide more visual inspection results of ViTAE using Grad-CAM [144] in Figure 2.7 2.8 2.9 2.10 and compare it with T2T-ViT [209] in Figure 2.11. It can be seen that our ViTAE can cover the targets more precisely and compress the noise introduced by the complex background. Such phenomena confirm that with the introduced inductive bias, the ViTAE model can better adapt to targets in different situations and thus achieves better performance on the vision tasks.

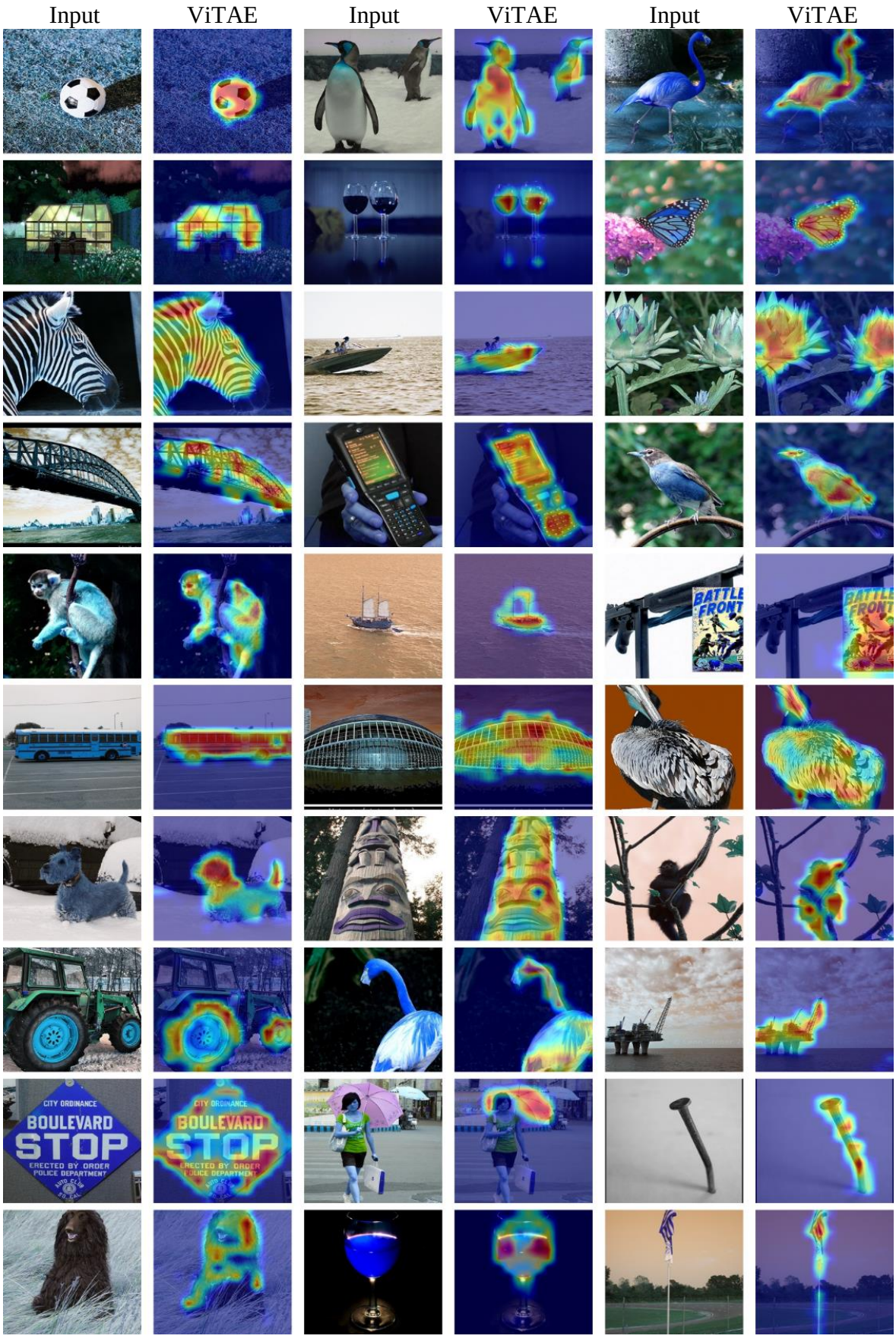


FIGURE 2.7. More visual results of ViTAE.

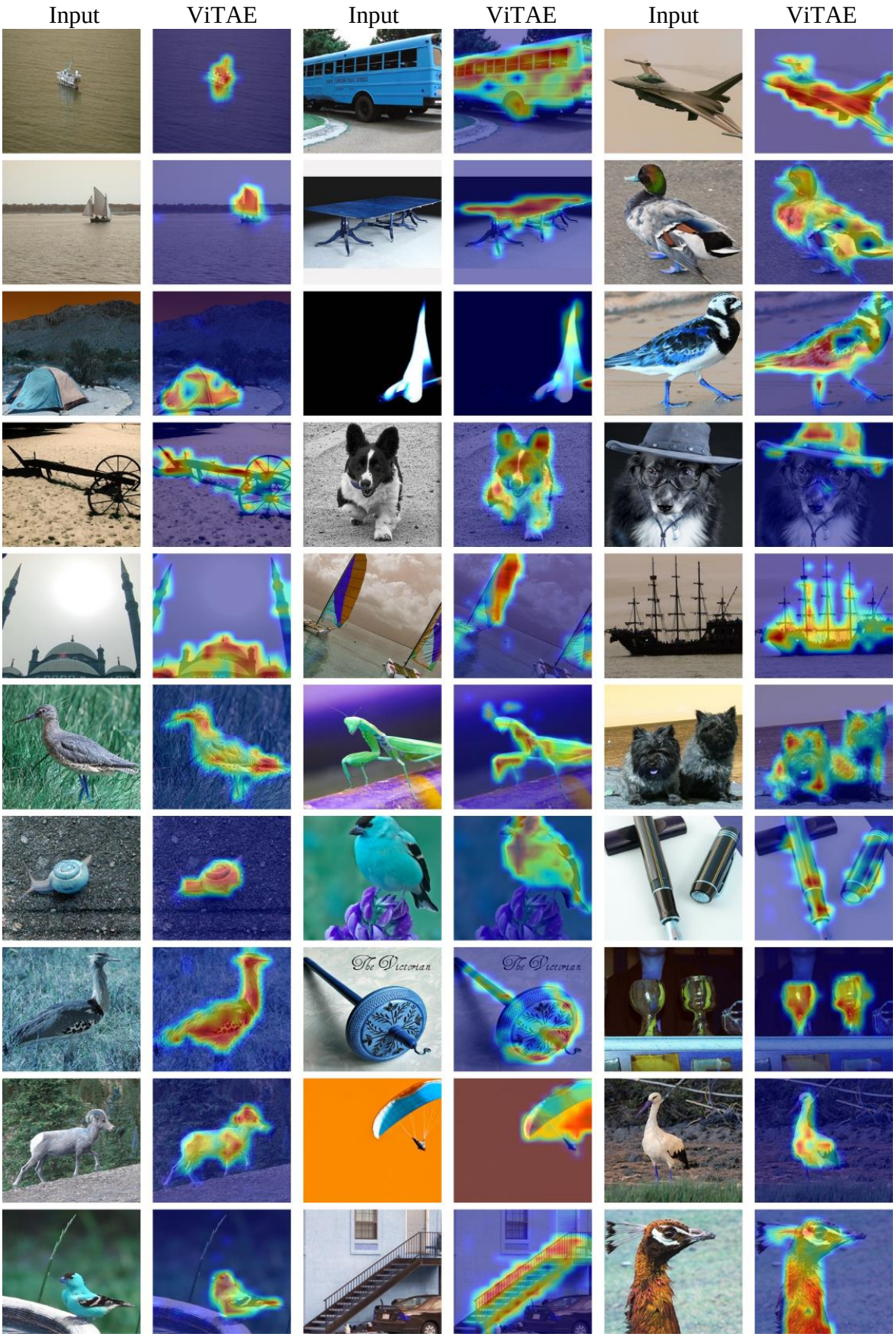


FIGURE 2.8. More visual results of ViTAE.

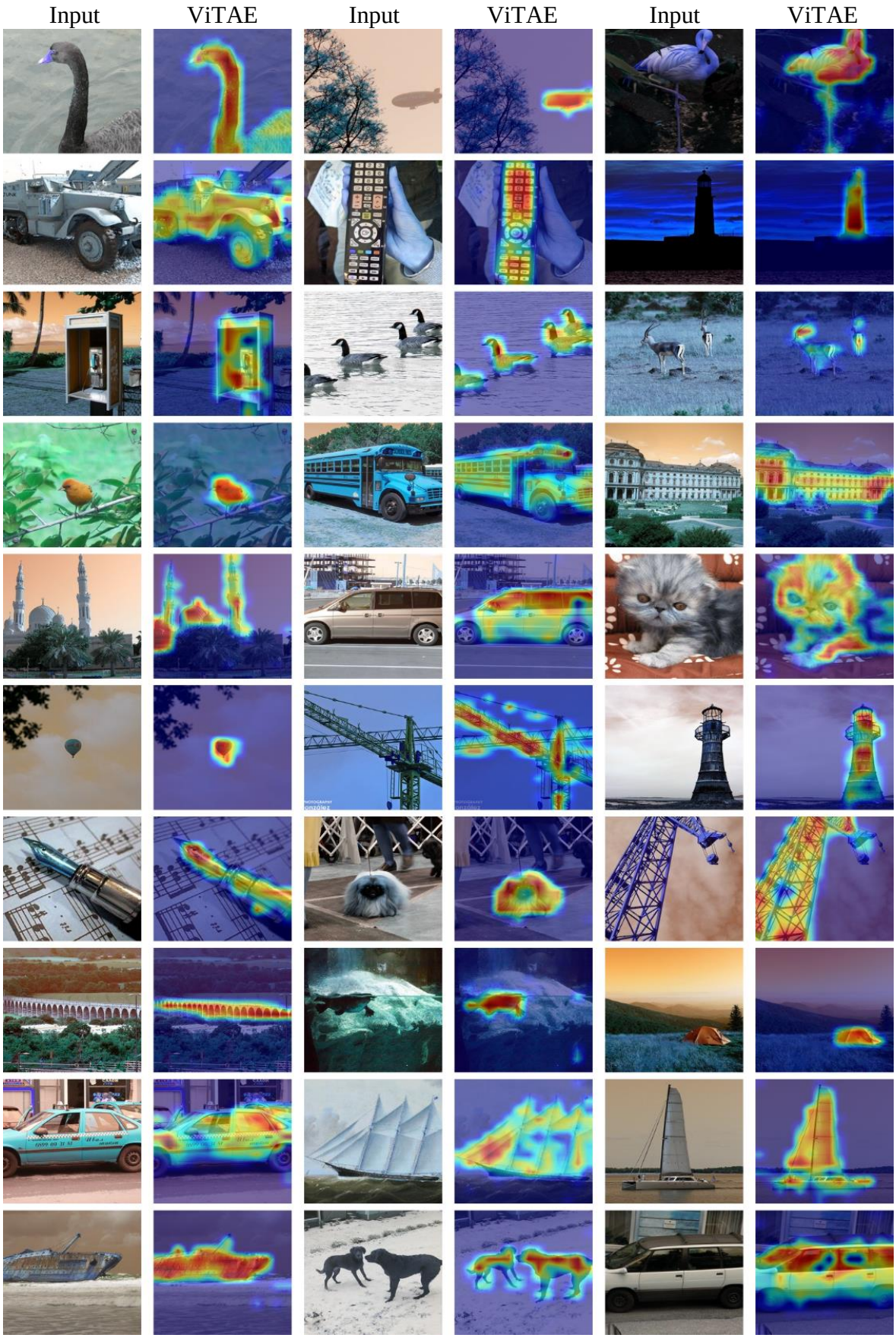


FIGURE 2.9. More visual results of ViTAE.

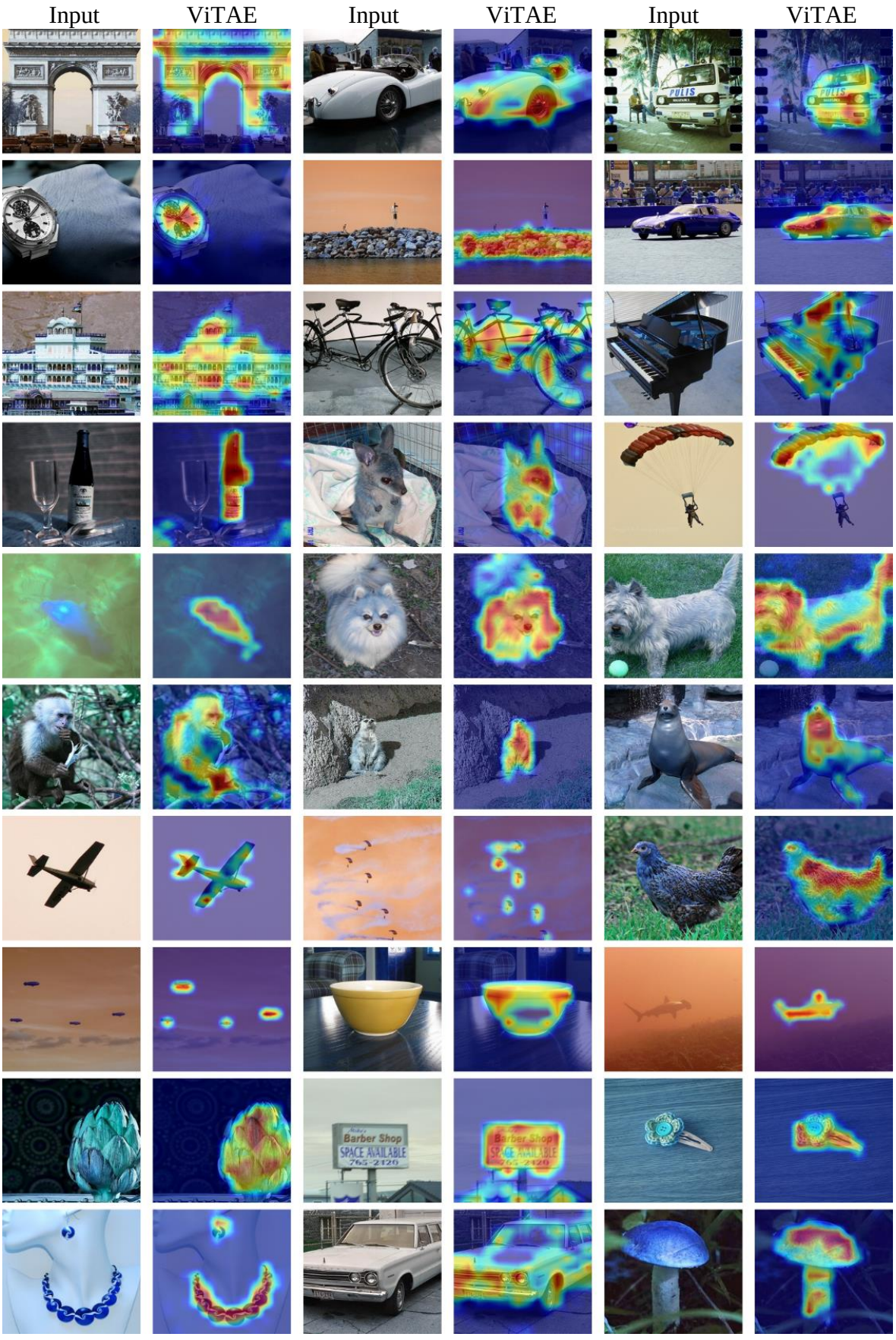


FIGURE 2.10. More visual results of ViTAE.

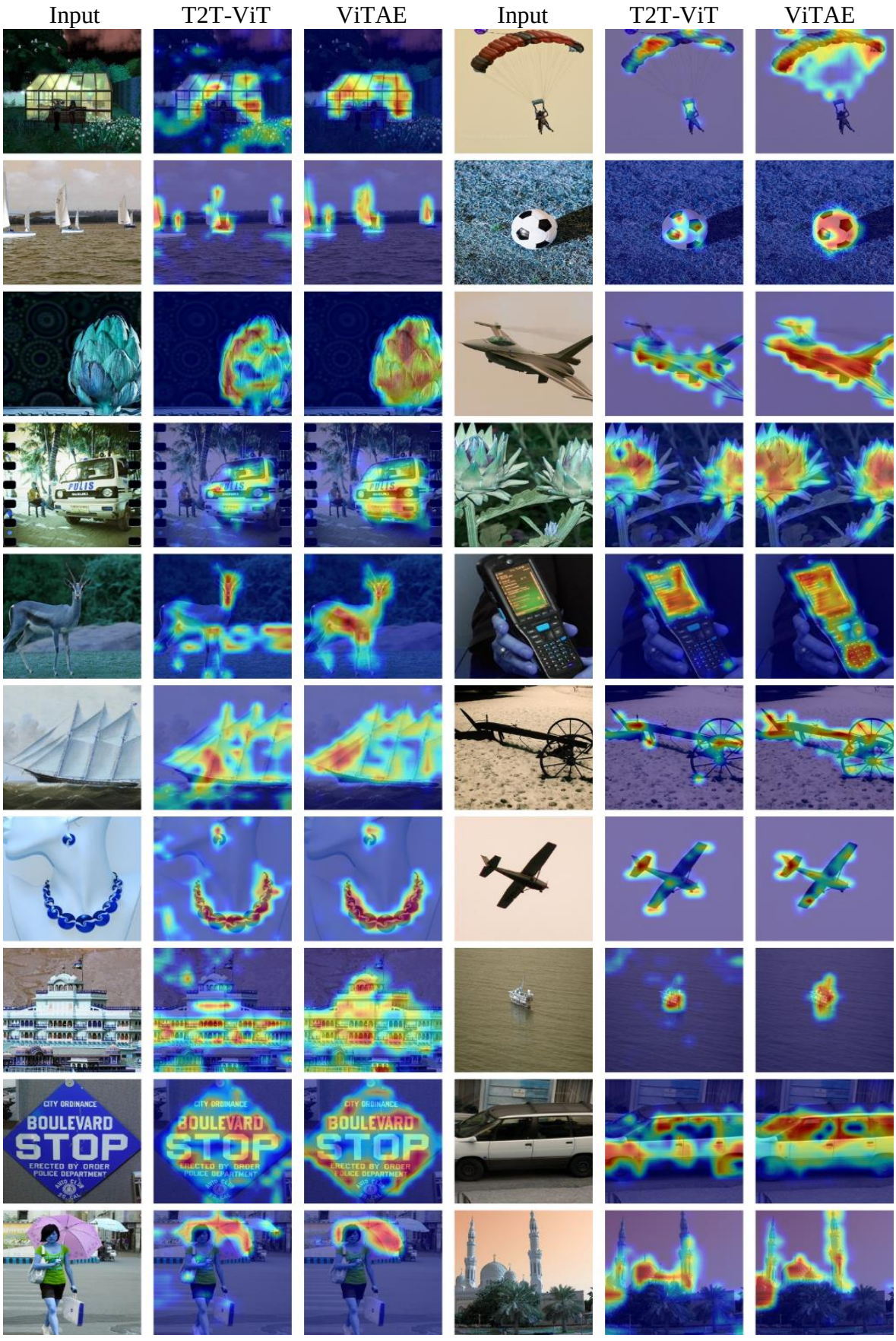


FIGURE 2.11. Visual comparison between ViTAE and T2T-ViT.

CHAPTER 3

RegionCL: Exploring Contrastive Region Pairs for Self-supervised Representation Learning

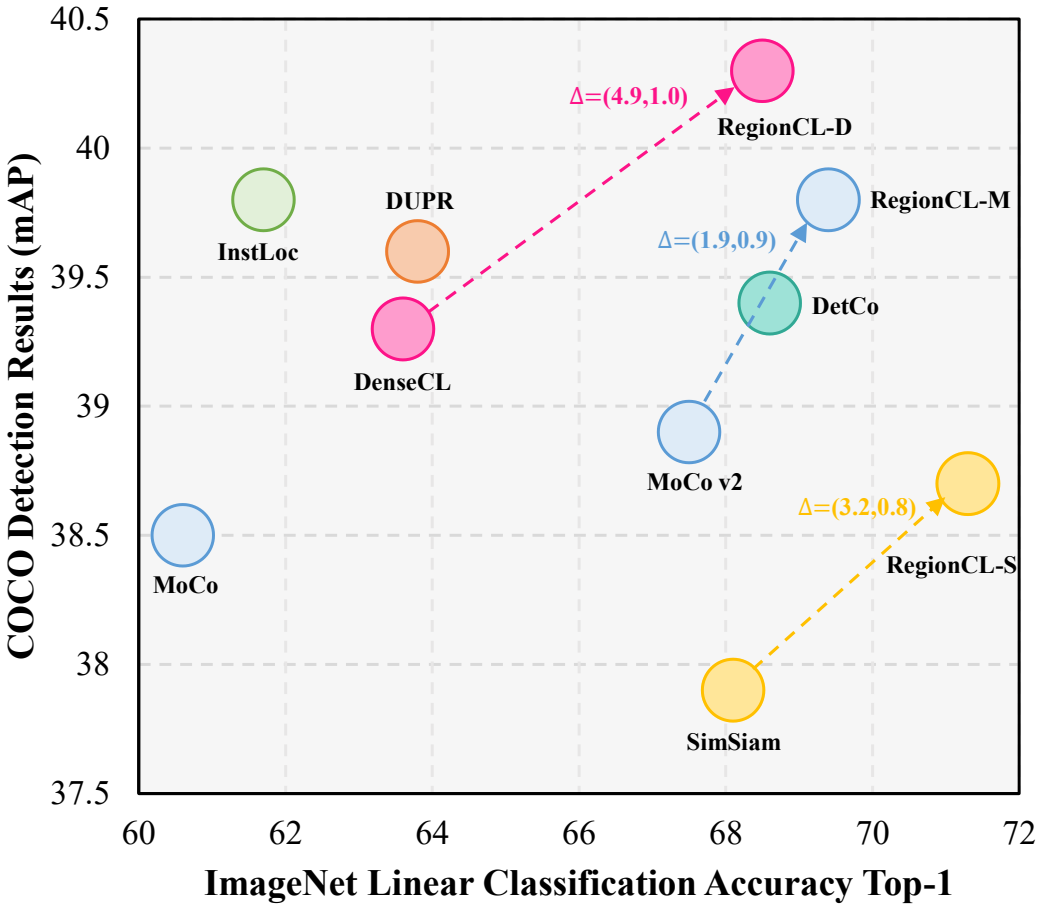
---

3.1 Introduction

Beyond structural modifications, another promising avenue lies in the realm of enhancing inductive bias through training methodologies. This chapter is dedicated to exploring and detailing the advancements made in this domain. Among different training methods, self-supervised learning (SSL) has become an active research topic in computer vision because of its ability to learn generalizable representations from large-scale unlabeled data and offer good performance in downstream tasks [216, 217, 92, 22, 171]. Contrastive learning, one of the popular directions in SSL, has attracted a lot of attention due to its ease of use in pretext designing and its capacity to generalize across various visual tasks.

Current contrastive learning methods typically use augmented views of the same image as positive pairs and maximize their mutual information. Cropping is by far the most popular augmentation technique. By randomly cropping regions from the same images and treating the cropped regions as positive pairs, the methods in [24, 26, 28, 52, 58, 25] have shown promising results in image classification. Multi-crop [17, 18] has been investigated as a way to improve performance even further by generating more diverse candidates and facilitating the model learning a better feature representation. Constrained cropping strategies [227, 186, 199, 140, 191] have recently been developed to ensure that two cropped views contain shared regions of a specific size and to improve models' performance on dense prediction tasks by constructing contrastive pairs within the shared regions. These methods have achieved

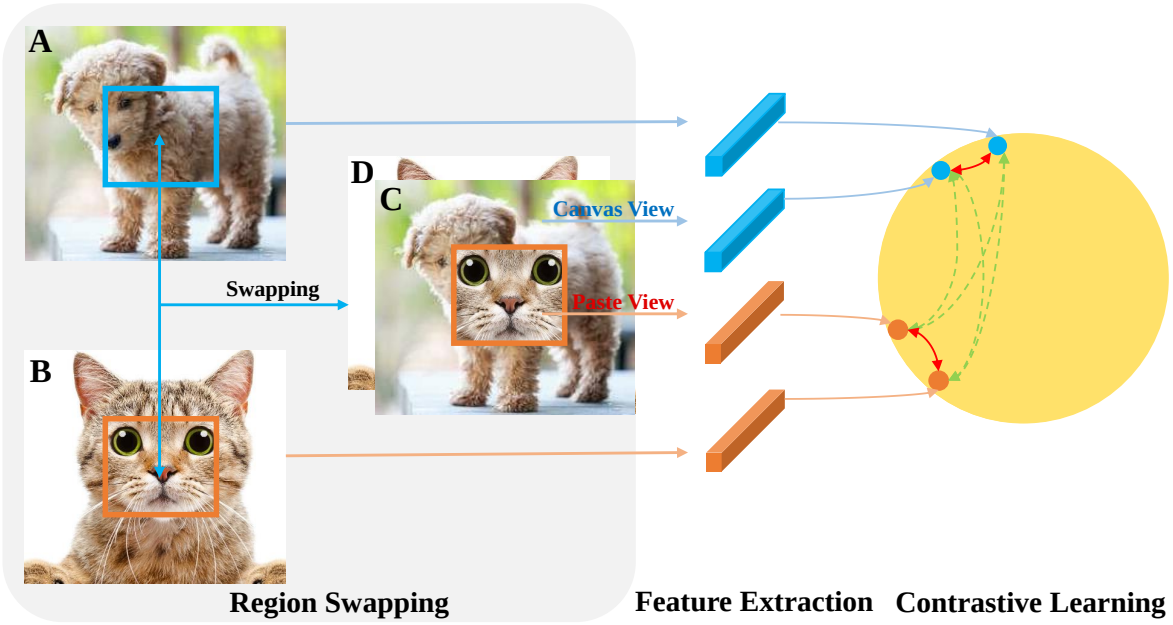
FIGURE 3.1. Transfer results on the ImageNet [40] (classification) and MS COCO [27] (detection) datasets. Involving region-level contrastive pairs during pretraining, RegionCL helps DenseCL [178], MoCov2 [26], and SimSiam [28] achieve a better performance trade-off between image classification and object detection tasks.



superior performance on a variety of visual tasks by leveraging various cropping strategies to construct contrastive pairs during pretraining.

However, the remained regions after cropping have received little attention, despite the fact that the cropped and remained regions together make up the same image instance and both contribute to the category's description. We argue that using both regions during pretraining would help the model learn better complete visual representations of object instances, which will improve the model's performance on downstream classification and dense prediction tasks.

FIGURE 3.2. Illustration of the region swapping strategy. Taking two images A and B as input, it randomly crops and swaps the paste views and generates the composite images C and D. A mask pooling strategy will be used to extract the features from the canvas and paste views respectively. As a result, the canvas and paste views in image C form a negative pair, while the canvas view and A (the paste view and B) are positive pairs. The red lines indicate the positive pairs and the green lines represent the negative pairs.



Based on this motivation, we propose a simple yet effective pretext task called Region Contrastive Learning (RegionCL) to demonstrate the effectiveness of using both regions for contrastive learning. Technically, given two different images, RegionCL randomly crops a region (called the **paste view**) from each image with the same size and swaps them to compose two new images together with the remained regions (called the **canvas view**), respectively. It is worth noting that the two views that compose the new images are from different source images. Then, contrastive pairs can be constructed from the regional perspective following the simple criteria, *i.e.*, each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. In this way, RegionCL generates abundant pairs that contain not only the instance-level pairs as other methods [26, 24] but also the region-level pairs, *e.g.*, the paste and canvas views in the composite images. By exploiting these pairs in popular SSL frameworks, RegionCL helps the models learn better feature representations of object instances owing to the abundant contrastive supervisory signals at

both instance and region levels, delivering better performance on various downstream tasks. As shown in Figure 3.1, RegionCL helps MoCov2 [26], DenseCL [178], and SimSiam [28] improve their linear classification accuracy by 2%∼5% on the ImageNet [40] dataset and object detection performance by 0.8∼1.0 mAP on the MS COCO [27] dataset, simultaneously.

In summary, the contribution of the paper is threefold:

(1) We make the first attempt to demonstrate the importance of both regions, *i.e.*, the cropped and remained regions in cropping, from a complete perspective for self-supervised learning.
(2) We propose a simple yet effective pretext task, *i.e.*, RegionCL, to demonstrate the effectiveness of using both regions for learning. It is compatible with various popular SSL methods with minor modifications and improves their performance on many downstream visual tasks.
(3) Extensive experimental results with MoCov2, SimSiam, and DenseCL on the ImageNet, MS COCO, and Cityscapes datasets demonstrate the effectiveness of the proposed RegionCL on classification, detection, and instance and semantic segmentation tasks.

3.2 Related Work

Self-supervised learning has shown great potential in learning visual representations that can generalize to a series of downstream visual tasks. Early works generate pseudo labels using specific tasks [123, 83, 220, 129] such as image corruption and restoration, reordering, re-colorization. However, the models pretrained in these tasks may be too coupled with the designed tasks and the transfer results on other visual tasks may not be competitive.

Recently, contrastive learning [28, 26, 24, 58, 52, 162, 25] has made rapid progress and shown promising transfer performance. Typically, they take augmented views from the same (different) images as positive (negative) pairs and learn to pull the features from positive pairs while pushing away those from the negative pairs via a contrastive loss. Among the

augmentation techniques, cropping plays an important role in improving the performance, as shown in [24]. Taking the cropped augmented views as input, SimCLR [24] obtains superior results on image classification. MoCo [58, 26] utilizes a momentum encoder to better utilize the cropped views during pretraining, as it provides consistent optimization direction. However, as cropping at a single resolution may not provide enough descriptions of the target object, a multi-crop strategy is explored in [17, 18] by fusing several cropped views at different resolutions. Such a strategy helps the models learn a better feature representation at different scales and boost their performance on the image classification task.

On the other hand, [178, 191, 140, 132] focus on advancing the performance on dense prediction tasks by establishing dense correspondences between the augmented cropped views. Some methods [227, 140, 21, 105] further design constrained cropping strategies during pretraining to improve the transfer results on detection, *e.g.*, they require the two cropped views have some shared regions and attract the dense positive features within the shared regions based on explicit spatial correspondences. By exploring different properties of cropping-based augmented views, these methods obtain superior performance. However, the remained regions after cropping have rarely been explored. Different from them, we make the first attempt to investigate the importance of both regions in this study via adopting a simple region-swapping strategy to generate abundant contrastive pairs at both instance and region levels for contrastive learning (RegionCL), from which the model can learn better visual representations of object instances.

Although several methods also explore region-level contrastive learning, they have not yet explored the complementary remained regions after cropping, *i.e.*, the canvas view in our paper. For example, SCRL [140], DUPR [42], and MaskCo [227] incorporate bounding boxes generation and alignment between the shared area of two cropped views during pretraining. InstLoc [199] further introduces anchors with bounding box augmentations to boost the transfer results on dense prediction tasks at the cost of decreased image classification accuracy. DetCo [186] designs delicate cropping strategies to generate separate patches at different resolutions and uses extra memory banks to capture patch features. Without the requirement of extra information such as bounding boxes alignment, RegionCL adopts the most simple

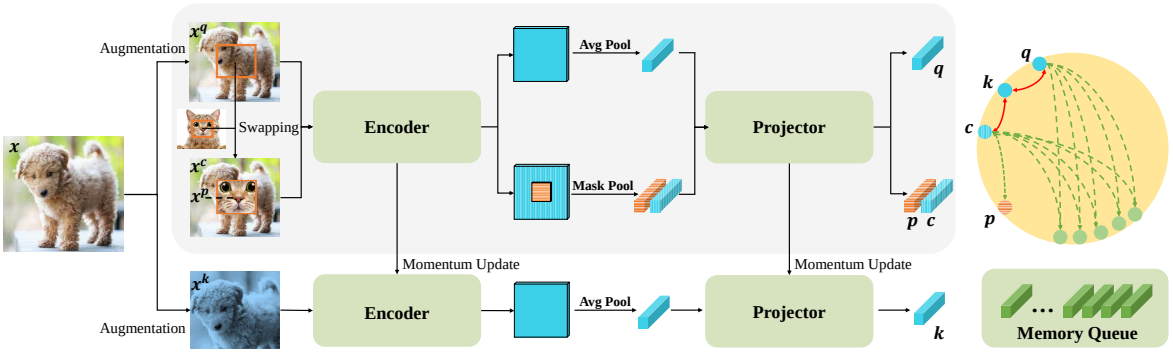


FIGURE 3.3. Illustration of the proposed RegionCL with the MoCov2 framework, *i.e.*, RegionCL-M. Taking the two augmented views x^q, x^k as inputs, RegionCL employs region swapping among the batch of x^q to generate the composite images with paste views x^p and canvas views x^c. Then, mask pooling is used to extract the features belonging to the paste and canvas views, respectively. The resultant region-level features (with stripes in the figure) are batched with the instance-level features and processed by the projector.

strategy to demonstrate that using both regions for contrastive learning isolation helps to boost the SSL methods performance without bells and whistles. The simple task is compatible with popular SSL frameworks, *e.g.*, we validate the effectiveness of using both regions for training with MoCov2 [26], SimSiam [28], and DenseCL [178], with only minor modifications to them. The theoretical analysis of these works has been well studied in [163, 160, 172, 174] and is beyond this paper's scope. It is also noteworthy that although the adopted swapping strategy is similar to co-current methods like UnMix [147]/HEXA [86]/InsCon [200], they do not explore region-level pairs, *i.e.*, they still treat the composite images as a whole from the global perspective. RegionCL explores both region- and global-level pairs from the composite images, which is more efficient and obtains better performance on various vision tasks.

3.3 Method

3.3.1 The region swapping strategy

Different from current methods that only use the cropped regions, we take both the cropped and remaining regions into consideration for self-supervised learning. Given two different

images, we randomly crop a region with the same size from each image and swap them to compose two new images. As shown in Figure 3.2-C, the composite image after region swapping contains two views: one is the paste view (the cropped region), *i.e.*, the cat's face, and the other is called the canvas view (the remained region), *i.e.*, the dog's body. Specifically, we first sample the size of the paste view, *i.e.*, the height and width, and then determine the coordinates of the origin point from which the cropping starts. We make sure the size and location of the cropped region match the network's downsampling ratio $\mathcal{R}$ during region swapping so that the region's feature can be directly extracted from the feature map by a simple operation of mask pooling. The height and width are determined by $\mathcal{R}$ and a discrete uniform distribution $\mathcal{C} \sim U(\mathcal{C}_L, \mathcal{C}_U)$, where the ratio $\mathcal{R}$ is typically 32 for ResNet [61] and $\mathcal{C}_L, \mathcal{C}_U$ are two predefined hyper-parameters shared for both spatial dimensions for simplicity. They are set to 3 and 5 in the paper unless specified. We sample twice from the distribution $\mathcal{C}$ and get two observations c_h and c_w. Then, we calculate the width and height as $r_h = c_h \times \mathcal{R}, r_w = c_w \times \mathcal{R}$, respectively. Then we uniformly sample the origin point coordinates (r_x, r_y) from a valid range that guarantees there is enough remaining area to crop a patch of size $r_w \times r_h$. In this way, the candidate region is determined by (r_x, r_y, r_w, r_h). It is noteworthy that within a mini-batch of the training images, we use the same coordinates (r_x, r_y, r_w, r_h) for efficient batch-wise implementation during training.

3.3.2 The region contrastive learning

The architecture. We take MoCov2 as an example here to describe the proposed RegionCL method in depth, denoted as RegionCL-M. The overall architecture is presented in Figure 3.3. As can be seen, RegionCL-M has exactly the same architecture with MoCov2 and only requires marginal modifications to the inputs and learning objectives, *i.e.*, a region-level branch in the middle.

RegionCL-M uses a Siamese network structure in pretraining. Given image instances x, RegionCL-M first creates two randomly augmented views, *i.e.*, the query view x^q and the key view x^k following the same augmentation strategy as in MoCov2. The online network processes the query view, and the other branch, *i.e.*, the momentum updated network, processes

the key view. Unlike other methods that also utilize region-level contrastive learning [140, 42, 227], we follow the same cropping strategies as in MoCov2 and do not need the two views x^q, x^k to have a sufficiently large overlap, which keeps the diversity of the contrastive pair candidates. We construct the region-level contrastive pairs using the region-swapping strategy. Specifically, given two image instances from the query view x^q, we randomly crop a region with the same size in each image instance and swap them to compose two new images x^{pc}, where the cropped region after swapping and the remained region in the new images are the paste view x^p and canvas view x^c, respectively.

The region- and instance-level contrastive loss. In this way, we have a total of four different views, *i.e.*, the query view, the paste view, the canvas view, and the key view, denoted as x^q, x^p, x^c, x^k, respectively. RegionCL-M projects these views into the corresponding feature representations q, p, c, k, among which the features q, k are instance-level feature representations while the features p, c are region-level feature representations. Note that features of the paste view and canvas view are extracted from the feature maps of x^{pc} via mask pooling, where the mask is obtained according to the coordinates (r_x, r_y, r_w, r_h) as described in Section 3.3.1. The other views' features are from the global average pooling upon the corresponding feature maps. Then we can efficiently construct the contrastive pairs for these views according to the simple criteria, *i.e.*, each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. We follow the practice of MoCov2 in our implementation and ignore the positive pairs whose features are both generated by the online network to stabilize the training.

We use contrastive loss [54, 58] as the learning objectives, which can be thought of as training an encoder for a dictionary lookup task at both instance and region levels. We first introduce the instance-level contrastive loss and then present the region-level one. Assume that we have a set of encoded samples $\{k_i|i = 1, 2, ..., K\}$ as keys of a dictionary. For each query feature q, if there is a single key (k^+) that matches the query q, the contrastive loss aims to increase the similarity between q and k^+ meanwhile reducing the similarity between q and all other keys (considered as the negative counterparts for q). We use L2-normalized dot product to measure the similarity between the queries and keys, and the contrastive loss, *i.e.*, the InfoNCE [125]

loss, is therefore formulated as:

$$\mathcal{L}_{ins} = -\log \frac{\exp\left(q \cdot k^{+}/\tau\right)}{\sum_{i=0}^{K} \exp\left(q \cdot k_i/\tau\right)}, \tag{3.1}$$

where τ is a temperature hyper-parameter (set to 0.2 by default) [183, 58]. Following MoCov2, the dictionary keys $\{k_i|i = 1, 2, ..., K\}$ in RegionCL-M are maintained using a first-in-first-out queue with a predefined maximum number of samples (K), which is set to 65,536. The features from the key view k is treated as the positive sample and used to progressively update the memory queue, which serves as the negative samples. This form of contrastive loss is the exact one that appeared in MoCov2 [58], while it can have other forms for different SSL methods [125, 28]. Apart from the instance-level pairs, the features of other views formulate the region-level pairs with the modified contrastive loss:

$$\begin{aligned}\mathcal{L}_{reg} = &-\frac{1}{2}\log \frac{\exp\left(p \cdot k_p^{+}/\tau\right)}{\sum_{i=0}^{K} \exp\left(p \cdot k_i/\tau\right) + \exp\left(p \cdot sg(c)/\tau\right)} \\ &-\frac{1}{2}\log \frac{\exp\left(c \cdot k_c^{+}/\tau\right)}{\sum_{i=0}^{K} \exp\left(c \cdot k_i/\tau\right) + \exp\left(c \cdot sg(p)/\tau\right)},\end{aligned} \tag{3.2}$$

where the features p, c are obtained from the identical composite image x^{pc} (thus the term is divided by $\frac{1}{2}$ for normalization). $sg(\cdot)$ represents the 'stop gradient', which helps stabilize the training. p and c are indeed hard negative pairs since they involve some context information from each other due to convolution and pooling operations, thereby helping the model to learn robust and discriminative feature representations. Thus the total contrastive loss is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{ins} + \mathcal{L}_{reg}. \tag{3.3}$$

Since the query, canvas, and paste views share the online network, we believe that the features of these views should be in the same feature space. Thus, RegionCL-M only needs a single queue to provide negative samples for features of all the three views, in contrast to the usage of multiple queues as in [186, 199].

3.3.2.1 Extension to other SSL methods.

As RegionCL defines a model-agnostic pretext task and requires minor modifications to the SSL methods, we also choose two other representative approaches, *i.e.*, DenseCL [178] and SimSiam [28], to further validate its effectiveness, denoted as RegionCL-D and RegionCL-S, respectively. Specifically, DenseCL [178] focuses on dense prediction tasks and has two learning objectives, *i.e.*, the instance-level contrastive loss as in MoCov2 and the pixel-level dense loss. Therefore, RegionCL-D includes the proposed region-level loss seamlessly as in RegionCL-M and keeps the pixel-level loss unchanged. For SimSiam [28], it only adopts instance-level positive pairs $\{p, k^+\}$ for training. Thus, we only enrich the positive pairs by collecting those abundant instance- and region-level positive pairs provided by RegionCL-S while retaining the other components. Similar strategy is adopted for vision transformer backbones [196, 109, 45] with MoBy [190]. Please refer to the supplementary for details.

3.4 Experiments

To thoroughly validate the improvements brought by introducing both regions into pretraining, we incorporate the RegionCL in representative state-of-the-art SSL methods, *i.e.*, MoCov2 [26], DenseCL [178], and SimSiam [28], and propose the RegionCL compatible models, *i.e.*, RegionCL-M, RegionCL-D, and RegionCL-S. The models are pretrained following the same settings as their own base methods, *i.e.*, we train RegionCL-M and RegionCL-D for 200 epochs, and RegionCL-S for 100 epochs, with an SGD [155] optimizer and corresponding augmentations, respectively. All the methods are based on ResNet-50 [61] backbone. Please refer to the supplementary material for more details. Vision transformer-based methods [190] are also explored to further evaluate RegionCL's effectiveness.

3.4.1 Image classification on ImageNet

Settings. To evaluate the effectiveness of regional contrastive learning for image classification, we benchmark the RegionCL on ImageNet [40], which contains 1.28M images in the training

set and 50K images in the validation set from 1,000 classes, respectively. The pretrained models of other SSL methods are either obtained from their authors or reproduced using their official codes. The performance of Top-1 and Top-5 accuracy on a single crop is reported. We have two experimental settings regards evaluation: linear classification and few-shot finetuning. The former setting follows the default setting of MoCov2 [58, 26] and SimSiam [28] with SGD [155] and LARS [203] optimizer. The latter one using randomly sampled data per class from the training set.

Results with linear classification. We report the linear classification results of different methods in Table 3.1 and 3.2. 'Real' indicates that the labels used for evaluation are provided by [8]. From the table, we can see that RegionCL improves the aforementioned SSL baseline methods significantly by a large margin: +1.9% for RegionCL-M, +4.8% for RegionCL-D, and +3.2% for RegionCL-S. This proves RegionCL helps various SSL methods learn better feature representations owing to the abundant contrastive supervisory signals with contrastive pairs from both regional and global perspective. Comparing with methods that also exploring mixing two images for contrastive learning but from the global perspective only, *i.e.*, UnMix [147] and HEXA [86], RegionCL-M obtains better performance no matter using 200/800 epochs for training, demonstrating the benefits of leveraging regional contrastive pairs. Besides, RegionCL-S reaches 71.3% Top-1 accuracy using only 100 epochs, while the vanilla SimSiam requires a significantly longer training schedule of 800 epochs, proving the effectiveness of RegionCL in accelerating the model convergence and improving the performance. It is noteworthy that DenseCL focuses on dense prediction tasks and does not perform that well on classification. In contrast, RegionCL brings a large improvement on DenseCL for image classification, indicating that introducing remained regions to promote pretraining is not only compatible with classification-favored SSL methods but also generalizes well on dense prediction-favored approaches.

Results with linear few-shot finetuning. Table 3.3 presents the results of different methods at the linear few-shot finetuning setting. Thanks to the abundant contrastive pairs brought by RegionCL, the models pretrained by RegionCL have learned better feature representations from a complete perspective and can generalize well on classification tasks, thus delivering

TABLE 3.1. Linear classification results comparison on ImageNet [40].

	Epochs	ImageNet Top-1	ImageNet Top-5	Real Top-1
MoCo [58]	200	60.6	-	69.1
SimCLR [24]	1000	69.3	89.0	77.6
PIRL [114]	800	64.3	-	71.7
CPC-v2 [63]	200	63.8	85.3	-
InstLoc [199]	200	61.7	-	-
MaskCo [227]	200	65.1	-	-
ISD [161]	200	69.8	-	-
PCLv1 [89]	200	61.5	-	-
PCLv2 [89]	200	67.6	-	-
DUPR [42]	200	63.8	85.6	-
DetCo [186]	200	68.6	88.5	-
UnMix [147]	200	68.6	-	-
HEXA [86]	200	68.9	-	-
MoCov3 [29]	100	68.9	-	-
SimSiam [28]	800	71.3	-	-
MoCov2 [26]	200	67.5	88.2	77.8
RegionCL-M	**200**	**69.4**	**89.6**	**78.7**
DenseCL [178]	200	63.6	85.5	72.3
RegionCL-D	**200**	**68.5**	**89.0**	**78.4**
SimSiam [28]	100	68.1	88.2	77.8
RegionCL-S	**100**	**71.3**	**90.4**	**80.8**

TABLE 3.2. Linear classification results with more pretraining epochs. * means using symmetric loss.

	Epochs	ImageNet Top-1	ImageNet Top-5
MoCov2 [26]	800	71.1	90.2
UnMix [147]	800	71.8	-
HEXA [86]	800	71,7	-
RegionCL-M	**800**	**73.1**	**91.5**
MoCov2* [28]	800	72.3	-
RegionCL-M*	**800**	**73.9**	**92.0**
MoCov3 [29]	1000	74.6	-
RegionCL-M3	**1000**	**75.4**	**92.6**

much more significant improvements over their baselines when only a limited number of data are available for finetuning, *i.e.*, a gain of +2.5%, +8.9%, +9.5% for 1% data and 1.8%, 6.4%, 8.1% for 10% data achieved by RegionCL-M, RegionCL-D, RegionCL-S.

TABLE 3.3. Linear classification results on ImageNet [40] using 1% and 10% data.

	1% Data		10% Data	
	Top-1	Top-5	Top-1	Top-5
MoCov2 [26]	43.6	70.9	58.8	82.4
RegionCL-M	**46.1**	**72.9**	**60.4**	**83.5**
DenseCL [178]	38.9	66.2	54.0	79.3
RegionCL-D	**47.8**	**74.0**	**60.4**	**83.1**
SimSiam [28]	32.8	61.5	51.8	77.7
RegionCL-S	**42.3**	**70.6**	**59.9**	**83.8**

3.4.2 Detection and segmentation on MS COCO

Settings. We show the detection performance of the models pretrained with the RegionCL pretext task. The experiments are conducted on the MS COCO dataset [27], which contains about 118K images with bounding boxes and instance segmentation annotations and covers 80 object categories in total. We choose two representative detectors: the two-stage detector Mask-RCNN [59] and the one-stage detector RetinaNet [99], following the same settings as in [186, 227].

TABLE 3.4. Object detection results on the MS COCO [27] dataset with Mask-RCNN [59] C4 and FPN (1x).

	Mask-RCNN-C4-1x						Mask-RCNN-FPN-1x					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	26.4	44.0	6.9	7.6	14.8	7.2	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	38.2	58.2	41.2	33.3	54.7	35.2	38.9	59.6	42.7	35.4	56.5	38.1
InsDis [183]	37.7	57.0	40.9	33.0	54.1	35.2	37.4	57.6	40.6	34.1	54.6	36.4
PIRL [114]	37.4	56.5	40.2	32.7	53.4	34.7	37.5	57.6	41.0	34.0	54.6	36.2
SwAV [17]	32.9	54.3	34.5	29.5	50.4	30.4	38.5	60.4	41.4	35.4	57.0	37.7
MoCo [58]	38.5	58.3	41.6	33.6	54.8	35.6	38.5	58.9	42.0	35.1	55.9	37.7
DetCo [186]	39.4	59.2	42.3	34.4	55.7	36.6	39.5	60.3	43.1	35.9	56.9	38.6
DetCo-AA [186]	39.8	59.7	43.0	34.7	56.3	36.7	40.1	61.0	43.9	36.4	58.0	38.9
MoCo v2 [26]	38.9	58.4	42.0	34.2	55.2	36.5	38.9	59.4	42.4	35.5	56.5	38.1
RegionCL-M	**39.8**	**59.8**	**43.0**	**34.8**	**56.4**	**36.9**	**40.1**	**60.7**	**43.9**	**36.3**	**57.7**	**39.0**
DenseCL [178]	39.3	59.1	42.2	34.5	55.6	36.8	39.1	59.4	42.5	35.5	56.4	38.0
RegionCL-D	**40.3**	**60.3**	**43.9**	**35.2**	**57.0**	**37.3**	**40.4**	**61.3**	**44.2**	**36.7**	**58.2**	**39.4**
SimSiam [28]	37.9	57.5	40.9	33.2	54.2	35.2	37.3	57.2	40.5	33.9	54.2	36.1
RegionCL-S	**38.7**	**58.2**	**41.3**	**33.7**	**55.0**	**35.6**	**38.8**	**58.8**	**42.4**	**35.2**	**56.0**	**37.6**

Results of Mask-RCNN on MS COCO. Table 3.4 and 3.5 summarize the Mask-RCNN results on 1x and 2x schedules respectively, where the RegionCL variants are highlighted in bold. We can see that RegionCL has significantly improved all approaches with ResNet50-C4 and ResNet50-FPN backbones, confirming the benefits of region-level contrastive learning on various SSL methods. According to the tables, incorporating RegionCL into MoCov2 (RegionCL-M) can further improve the performance over the MoCov2 baseline with both backbones. It is also noticeable that RegionCL-M has already surpassed the previous representative SSL methods designed for dense prediction, *e.g.*, DetCo [186] and DenseCL [178]. More importantly, when incorporating RegionCL into DenseCL, RegionCL-D achieves the best scores for all metrics in both the 1x and 2x settings. It suggests that RegionCL can still help the dense prediction-favored methods to learn more discriminative features from a complete perspective.

TABLE 3.5. Object detection results on the MS COCO [27] dataset with Mask-RCNN [59] C4 and FPN (2x).

	Mask-RCNN-C4-2x						Mask-RCNN-FPN-2x					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	35.6	54.6	38.2	31.4	51.5	33.5	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	40.0	59.9	43.1	34.7	56.5	36.9	40.6	61.3	44.4	36.8	58.1	39.5
MoCo [58]	40.7	60.5	44.1	35.4	57.3	37.6	40.8	61.6	44.7	36.9	58.4	39.7
MaskCo [227]	40.8	60.5	44.2	35.5	57.1	38.0	-	-	-	-	-	-
UnMix [147]	-	-	-	-	-	-	41.2	60.9	44.7	-	-	-
DetCo [186]	41.4	61.2	44.7	35.8	57.8	38.3	41.5	62.1	45.6	37.6	59.2	40.5
DetCo-AA [186]	41.3	61.2	45.0	35.8	57.9	38.2	41.5	62.5	45.6	37.7	59.5	40.5
MoCo v2 [26]	41.0	60.6	44.5	35.6	57.2	38.0	40.9	61.5	44.7	37.0	58.7	39.8
RegionCL-M	**41.5**	**61.4**	**44.9**	**35.9**	**57.7**	**38.6**	**41.6**	**62.5**	**45.6**	**37.7**	**59.3**	**40.4**
DenseCL [178]	39.7	59.1	43.0	34.5	55.9	37.3	41.4	62.1	45.1	37.5	58.8	40.3
RegionCL-D	**41.8**	**61.6**	**45.4**	**36.4**	**58.5**	**39.2**	**42.1**	**62.9**	**45.9**	**38.0**	**60.0**	**40.7**
SimSiam [28]	38.8	58.0	41.9	34.0	55.1	36.2	40.1	60.6	43.8	36.4	57.7	39.1
RegionCL-S	**40.7**	**60.4**	**44.4**	**35.4**	**57.0**	**37.7**	**41.0**	**61.6**	**44.7**	**37.1**	**58.6**	**39.8**

Results of RetinaNet on MS COCO. The results of RetinaNet on MS COCO using different SSL methods are presented in Table 3.6. From the table, we can see that the improvement brought by RegionCL still holds in all metrics and at all the training settings. Similarly, RegionCL-D achieves the best results at 38.8 AP and 40.6 AP for the two training schedules respectively, significantly surpassing the supervised baseline by 1.4 AP and 1.7 AP. It is also

noted that the improvement in the more stringent metric AP^{bb}_{75} is more significant than that in the AP^{bb} metric, demonstrating that leveraging both cropped and remained regions for contrastive learning contributes to learning better feature representations for object detection and thus improving the detection accuracy. These results show that the simple strategy RegionCL can help existing SSL methods achieve a better trade-off between the classification and detection tasks (see Figure 3.1), further validating the benefits of using both regions for pretraining.

3.4.3 Segmentation on Cityscapes

Settings. Further, we evaluate the models' transfer performance for both instance and semantic segmentation on the Cityscapes [36] dataset, which contains over 5K well-annotated images of street scenes from 50 different cities. We follow the same setting as in MoCov2 [58] for instance segmentation, with Mask-RCNN and trained for 24K iterations. UPerNet [185] in mmseg [35] is employed for semantic segmentation evaluation.

Results on Cityscapes. Table 3.7 presents the performance of different SSL methods and their variants with RegionCL. The second and third columns show the performance for instance and semantic segmentation, respectively. According to the table, RegionCL consistently improves the three representative SSL methods by large margins. For example, RegionCL-M reaches the best on instance segmentation at 34.9 AP and 62.5 AP_{75}, while RegionCL-D outperforms the others on semantic segmentation at 78.7 mIoU and 79.5 mIoU with different training schedules, confirming the generalization and the effectiveness of region-level contrastive learning with remained regions on segmentation tasks.

3.4.4 Generalization on vision transformer

To further evaluate the benefits of using both regions for pretraining, we apply RegionCL on vision transformer-based methods and train them following the design in MoBy [190]. As shown in Table 3.8, RegionCL improves MoBy by over 3.0 accuracy on both ImageNet and ImageNet Real with 100 epochs pretraining and over 1.5 accuracy with 300 epochs

TABLE 3.6. Detection results on MS COCO [27] with RetinaNet [99].

	RetinaNet-1x			RetinaNet-2x		
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
Rand Init	24.5	39.0	25.7	32.2	49.4	34.2
Supervised	37.4	56.5	39.7	38.9	58.5	41.5
InsDis [183]	35.5	54.1	38.2	38.0	57.4	40.5
PIRL [114]	35.7	54.2	38.4	38.5	57.6	41.2
SwAV [17]	35.2	54.9	37.5	38.6	58.8	41.1
MoCo [58]	36.3	55.0	39.0	38.7	57.9	41.5
DUPR [42]	38.0	57.2	40.7	40.0	59.6	43.0
DetCo [186]	38.0	57.4	40.7	39.8	59.5	42.4
DetCo-AA [186]	38.4	57.8	41.2	39.7	59.3	42.6
MoCov2 [26]	37.2	56.2	39.6	39.3	58.9	42.1
RegionCL-M	**38.4**	**58.1**	**41.2**	**40.1**	**59.9**	**43.2**
DenseCL [178]	37.7	56.4	40.2	39.8	59.2	42.8
RegionCL-D	**38.8**	**58.6**	**41.6**	**40.6**	**60.4**	**43.6**
SimSiam [28]	35.5	53.7	38.1	38.1	57.4	40.8
RegionCL-S	**36.8**	**55.9**	**39.5**	**39.1**	**58.5**	**41.8**

TABLE 3.7. Instance (Inst.) and semantic (Sem.) segmentation results (mIoU) on Cityscapes [36].

	Inst. Seg		Sem. Seg	
	AP	AP_{75}	40K	80K
Supervised	32.9	59.6	77.1	78.2
MoCov2 [26]	33.9	60.8	77.8	78.6
RegionCL-M	**34.9**	**62.5**	**78.1**	**79.0**
DenseCL [178]	34.5	61.9	78.3	79.1
RegionCL-D	**34.8**	**62.3**	**78.7**	**79.5**
SimSiam [28]	33.6	61.0	76.2	78.1
RegionCL-S	**34.9**	**61.6**	**77.8**	**78.7**

pretraining. It can be concluded that RegionCL can successfully improve the vision transformer's performance, validating its effectiveness for advanced neural architectures like vision transformers.

3.4.5 Ablation Study

We conduct the ablation studies with RegionCL-M. All models are trained for 100 epochs due to the limitation of computation resources and follow the practice of previous works [186, 227].

TABLE 3.8. Linear classification results comparsion on ImageNet with vision transformer-based method MoBy [190].

	Backbone	Epochs	ImageNet Top-1	ImageNet Top-5	Real Top-1
MoBy [190]	Swin-T	100	70.9	89.7	77.5
RegionCL+MoBy	Swin-T	100	73.9	91.8	81.2
MoBy [190]	Swin-T	300	75.3	92.2	82.4
RegionCL+MoBy	Swin-T	300	77.0	93.1	83.9

TABLE 3.9. The influence of paste view's size. $\mathcal{C}_L$ and $\mathcal{C}_U$ denote the lower and upper bound of the size for the paste view generation. '-' denotes no paste view is used during the training, *i.e.*, downgrading to the original MoCov2.

Configuration		ImageNet		MS COCO	
$\mathcal{C}_L$	$\mathcal{C}_U$	20-NN	100-NN	AP^{bb}	AP^{mk}
-	-	49.3	47.3	26.3	24.0
1	5	51.8	49.8	27.3	24.9
2	5	53.0	51.2	27.9	25.5
3	**5**	**54.7**	**52.7**	**28.0**	**25.6**
4	5	53.9	51.8	27.9	25.5
3	4	**55.1**	**52.8**	27.8	25.5
3	6	54.3	52.5	27.8	25.5

TABLE 3.10. The influence of paste and canvas views. 'Paste'/'Canvas' denote using paste/canvas views as positive pairs. 'Neg' means using the canvas and paste counterpart views in the composited images as negative pairs.

Configuration			ImageNet		MS COCO	
Paste	Canvas	Neg	20-NN	100-NN	AP^{bb}	AP^{mk}
×	×	×	49.3	47.3	26.3	24.0
×	✓	×	51.8	50.0	26.9	24.7
✓	×	×	50.0	50.2	27.0	24.7
✓	✓	×	54.6	52.4	27.8	25.5
✓	✓	✓	54.7	52.7	28.0	25.6

We adopt a k-NN classifier to evaluate their classification accuracy on ImageNet [40] and train these models for 12K iterations on MS COCO [27] to evaluate their dense prediction performance.

The size of the paste view. We investigate the influence of the size of the paste view by varying the lower and upper bounds $\mathcal{C}_L$ and $\mathcal{C}_U$. Note that the image size is set as 224 during the training and the downsampling ratio for the backbone network ResNet-50 [61] is 32, thus

$\mathcal{C}_L, \mathcal{C}_U$ are valid in the range $[1, 7]$. The optimal hyper-parameters are determined through two steps. **(1)** We first fix the upper bound $\mathcal{C}_U$ to 5 and search different configurations for the lower bound $\mathcal{C}_L$. As shown in Table 3.9, the performance on both classification and detection peaks with $\mathcal{C}_L = 3$. **(2)** Then we fix $\mathcal{C}_L$ to 3 and search for $\mathcal{C}_U$. It is interesting to see that decreasing $\mathcal{C}_U$ from 5 to 4 slightly improves the performance on classification but degrades that on detection. This suggests that the optimal configurations of $\mathcal{C}_L, \mathcal{C}_U$ for classification and dense prediction tasks may be slightly different, and we select $\mathcal{C}_L = 3, \mathcal{C}_U = 5$ as default values to achieve a trade-off on both classification and dense prediction tasks.

The influence of using paste and canvas views. We further investigate the importance of using both paste and canvas views during pretraining. The results are concluded in Table 3.10, where ✓under Paste or Canvas denotes whether to use the former or latter term in Eq. (3.2). The 'Negative' option means whether to treat the canvas and paste counterpart views from the same composite image x^{pc} as negative pairs, *i.e.*, $\exp(c \cdot sg(p)/\tau)$ or $\exp(p \cdot sg(c)/\tau)$ in the denominator in Eq. (3.2). With all columns marked $\times$, the method becomes standard MoCov2. From the first three rows in the table, we can see that using either the paste or canvas views can bring performance gains, and the performance will be further boosted when considering both regions. For example, in the 4th row, the model gains more than 5% accuracy improvement over MoCov2 in both ImageNet 20- and 100-NN classification. We attribute it to that leveraging both regions during pretraining can help models learn better category features from a complete perspective. Comparing the 4th row with the last row, where intra-image negative pairs are utilized, the performance on both tasks is slightly improved. It demonstrates that using negative pairs within images can help models learn more discriminative features between different regions, again validating the importance of introducing region-level contrastive pairs in self-supervised learning.

3.4.6 Analysis of RegionCL

To better analyze the performance gains brought by RegionCL, we monitor the KNN accuracy and gradients of the target regions during training. We use YOLOX [49] to detect the objects in the images and record the average magnitude of gradients back-propagated from the loss

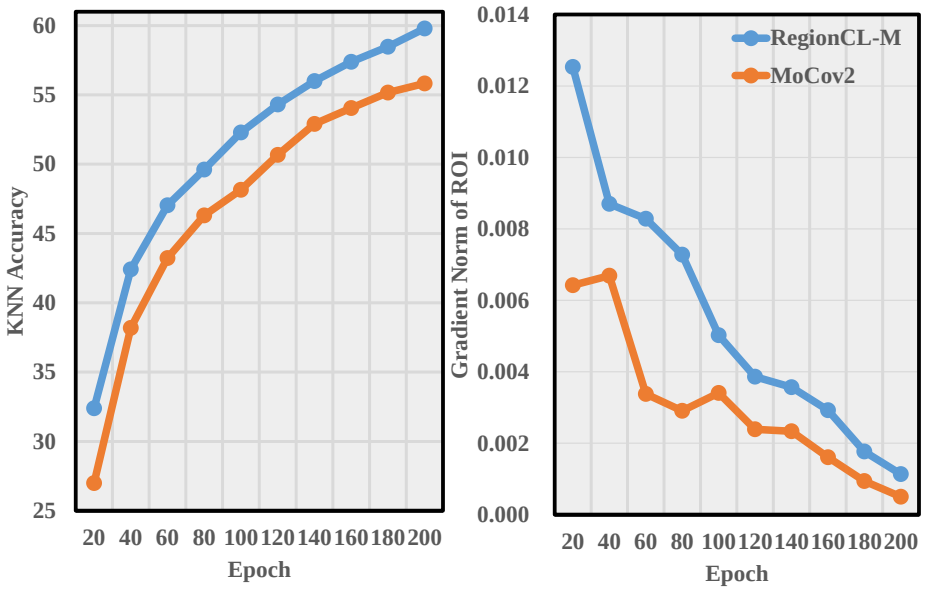


FIGURE 3.4. The KNN accuracy and average gradient magnitude of MoCov2 [26] and RegionCL-M with different training epochs.

of MoCov2 and RegionCL-M in the object bounding boxes on the training set. As shown in Figure 3.4, the KNN accuracy of RegionCL-M consistently outperforms MoCov2 while the gradients inside the object regions of RegionCL-M are larger than those of MoCov2, implying that using both regions for training can help models pay more attention on the targets and learn better object representation from a complete perspective.

3.5 Limitation and Discussion

We make the first attempt to demonstrate the importance of considering both cropped and remained regions in SSL via a simple yet effective RegionCL pretext task. The simple task makes minor modifications to representative SSL methods to show the benefits brought by leveraging regional contrastive pairs. Although RegionCL effectively improves these methods, there is still much to be explored in utilizing the remained regions for more tasks [194, 195, 92, 6], *e.g.*, introducing additional bounding box selection and alignment modules or adopting multi-level supervision for learning. Besides, RegionCL temporally costs more computations per iteration (about 50% for RegionCL-M). Although it demonstrates better results with fewer

computation resources as discussed in the supplementary, it will be beneficial to reduce the training costs, which we leave as our future work. Besides, we demonstrate the effectiveness of RegionCL using the ImageNet dataset following the common practice. However, more types of datasets can be explored during the pre-training stage to bring better task-related features. For example, using the MS COCO dataset during pre-training can help the network learn better features for dense prediction tasks as demonstrated in DenseCL. It will be interesting to explore the potential of RegionCL on these kinds of datasets since RegionCL is compatible with the dense prediction favored methods like DenseCL. However, since the region-swapping operation changes the overall semantics of the composite images, the proposed RegionCL method may not be suitable for cross-modality pre-training like CLIP [134] but could provide a better initialization for the image tower.

3.6 Conclusion

This paper demonstrates the importance of using both cropped and remained regions after cropping for self-supervised learning. A simple yet effective pretext task RegionCL is proposed to validate the models can learn better category feature representation from a complete perspective. Experimental results on image classification, object detection, and instance and semantic segmentation benchmarks demonstrate the effectiveness of leveraging remained regions in pretraining and its compatibility to representative self-supervised learning methods. We hope that this study will provide valuable insights into the subsequent studies of self-supervised learning in exploring region-based contrast methodology.

3.7 Appendix

In the appendix, we demonstrate the benefits of leveraging both cropped and remained regions for training with longer training epochs (3.7.1) and larger models (3.7.2). We also demonstrate the training efficiency of RegionCL in section 3.7.1. Pose estimation for both human and animals are also employed to evaluate the effectiveness of region-level contrastive learning on downstream tasks (3.7.3). Besides, we also independently feed the paste and canvas views

into the networks for training to further validate the performance of regional contrastive pairs without swapping (3.7.4). The analysis of parameters, architecture details, and implementation details are provided in 3.7.5, 3.7.6, and 3.7.7, respectively.

3.7.1 Training efficiency with longer training epochs

TABLE 3.11. Results of MoCo v2 [26] and RegionCL-M trained for 200, 400, and 800 epochs. '*' denotes that we end-to-end finetune RegionCL-M pretrained models for 50 epochs [56, 14].

	ImageNet Top-1			MS COCO AP^{bb}			MS COCO AP^{mk}		
Epochs	200	400	800	200	400	800	200	400	800
MoCo v2	67.5	69.6	71.1	40.9	41.3	41.5	37.0	37.5	37.6
RegionCL-M	69.4	72.1	73.1	41.6	41.9	42.1	37.7	38.0	38.2
RegionCL-M*	76.8	77.8	78.1	-	-	-	-	-	-

We investigate effectiveness of leveraging remained regions for longer pretraining by extending the epochs to 200, 400, and 800 respectively. We train MoCov2 [26] and the corresponding RegionCL-M respectively and present their results in Table 3.11. We evaluate these models' image classification performance on the ImageNet [40] dataset with linear probing. Their object detection and instance segmentation performance are also evaluated on the MS COCO [27] dataset with ResNet50-FPN and Mask-RCNN [59]. The detection and segmentation models are trained following the $2\times$ schedule, *i.e.*, the models are trained for 180K iterations in total.

As can be seen, with only 200 epochs for training, the proposed RegionCL-M obtains competitive results compared with MoCov2 trained for 400 epochs, no matter on classification or dense prediction tasks. The performance of RegionCL-M increases with the total training epochs increasing, and RegionCL-M can obtain better performance with less training cost, *i.e.*, RegionCL-M with 400 epochs pretraining has significantly outperformed MoCov2 trained with 800 epochs, confirming the good property of training efficiency brought by simply training with the regional contrastive pairs. Although RegionCL costs more forwards each iterations than the baseline method, it improves the overall training efficiency, *i.e.*, RegionCL with 400 epochs training (1200 forwards) beats the baseline methods with 800 epochs training (1600 forwards).

Besides, RegionCL-M sees an further improvement especially for classification (by 1% accuracy) when extending to 800 training epochs, reaching 73.1% Top-1 accuracy for classification, 42.1 AP for object detection and 38.2 AP for instance segmentation. Such observation demonstrates that the abundant contrastive pairs with both cropped and remained regions can not only improves the model's convergence but also effectively enhance the model's representation capacity with more training epochs.

3.7.2 Generalization ability for models with variant sizes

TABLE 3.12. Results of MoCo v2 [26] and RegionCL-M with R50 [61], R50w2 [61], and R50w4 [61] as backbones. '*' denotes that we end-to-end finetune RegionCL-M pretrained models for 50 epochs [56, 14].

	ImageNet Top-1			MS COCO AP^{bb}			MS COCO AP^{mk}		
Models	R50	w2	w4	R50	w2	w4	R50	w2	w4
MoCo v2	67.5	72.0	74.1	40.9	43.2	43.9	37.0	38.7	39.4
RegionCL-M	69.4	75.2	76.2	41.6	43.8	44.5	37.7	39.3	39.7
RegionCL-M*	76.8	79.7	80.4	-	-	-	-	-	-

To investigate the benefits of regional contrastive pairs on models with variant sizes, we adopt ResNet50 [61], ResNet50-w2 (2×parameters), and ResNet50-w4 (4×parameters) as backbone networks and train them for 200 epochs with MoCov2 and RegionCL-M, respectively. The results of linear probing on ImageNet along with object detection and instance segmentation on MS COCO are reported in Table 3.12. We use Mask-RCNN with ResNet50-FPN as the object detection and instance segmentation framework and train them for 180K iterations, following the 2× schedule. It can be observed that RegionCL-M with ResNet50-w2 outperforms MoCov2 with ResNet50-w4 on image classification and obtains competitive performance on both object detection and instance segmentation tasks on MS COCO. RegionCL-M with ResNet50-w4 obtains the best performance on all tasks. It indicates that the proposed RegionCL method is scalable to large models and can improve their performance on both classification and dense prediction tasks, further validating the importance of using both cropped and left regions in self-supervised learning.

TABLE 3.13. Results of RegionCL compatible models on human (MS COCO [27]) and animal (AP-10K [206]) pose estimation.

	MS COCO [27] AP	AP-10K [206] AP
Supervised	71.8	69.9
MoCo v2 [26]	72.0	70.1
RegionCL-M	72.3	70.6
DenseCL [178]	72.4	71.1
RegionCL-D	72.6	72.1
SimSiam [28]	71.9	70.5
RegionCL-S	72.2	71.6

3.7.3 Results on pose estimation

Besides the evaluation on detection and segmentation tasks, we also evaluate the models' performance on both human pose estimation and animal pose estimation tasks on MS COCO [27] and AP-10K [206] datasets. We adopt SimpleBaseline [184] as the base pose estimation framework and utilizes backbone models pretrained by MoCov2 [26], DenseCL [178], SimSiam [28], and their RegionCL compatible counterparts. We train these models for 210 epochs. We adopt an Adam optimizer with initial learning rate at 1e-4, which decreases by a factor of 10 at the 170 and 200 epochs respectively, following the same setting as in mmpose [34]. The results are available in Table 3.13. It can be observed that the SSL pretrained models outperforms the supervised counterpart. Besides, with both cropped and left regions taken into consideration, RegionCL improves the pretrained models' transfer performance on both pose estimation tasks, especially on the smaller animal pose dataset AP-10k. Such observation further demonstrates that exploiting supervisory signals from both instance and region levels can help the model obtain a better trade-off on both classification and dense prediction tasks.

3.7.4 Influence of the region swapping operation

TABLE 3.14. Influence of the region swapping strategy.

Configuration	ImageNet Top-1	MS COCO AP^{bb}
MoCov2	67.5	40.9
w/o swapping	69.2	41.4
w/ swapping	69.4	41.6

To further understand the performance gains brought by the abundant contrastive pairs, we adopt a simple RegionCL variant without the swapping operation, *i.e.*, simply cropping a region from candidate images as paste views and filling zeros into the cropped regions to formulate the canvas views. Using RegionCL-M as the base, we train RegionCL-M and its variant without the region swapping operation for 200 epochs, and evaluate their performance on the ImageNet [40] dataset and on the MS COCO [27] dataset with ResNet50-FPN and Mask-RCNN [59] following the 2× schedule. The results are available in Table 3.14. Without the region swapping operations, the proposed RegionCL still improves the model's performance on both classification and dense prediction tasks, while the region swapping operation can further improve the performance on both tasks, *e.g.*, 0.2% accuracy gains for ImageNet Top-1 accuracy and 0.2 mAP gains for object detection. It indicates that although taking both regions into consideration without swapping can facilitate the models learning, composing the hard negative samples, *i.e.*, the paste and canvas views via swapping, in the same images can further help the model learn better and discriminative feature representations from both instance- and region-level pairs, since features of these two kinds of views share some context from each other during network forward calculation.

3.7.5 Influence of different batch size

TABLE 3.15. The influence of batch size of MoCo v2 and RegionCL.

			ImageNet		MS COCO	
	Batch Size	LR	Top-1	Top-5	AP^{bb}	AP^{mk}
MoCo v2	256	0.03	67.5	-	40.9	37.0
MoCo v2	1024	0.15	67.5	88.2	41.0	37.2
RegionCL-M	256	0.03	70.0	90.0	41.6	37.8
RegionCL-M	1024	0.15	69.4	89.6	41.6	37.7

As the number of negative pairs plays an important role in the InfoNCE [125] loss and affects the pretrained model's performance as pointed in [24], MoCov2 maintains a huge memory queue to provide enough negative samples, which makes the calculation of the InfoNCE loss and the training process not coupled with the batch size. Thus we accelerate the training of MoCov2 [26] and RegionCL-M by increasing the batch size from 256 to 1,024. We train the models for 200 epochs with an initial learning rate 0.15 (around linear growth w.r.t. the

batch size) and a cosine learning rate scheduler, while the origin training setting is 200 epochs with an initial learning rate 0.03 and a cosine learning rate scheduler. The other settings are exactly the same, including the data augmentation strategies, optimizers, and the values of hyper-parameters. We validate the performance difference of the two training settings and present the results in Table 3.15. It can be observed that MoCov2's performance are consistent for both classification and dense prediction tasks with both training settings, confirming the rationality of the batch size to be 1,024 for MoCo v2. Thus, we choose such batch size for MoCo v2-based models in our paper.

We also conduct similar experiments for RegionCL-M as shown in the last two rows in Table 3.15. Similar conclusion can be observed in the evaluation of RegionCL-M with different batch size for training. Besides, as we adopt the batch-wise implementation for region swapping, RegionCL-M with small batch size and thus more iterations can see more diverse paste views and canvas views in terms of different sizes and locations, thus learning better feature representations. As a result, RegionCL-M with a batch size of 256 obtains slightly better performance for image classification by 0.6% Top-1 accuracy. Nevertheless, we choose the batch size of 1024 in this paper for acceleration purpose.

3.7.6 Architecture details

We present the details of the proposed RegionCL-M (MoCov2), RegionCL-D (DenseCL), and RegionCL-S (SimSiam) in this section. We also provide the pseudo codes for RegionCL-M, RegionCL-D, and RegionCL-S as in Algorithm 1, 2, and 3, respectively, with ***red*** color denoting the modifications of RegionCL compared with the base architecture.

RegionCL-D. As shown in Algorithm 2 and Figure 3.5, DenseCL [178] adopts both instance-level and pixel-level losses during pretraining. The modifications from DenseCL to RegionCL-D appears at the instance-level branch in a same way as the modifications from MoCov2 [26] to RegionCL-M, *i.e.*, we use the encoder, mask pooling, and the instance-level projector to extract the features belonging to the canvas and paste views, separately. Then, the contrastive

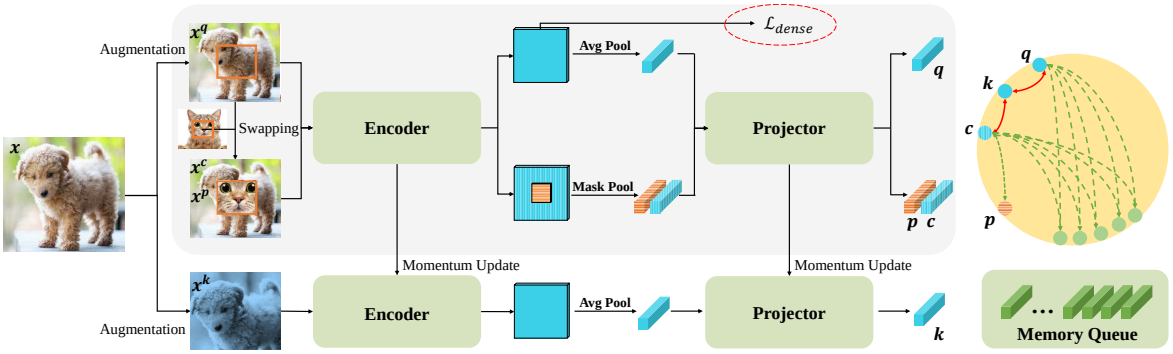


FIGURE 3.5. Illustration of the proposed RegionCL with the DenseCL framework, *i.e.*, **RegionCL-D**. Taken the instance-level views x^q and x^k, and the region-level views x^c and x^p as inputs, RegionCL-D firstly extract the instance- and region-level features using the same way as RegionCL-M. The extracted and projected features q, p, c, and k are constructed the contrastive pairs. The instance-level views x^q and x^k are also used to enhance dense feature correspondences before the average pooling operation, with another projector for pixel-wise feature projection and memory queue.

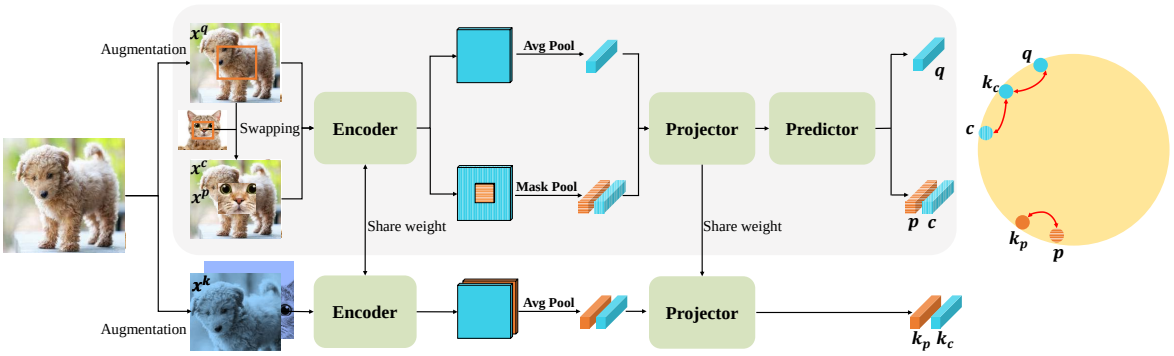


FIGURE 3.6. Illustration of the proposed RegionCL with the SimSiam framework, *i.e.*, **RegionCL-S**. Taking the two augmented views as inputs, RegionCL-S extracts the region-level features in the same way as RegionCL-M. The pooled region-level features are then batched with the instance-level features and processed by the projector. Following SimSiam, the projected features q, p, c, k build positive pairs.

pairs are also enriched by the regions while the dense correspondences related loss functions are remained the same as in DenseCL.

RegionCL-S. As shown in Algorithm 3 and Figure 3.6, SimSiam [28] does not require the negative pairs during pretraining and focuses on attracting the features among positive pairs. The modifications from SimSiam [28] to RegionCL-S are simply providing abundant positive

Algorithm 1 Example code of RegionCL-M.

Input: Two augmented views x^q, x^k
Input: The negative Queue Q
Output: The contrastive loss L

```
/* Feature extraction                                          */
// Online Branch
```
x_c^p = RegionSwapping(x^q)
q = Projector(AvgPool(Encoder(x^q)))
p,c = Projector(MaskPool(Encoder(x_c^p)))
```
// Momentum Branch
```
k = Projector$_M$(AvgPool(Encoder$_M$(x^k)))
```
/* Loss computation                                            */
// Eq. 1
```
$$L_{ins} = Loss(q, k|Q)$$
```
// Eq. 2
```
$$L_{reg} = (Loss(p, k|Q, sg(c)) + Loss(c, k|Q, sg(p)))/2$$
$$L = L_{ins} + L_{reg}$$

Algorithm 2 Example code of RegionCL-D.

Input: Two augmented views x^q, x^k
Input: The instance negative Queue Q_{ins}
Input: The dense negative Queue Q_{dense}
Output: The contrastive loss L

```
/* Feature extraction                                          */
// Online Branch
```
x_c^p = RegionSwapping(x^q)
q = Projector(AvgPool(Encoder(x^q)))
p,c = Projector(MaskPool(Encoder(x_c^p)))
```
// Extract dense feature
```
q_d = Projector$_d$(Encoder(x^q))
```
// Momentum Branch
```
k = Projector$_M$(AvgPool(Encoder$_M$(x^k)))
k_d = Projector$_{dM}$(Encoder$_M$(x^k))
```
/* Loss computation                                            */
// Eq. 1
```
$$L_{ins} = Loss(q, k|Q_{ins})$$
$$L_{dense} = DenseLoss(q_d, k_d|Q_{dense})$$
```
// Eq. 2
```
$$L_{reg} = (Loss(p, k|Q, sg(c)) + Loss(c, k|Q, sg(p)))/2$$
$$L = L_{ins} + L_{reg} + L_{dense}$$

pairs from both instance and region levels. Specifically, given the augmented views x^q, x^k and the composite images with the canvas view x^c and the paste view x^p as inputs, RegionCL-S adopts an encoder, average pooling (mask pooling) layer, a projector, and a predictor to get the instance-level (region-level) features q (p and c). The other view x^k are processed

Algorithm 3 Example code of RegionCL-S.

Input: Two augmented views x^q, x^k
Output: The contrastive loss L

```
/* Feature extraction                                                    */
// 1st Branch
```
x_c^p = RegionSwapping(x^q)
q = Predictor(Projector(AvgPool(Encoder(x^q))))
p,c = Predictor(Projector(MaskPool(Encoder(x_c^p))))
```
// 2nd Branch, weight sharing with the 1st Branch
```
k = Projector(AvgPool(Encoder(x^k)))
```
/* Loss computation                                                      */
// Eq. 1
```
$$L_{ins} = Loss(q, sg(k))$$
```
// Eq. 2
```
$$L_{reg} = (Loss(p, sg(k)) + Loss(c, sg(k))/2$$
$$L = L_{ins} + L_{reg}$$

by the weight-shared encoder and projector to get the feature k, where an stop gradient operation is applied on k to stabilize the training. Thus, there are three cases of positive pairs in the modified RegionCL-S, *i.e.*, the instance-level pairs q and k as in origin SimSiam, the region-level pairs c and k_c, and p and k_p, and we keep the learning objectives and architecture the same as in SimSiam.

3.7.7 Implementation details

In this section we give the implementation details of all the three RegionCL models, *i.e.*, RegionCL-M (MoCov2 [26]), RegionCL-D (DenseCL [178]), and RegionCL-S (SimSiam [28]), respectively.

3.7.7.1 RegionCL-M and RegionCL-D

- **Training settings.** To accelerate the training, we train the RegionCL-M and RegionCL-D with a total batch size of 1,024 and initial learning rate 0.15, which is slightly different from the original setting but does not affect the performance and our conclusion as shown in Sec 3.7.5. The other settings are the same as in MoCov2 [26] and DenseCL [178]. For example, the input images are first randomly cropped and resized to 224 $\times$ 224 by remaining 20% $\rightarrow$ 100% regions, which is

followed by color jitter with a probability of 0.8, grayscale with a probability of 0.2, Gaussian blur with a probability of 0.5, and random horizontal flips, which are the same as the origin settings in the two base methods. The SGD [155] optimizer is adopted with weight decay at 1e-4 and momentum at 0.9.

- **Architecture settings.** There is no difference in the model's architectures comparing the base models and RegionCL models. The instance-level and region-level features share the same projector, which has the structure of *Linear(2048, 2048)* $\rightarrow$ *ReLU* $\rightarrow$ *Linear(2048, 128)*, where 2048 is the hidden dimension and 128 is the output dimension. The dense correspondences projectors in DenseCL and RegionCL-D have the structure of *Conv2d(2048, 2048)* $\rightarrow$ *ReLU* $\rightarrow$ *Conv2d(2048, 128)*, where the kernel size for the convolutions is 1×1, and 128 is the output dimension. The length of the memory queue is 65,536.

3.7.7.2 RegionCL-S

- **Training settings.** As there is no component like memory queues in SimSiam's architecture, we follow exactly the same training settings as in origin SimSiam to train the RegionCL model RegionCL-S. Specifically, a total batch size of 512 is employed during the pretraining process. The SGD optimizer with learning rate 0.1, weight decay 1e-4, and momentum 0.9 is adopted to train the model. The data augmentation strategy is the same as the in MoCov2 [26] and DenseCL [178], *i.e.*, the input images are first randomly cropped and resized to 224×224 by remaining $20\% \rightarrow 100\%$ regions, which is followed by color jitter with a probability of 0.8, grayscale with a probability of 0.2, Gaussian blur with a probability of 0.5, and random horizontal flips. The models are trained for 100 epochs with cosine learning rate scheduler.
- **Architecture settings.** There is no modification on the model's architectures settings comparing the SimSiam and RegionCL-S. The projection head follows average pooling or mask pooling and has 3 layers to project the features, which takes the form of $Linear(2048, 2048) \rightarrow BN(2048) \rightarrow ReLU \rightarrow Linear(2048, 2048) \rightarrow BN(2048) \rightarrow ReLU \rightarrow Linear(2048, 2048) \rightarrow BN(2048)$. The predictor head

has 2 layers to further process the features and align them with the features extracted from the key views. The structure of the predictor head is $Linear(2048, 512) \rightarrow BN(512) \rightarrow ReLU \rightarrow Linear(512, 2048)$. These structures are the same as the original settings in SimSiam.

CHAPTER 4

ViTPose++: Vision Transformer for Generic Body Pose Estimation

4.1 Introduction

With these pre-trained models, it is essential to evaluate their performance on downstream vision tasks. This chapter will focus on the Body keypoint detection (a.k.a. body pose estimation of human, animal, etc.) task, which is one of the fundamental tasks in computer vision and has a wide range of real-world applications [216, 101, 87]. As a representative example, human pose estimation aims to localize human anatomical keypoints and is challenging due to the variations of occlusion, truncation, scales, and human appearances. To deal with these challenges, there has been rapid progress in deep learning-based methods [165, 184, 137, 225], which are typically built upon convolutional neural networks (CNN).

Recently, vision transformers [45, 109, 16, 152] have shown great potential in many computer vision tasks. Inspired by their success, different vision transformer structures have been explored for human pose estimation. Most of them adopt CNN as the backbone and then use a transformer of elaborate structures to refine the extracted features and model the relationship between the body keypoints. For example, PRTR [90] incorporates both transformer encoders and decoders to gradually refine the locations of the estimated keypoints in a cascade manner. TokenPose [94] and TransPose [201], instead, adopt an encoder-only transformer structure to further process the features extracted by the CNN backbone. Inspired by the success of anchor-based methods in detection [16], a query-based decoder is adopted in Poseur [112] with noisy augmentations. On the other hand, HRFormer [211] employs the transformer to directly extract features and introduce high-resolution representations via multi-resolution parallel transformer modules. These methods have obtained superior performance on human

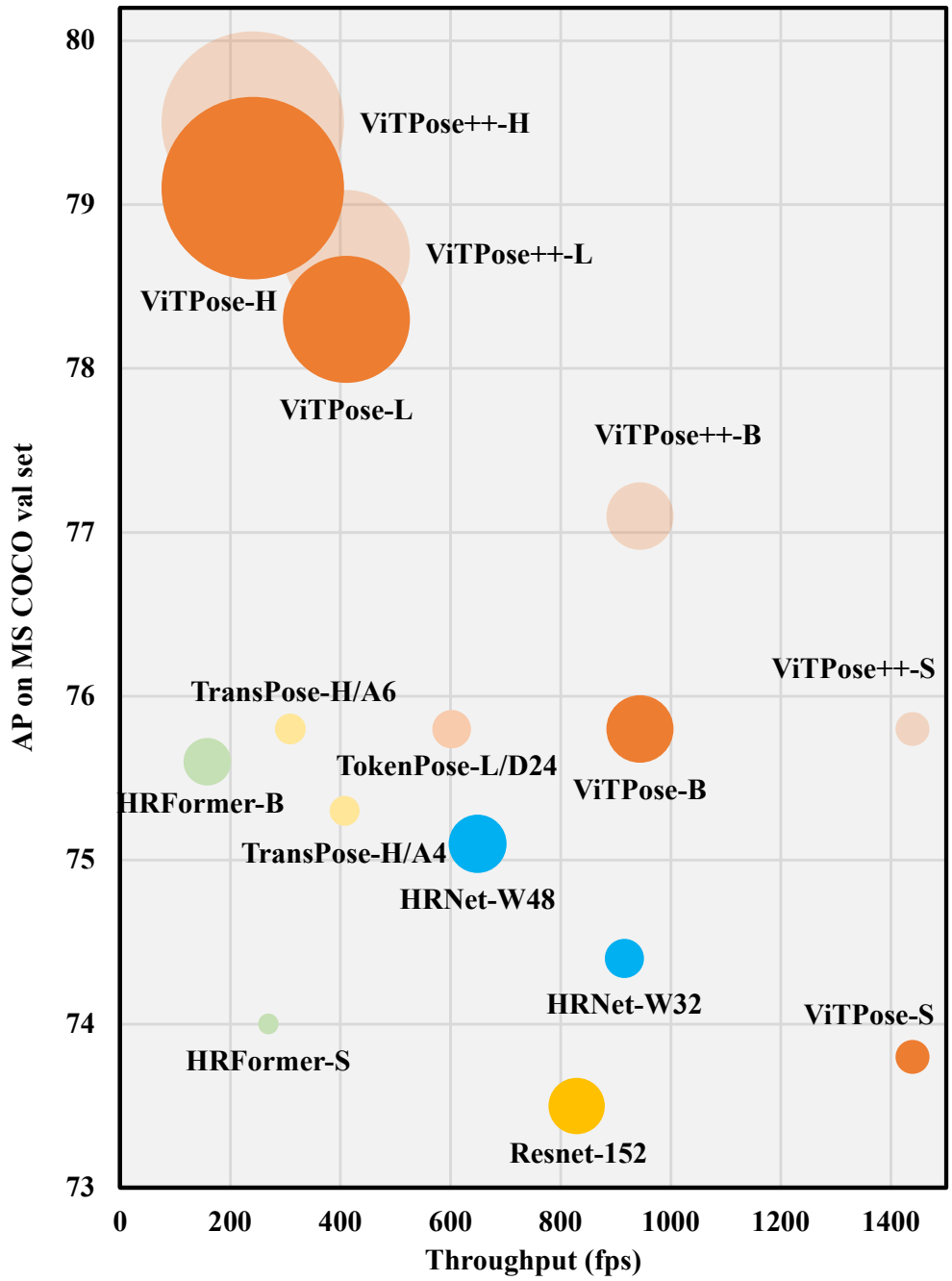


FIGURE 4.1. Comparison of ViTPose and SOTA methods on the MS COCO val set regarding model size, throughput, and accuracy. The size of each bubble represents the number of model parameters.

pose estimation. However, they either need an extra CNN backbone for feature extraction or adopt transformer structures specially designed for pose estimation. Besides, most of them

favor body pose estimation only on a single type, *i.e.*, human, while ignoring the potential of transformers in modeling the correlations between different types of body keypoints (*e.g.*, human and animal), which is essential for developing a foundation model for generic body keypoint detection. This motivates us to think from a different direction, *how well can plain vision transformers do for body pose estimation towards a foundation model?*

To find the answer to this question, we propose a simple baseline model dubbed **ViTPose** and demonstrate the potential of simple plain vision transformers for human pose estimation in both the top-down and bottom-up manner. Then, a novel **ViTPose++** model is proposed to deal with multiple types of body keypoint detection via knowledge factorization. Specifically, ViTPose employs a plain and non-hierarchical vision transformer [45] as the backbone to extract features for an input person instance or image, determined by the adopted pose estimation paradigm, *i.e.*, either the top-down paradigm or the bottom-up paradigm. The backbone is pre-trained with masked image modeling pretext tasks, *e.g.*, MAE [57], to provide a good initialization. Then, a lightweight decoder, which is composed of two deconvolution layers and one prediction layer, further processes the extracted features by up-sampling and regression to get the heatmaps of different keypoints. It is noteworthy that the associate maps are also regressed together with the heatmaps for the bottom-up paradigm. Despite no elaborate task-specific designs, ViTPose achieves a state-of-the-art (SOTA) performance of 80.9 AP on the challenging MS COCO test-dev set for human keypoint detection. ViTPose++ adopts the idea of the mixture of experts (MoE) in the backbone networks to factorize the knowledge between different types of body keypoints[1], considering that different body pose estimation tasks may share common knowledge of body keypoints while also requiring task-specific knowledge. In detail, ViTPose++ decomposes the feed-forward networks (FFN) into shared and task-specific parts to encode the common and task-specific features, respectively. In this way, ViTPose++ effectively addresses the task conflict issue and thus achieves better performance on each task than the single-task and naive multi-task learning methods. It also sets new SOTA on four challenging benchmarks, including MS COCO [100], OCHuman [101], MPII [4], and AP-10K [206]. Besides, ViTPose++ retains the inference speed as ViTPose since no more parameters and computations are introduced for each task.

[1]We treat each type of body keypoint detection as an individual task.

aBesides the superior performance, we also show the surprisingly good properties of ViTPose from various aspects, namely simplicity, scalability, flexibility, and transferability. 1) For simplicity, thanks to the strong feature representation ability of vision transformers, the ViTPose pipeline can be extremely simple. For example, it does not require any specific domain knowledge to design the backbone encoder and enjoys a plain and non-hierarchical encoder structure by simply stacking several transformer layers. The decoder can be further simplified to a single bilinear up-sampling layer followed by a common convolutional prediction layer with a negligible performance drop. This structural simplicity makes ViTPose enjoy better computational parallelism so that it reaches a new Pareto front for inference speed and performance, as shown in Fig. 4.1. 2) In addition, the simplicity in structure brings the excellent scalability of ViTPose, which can benefit from the rapid development of scalable pre-trained vision transformers. Specifically, one can easily control the model size by stacking different numbers of transformer layers and setting different feature dimensions (*e.g.*, using ViT-S, ViT-B, ViT-L, or ViT-H) to balance the inference speed and performance for various deployment requirements. 3) Furthermore, we demonstrate that ViTPose is very flexible in the training paradigm. It can adapt well to higher input and feature resolutions with minor modifications and deliver better pose estimation results. In addition, ViTPose can obtain competitive performance even if it is pre-trained using smaller unlabelled datasets or fine-tuned with the attention modules frozen at a lower training cost. 4) Last but not least, we show that the performance of small ViTPose models can be easily improved by transferring the knowledge from large ViTPose models through an extra learnable knowledge token, demonstrating a good transferability of ViTPose.

In summary, the main contribution of this paper is threefold. 1) We propose simple yet effective baseline models named ViTPose for human pose estimation and ViTPose++ for generic body keypoint detection. It obtains SOTA performance on representative benchmarks without elaborate designs for the backbone and pipeline. 2) The simple ViTPose model demonstrates to have surprisingly good properties, including structural simplicity, model size scalability, training paradigm flexibility, and knowledge transferability. These properties can shed light on the future development of vision transformer-based methods in the field of pose estimation. 3) Comprehensive experiments on challenging public benchmarks are conducted

to evaluate and analyze the performance of ViTPose and ViTPose++, which set new SOTA on the benchmarks for different body keypoint detection tasks, making a solid step towards developing a foundation model for generic (body) keypoint detection.

4.2 Related Work

4.2.1 Representative pose estimation methods

4.2.1.1 Top-down methods

Pose estimation has undergone rapid development, transitioning from CNN-based approaches [184, 138] to the utilization of vision transformers [45, 211, 217]. The majority of existing methods concentrate on estimating poses from provided human instances, following a top-down pipeline. Notably, G-MRI [126] introduced Faster-RCNN for generating bounding boxes and proposed a combined regression and classification task for keypoint estimation. RMPE [48] further employed a spatial transformer network to rectify noise in detected bounding boxes and facilitate subsequent pose estimation. To achieve accurate keypoint localization, some approaches employed high-resolution features in CNN networks through techniques such as skip feature concatenation [119] or highway structures [154]. CPN [30] introduced a two-stage network to refine estimated keypoints using hierarchical features. Other methods like OKDHP [96] leveraged multiple-branch networks and dynamic fusion strategies to utilize multi-scale information effectively. AlphaPose [47] improved performance by leveraging a symmetric keypoint localization technique. In contrast, SimpleBaseline [184] directly recovered high-resolution features with decoders. As vision transformers have demonstrated superior performance across various vision tasks [118, 92], some approaches explored the use of transformers as decoders after the CNN backbone [90, 94, 201]. For instance, TransPose [201] processed features extracted by CNN to model global relationships, while TokenPose [94] introduced token-based representations and extra tokens to estimate the locations of occluded keypoints and model the relationship among keypoints. Poseur [112] explored an anchor-based strategy with noise augmentation using transformer decoders. These

transformer-based pose estimation methods have achieved superior performance on popular human pose estimation benchmarks. However, these methods still rely on CNN backbones for feature extraction. In contrast, HRFormer [211] utilized transformers to directly extract high-resolution features. It incorporated a parallel transformer module to gradually fuse multi-resolution features and achieved outstanding performance. Nevertheless, the extent to which attention mechanisms contribute to pose estimation remains unclear and few efforts have explored the potential of plain vision transformers for pose estimation. In this paper, we fill this gap by proposing a simple yet effective baseline model dubbed ViTPose based on the plain vision transformers.

4.2.1.2 Bottom-up methods

In contrast to top-down methods that operate on individual human instances, bottom-up methods directly localize the keypoints of all humans present in the input images. DeepCut [133] and DeeperCut [71] employ regression to estimate all keypoints and subsequently group them into separate individuals. OpenPose [15] adopts a two-branch network, with one branch dedicated to keypoint estimation and the other to grouping. Associate embedding [117] simplifies the bottom-up pipeline by jointly estimating keypoint locations and associating them within the same network. Pifpaf [78] employs a Part Intensity Field and a Part Association Field to regress and associate body keypoints for all humans. SPM [121] simplifies the pipeline by directly predicting the root joint for each human and the displacement between the root joint and other keypoints based on a structured pose representation. To enhance the performance of bottom-up methods, HigherHRNet [31] introduces multi-stage supervision. In contrast, we embark on the pioneering exploration of plain vision transformers for bottom-up pose estimation, which shows promising performance.

4.2.2 Vision transformer pre-training

Inspired by the success of ViT [45], many different vision transformers [109, 196, 176, 229, 175, 217, 173, 218] have been proposed, which are typically pre-trained on the ImageNet [40] dataset at the fully supervised setting. However, the fully-supervised learning paradigm needs

a large scale of labeled data, which is expensive. Besides, such pre-trained models may not generalize well on the tasks with a very different data distribution as ImageNet. To provide a better initialization for the vision transformers, self-supervised pre-training methods have been proposed, either in the contrastive way [29, 197] or the generative way [57, 6]. A typical example is MAE [57] that adopts a masked image modeling (MIM) pretext task inspired by the success of masked language modeling pretext tasks in natural language processing. Surprisingly, MIM pre-trained models on ImageNet without using the labels demonstrate better generalization performance on image classification and downstream tasks. In this paper, we focus on pose estimation tasks and adopt plain vision transformers with MIM pre-training as backbones. Besides, we explore whether or not using ImageNet for pre-training is necessary for pose estimation tasks. Surprisingly, using smaller unlabelled pose datasets for pre-training can also provide a good initialization.

4.2.3 Foundation models

Foundation model [10], with the aim to deal with diverse real-world problems with a unified model, has received increasing attention recently. Owing to their strong representation and scalability abilities, vision transformers have become popular in building foundation models and leveraging massive data from multiple modalities for training. For example, Florence [210] employs different adapter networks on the pre-trained backbones to handle many tasks, including classification, detection, and segmentation. Encoder-decoder structures are adopted in [3] to solve multi-modality tasks via a novel Perceiver module. To further enhance the representation ability of backbone networks, MoE [146] is adopted by allowing the network to ensemble the outputs from different sub-networks without introducing too much extra computational costs. Typically, each MoE layer contains a gate layer to determine the appropriate expert to use and fuse the outputs of the selected experts in a weighted sum manner, *e.g.*, UniPerceiver-MoE [231] adopts MoE-based attention layers and FFN layers to construct the networks. In this paper, we aim to explore the potential of plain vision transformers for generic body keypoint detection. We propose ViTPose++, which decomposes the FFN layers into task-agnostic and task-specific experts to handle different types of body

pose estimation tasks simultaneously. Without extra parameters and computations, ViTPose++ outperforms ViTPose trained in either a single-task or multi-task manner and sets new SOTA on challenging public benchmarks.

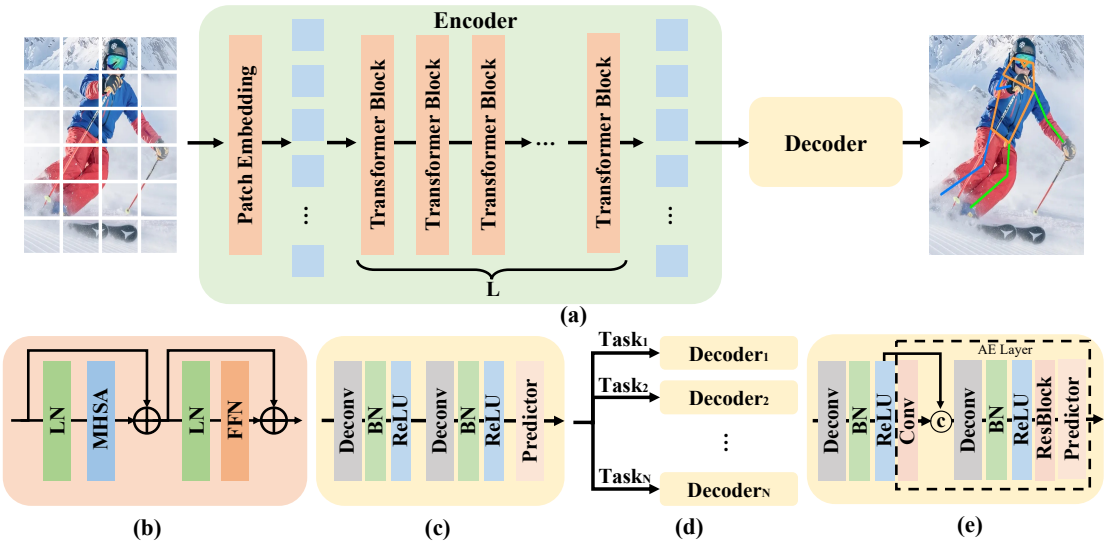


FIGURE 4.2. (a) The diagram of the proposed ViTPose model. (b) The transformer block. (c) The classic decoder in the top-down paradigm. (d) The multi-task decoder used to deal with multiple datasets. (e) The decoder in the bottom-up paradigm.

4.2.4 Comparison to the conference version

A preliminary version of this paper is presented in [195]. This paper extends the previous study with three major improvements. **1)** We explore more model sizes of ViTPose to thoroughly compare with representative models from many aspects, *i.e.*, number of parameters, inference speed, input resolution, and performance. For example, the ViTPose-S model with 24M parameters obtains 73.8 AP with 1,432 fps on MS COCO, which is better than ResNet-50 [184] (71.8 AP, 1,351 fps) and ResNet-152 [184] (73.5 AP, 829 fps) and comparable with the frontier transformer-based model HRFormer-S [211] (73.8 AP, 269 fps) but with a much faster speed. As demonstrated in Fig. 4.1, different ViTPose variants from the small to large model sizes set a new Pareto front for throughput and performance, demonstrating the superiority of plain vision transformers for human pose estimation. **2)** We extend ViTPose to the bottom-up paradigm from the top-down paradigm by directly predicting the keypoint

locations and their associations. The experimental results on the MS COCO dataset again demonstrate the potential and flexibility of vision transformers, *i.e.*, plain vision transformers can perform well for human pose estimation in both top-down and bottom-up manners. **3)** A novel model dubbed ViTPose++ is further proposed to deal with multiple types of body pose estimation tasks via knowledge factorization. It obtains better performance than previous representative methods on public challenging datasets, *i.e.*, MS COCO [100], OCHuman [222], MPII [4], AI Challenger (AIC) [182], COCO-Wholebody (COCO-W) [73], AP-10K [206], and APT-36K [202], for various types of body keypoint detection, making the first attempt towards developing a foundation model for generic (body) keypoint detection. Besides, more experiment results, ablation studies, and analyses are presented. We also provide some visual results to demonstrate the promising performance of ViTPose++.

4.3 ViTPose

As shown in Fig. 4.2, ViTPose employs a plain vision transformer for feature extraction and is compatible with different decoders for keypoint estimation, *e.g.*, using classic (c) or simple (d) decoder in top-down keypoint estimation and associate embedding layer (e) in a bottom-up manner. Different properties of ViTPose are illustrated, *i.e.*, simplicity, scalability, flexibility, and transferability. By exploiting these properties, a novel ViTPose++ model is further proposed and obtains SOTA performance on various pose estimation datasets. The details will be described below.

4.3.1 The simplicity of ViTPose

The goal of this paper is to provide a simple yet effective vision transformer baseline for body pose estimation tasks and explore the potential of plain and non-hierarchical vision transformers [45]. Thus, we keep the structure as simple as possible and try to avoid fancy but complex modules, even though they may improve performance. To this end, we simply append several decoder layers after the transformer backbone to regress the heatmaps of keypoints, as shown in Fig. 4.2(a). Specifically, given a person instance image $X \in \mathcal{R}^{H \times W \times 3}$,

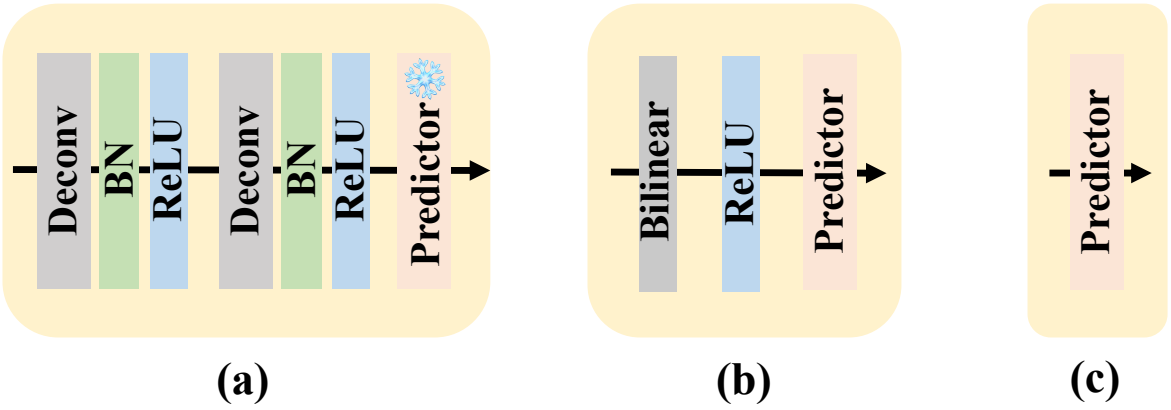


FIGURE 4.3. (a) Classic decoder (Fig. 4.2(c)) with a frozen predictor. (b) Simple decoder. (c) Minimal decoder.

it is first embedded into tokens via a patch embedding layer, *i.e.*, $F \in \mathcal{R}^{\frac{H}{d} \times \frac{W}{d} \times C}$, where H and W are the height and width of the input image, respectively, d (*e.g.*, 16 by default) is the down-sampling ratio of the patch embedding layer, and C is the channel dimension. Then, the tokens are processed by several transformer layers, each of which is consisted of a multi-head self-attention (MHSA) layer and an FFN, *i.e.*,

$$\begin{aligned} F'_{i+1} &= F_i + \mathrm{MHSA}(\mathrm{LN}(F_i)), \\ F_{i+1} &= F'_{i+1} + \mathrm{FFN}(\mathrm{LN}(F'_{i+1})), \end{aligned} \tag{4.1}$$

where F_i denotes the output of the ith transformer layer and we use $F_0 = \mathrm{PatchEmbed}(X)$ to denote the features after the patch embedding layer. It should be noted that the spatial and channel dimensions are constant for each transformer layer. We denote the output feature of the backbone network as $F_{out} \in \mathcal{R}^{\frac{H}{d} \times \frac{W}{d} \times C}$.

We adopt different decoder designs to process the features extracted from the backbone network and regress the heatmaps, *i.e.*, the classic decoder, the simple decoder, and the minimal decoder, as shown in Fig. 4.2(c) and Fig. 4.3. The classic decoder is composed of two deconvolution blocks, each containing one deconvolution layer followed by batch normalization [72] and ReLU [2]. Following the common settings in previous methods [184, 215], each deconvolution blocks first up-samples the feature maps by a factor of 2. Then, an

1×1 convolution prediction layer is used to regress the heatmaps, *i.e.*,

$$K = \text{Conv}_{1\times1}(\text{Deconv}(\text{Deconv}(F_{out}))), \tag{4.2}$$

where $K \in \mathcal{R}^{\frac{H}{4}\times\frac{W}{4}\times N_k}$ denotes the heatmaps of keypoints and N_k is the number of keypoints, *e.g.*, 17 for the MS COCO dataset.

In addition to the classic decoder, we propose three simplified designs: classic-FP, simple decoder, and minimal decoder, to leverage the strong representation capabilities of the vision transformer backbone. Classic-FP retains the same structure as the classic decoder but with a randomly initialized and frozen predictor layer. For the simple decoder as shown in Fig. 4.3(b), the feature maps extracted by the backbone are upsampled by a factor of 4 using bilinear interpolation and then fed into a ReLU activation function and a 3×3 convolutional layer to predict the heatmaps, *i.e.*,

$$K = \text{Conv}_{3\times3}(\text{Bilinear}(\text{ReLU}(F_{out}))). \tag{4.3}$$

The minimal decoder is the most simple design, which utilizes only a single linear projection layer for heatmap prediction, *i.e.*,

$$\begin{aligned} K' &= \text{Conv}_{1\times1}(F_{out}) \in \mathcal{R}^{\frac{H}{16}\times\frac{W}{16}\times(16N_k)}, \\ K &= \text{PixelUnshuffle}(K') \in \mathcal{R}^{\frac{H}{4}\times\frac{W}{4}\times N_k}, \end{aligned} \tag{4.4}$$

where PixelUnshuffle [148] refers to the efficient operator that restores the spatial dimensions of the heatmap by reshaping it from the channel dimension to the spatial dimension. This design choice allows for a focused investigation into the representation capability of vision transformers in pose estimation

4.3.2 The scalability of ViTPose

While small-scale CNN-based models have shown promising results, they often face challenges when scaling up due to the overfitting problem caused by a strong inductive bias, as discussed in [164]. One intriguing characteristic of vision transformers is their improved

performance as the model size increases [45]. However, it remains under-explored whether this scalability holds true for specific downstream tasks such as pose estimation. Based on ViTPose, we can easily control its model size by adjusting the number of stacked transformer layers and feature dimensions. This allows us to develop different sizes of models and fully exploit the benefits of pre-trained plain vision transformers. Specifically, we use the vision transformers of different model sizes as the backbone, *i.e.*, ViT-B, ViT-L, ViT-H [45], and ViTAE-G [217], which are MIM pre-trained on ImageNet, and fine-tune them on the MS COCO dataset to investigate the scalability of ViTPose. For ViT-H and ViTAE-G, which use patch embedding with size 14×14 during pre-training, we use zero padding to reformulate a patch embedding with size 16×16 for the same fine-tuning setting with ViT-B and ViT-L. As shown in Fig. 4.1, ViTPose delivers continuing performance gains with the increased model size.

4.3.3 The flexibility of ViTPose

Pre-training data. Pre-training the backbone networks on ImageNet [40] has been a *de facto* routine for a good initialization of network parameters. However, this approach necessitates the use of large amounts of additional data beyond the pose data, thereby imposing higher requirements, particularly for vision transformers employed in pose estimation tasks. This motivates us to investigate whether it is possible to alleviate the data requirements for vision transformers by exclusively utilizing pose data throughout the entire training phase. To answer this question, apart from the default setting of ImageNet [40] pre-training, we also use MAE pre-training on MS COCO [100] and a combination of MS COCO and AIC [182], respectively, by random masking 75% patches from the images and reconstructing those masked patches. Then, we use the pre-trained weights to initialize the backbone of ViTPose and fine-tune it on the MS COCO dataset. Surprisingly, although the volume of the pose data is much smaller than ImageNet, ViTPose trained only with pose data can obtain competitive performance, demonstrating its flexibility regarding the pre-training data.

Resolution of images and features. We vary the input image size and down-sampling ratios d of ViTPose to evaluate its flexibility regarding the input and feature resolution. Specifically,

to adapt ViTPose to input images at a higher resolution, we simply resize the input images and train the model on them accordingly. Besides, to adapt the model to lower down-sampling ratios, *i.e.*, higher resolution of feature maps, we simply change the stride of the patch embedding layer to partition tokens with overlap while keeping the size of each patch. We show that the performance of ViTPose increases consistently regarding either higher input resolution or higher feature resolution.

Attention type. Using full attention on high-resolution feature maps will cause a huge memory footprint and computational cost due to the quadratic computational complexity of vanilla self-attention. Window-based attention with relative position embedding [92, 93] have been explored to address this issue and can be used to deal with high-resolution feature maps. However, simply using window-based attention for all transformer blocks degrades the performance due to the lack of global context modeling ability. To address the problem, we adopt two techniques. *i.e.*, 1) *Shift window:* Instead of using fixed windows for attention calculation, we use the shifted window mechanism [109] to help broadcast the information between adjacent windows. 2) *Pooling window:* Apart from the shifted window, we try another solution via pooling. Specifically, we first extract the global token for each window by average pooling all the tokens in the window. The global tokens from all windows are then appended to the tokens in each window to serve as key and value tokens for attention calculation, thus enabling cross-window information exchange. In addition, we further show that the two techniques are complementary and can work together to improve performance and reduce memory footprint without introducing extra parameters.

Fine-tuning strategy. As demonstrated in NLP fields [104, 3], pre-trained transformer models can generalize well to other tasks by only tuning partial parameters. To investigate whether it still holds for vision transformers, we fine-tune ViTPose on MS COCO with all parameters unfrozen, MHSA frozen, and FFN frozen, respectively. We empirically demonstrate that with the MHSA frozen, ViTPose obtains comparable performance to the fully fine-tuning setting while using less memory footprint.

Keypoint detection paradigm. Although some works [211, 94, 112] have explored vision transformers for body keypoint detection in the top-down paradigm, it still needs to be

determined how well the vision transformers, especially the plain vision transformers, can perform in the bottom-up paradigm. To this end, we thoroughly explore the potential of plain vision transformers in both top-down and bottom-up paradigms. Technically, in the top-down paradigm, the individual human instance is fed into the backbone network for feature extraction. In contrast, the whole image with several human instances is used as input to the backbone network in the bottom-up paradigm. The associate embedding technique [117] is adopted to predict the keypoints and tag vectors based on the extracted features. Instead of using 1/16 feature resolution and vanilla self-attention in the top-down paradigm, we adopt 1/8 feature resolution and window-based attention in the bottom-up paradigm. The decoder of HigherHRNet [31] is adopted with an extra up-sample layer as shown in Fig. 4.2 (d), *i.e.*,

$$K = AE(Deconv(F_{out})), \tag{4.5}$$

where AE represents the associate embedding layer used to predict the tag vectors and the keypoint locations.

4.3.4 The transferability of ViTPose

One common method to improve the performance of small models is to transfer the knowledge from larger ones, *i.e.*, knowledge distillation [66, 50]. Some interesting distillation methods [96] have been developed to enhance the performance of small-scale models in pose estimation tasks. In our study, we show that even a very simple distillation method can effectively improve the performance of ViTPose. Specifically, given a teacher network T and a student network S, a simple distillation method is to add an output distillation loss $L^{od}_{t\to s}$ to force the student network's output imitating the teacher network's output, *e.g.*,

$$L^{od}_{t\to s} = \text{MSE}(K_s, K_t), \tag{4.6}$$

where K_s and K_t are the outputs (*e.g.*, heatmaps) from the student and teacher network given the same input.

Transformers exhibit a distinct property of flexibility in handling inputs of various sizes. For instance, it has been shown that attaching several learnable prompt tokens to the inputs

can improve transformers' performance on specific tasks [106]. These learnable tokens effectively capture task-related information from the training data, aiding in the optimization of transformers. Motivated by this observation, we propose a simple token-based distillation method to bridge the gap between large and small models. It complements the heatmap-based distillation methods and further enhances the performance of the models. Specifically, we randomly initialize an extra learnable knowledge token t and append it to the visual tokens after the patch embedding layer of the teacher model. Then, we freeze the well-trained teacher model and only update the knowledge token, *i.e.*,

$$t^* = \arg\min_t(\text{MSE}(T(\{t; X\}), K_{gt}), \tag{4.7}$$

where K_{gt} is the ground truth heatmaps, X is the input images, $T(\{t; X\})$ denotes the predictions of the teacher, and t^* represents the optimal token that minimizes the loss. Then, the knowledge token t^* is frozen and concatenated with the visual tokens in the student network during training, thus transferring the knowledge from teacher to student. The loss for the student network is:

$$L^{td}_{t\to s} = \text{MSE}(S(\{t^*; X\}), K_{gt}), \tag{4.8}$$

$$L^{tod}_{t\to s} = \text{MSE}(S(\{t^*; X\}), K_t) + \text{MSE}(S(\{t^*; X\}), K_{gt}), \tag{4.9}$$

where $L^{td}_{t\to s}$ and $L^{tod}_{t\to s}$ denote the token distillation loss and its combination with the output distillation loss, respectively.

4.3.5 ViTPose++

A basic requirement for generic body keypoint detection should be able to deal with different body pose estimation tasks after training on multiple datasets with heterogeneous categories of body keypoint annotations. One critical challenge is how to deal with the differences of body keypoints in different pose estimation tasks, *e.g.*, the same keypoint (e.g., nose) of humans and animals with distinct appearances, and the different categories of keypoints in COCO-W [73] which are absent in MS COCO [100] and MPII [4]. Moreover, the data distribution of different species is also different, *e.g.*, the human head is always above the shoulder, while the cow's head is always to the left or right of the shoulder.

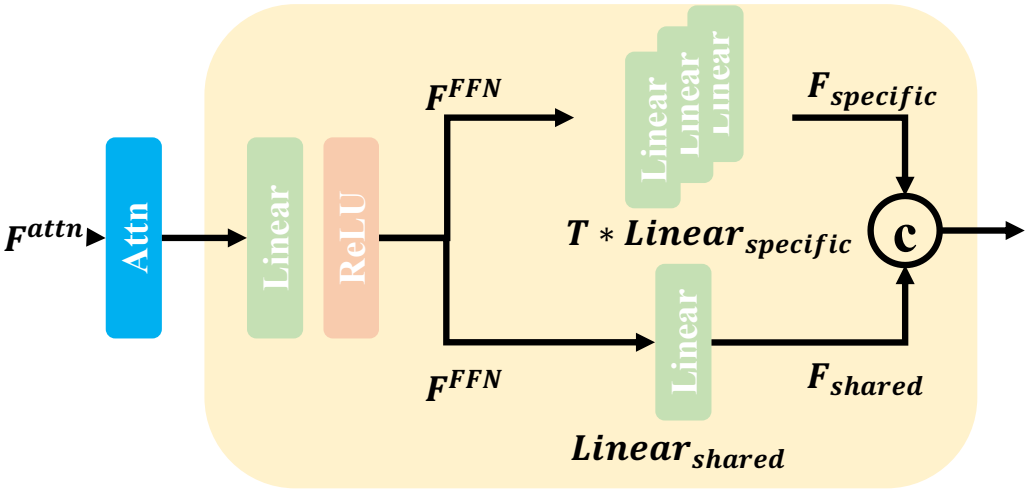


FIGURE 4.4. The structure of FFN in the proposed ViTPose++ model. T is the number of tasks processed by the model.

A naive solution is to train a ViTPose model by multi-task learning, *i.e.*, using a shared backbone and different decoders, each of which is responsible for a specific type of pose estimation task. However, there may be conflicts between different tasks [103], affecting learning performance. In this paper, we propose a novel model dubbed ViTPose++ to address the challenge from the perspective of knowledge factorization. Specifically, since MHSA layers are not sensitive to pose estimation tasks (as evidenced by the experiment results of fine-tuning strategy in Section 4.4.3), we adopt the idea of MoE [146], *i.e.*, splitting FFN layers into a task-agnostic expert and multiple task-specific experts to encode the common and task-specific knowledge for pose estimation, respectively. Similar to the aforementioned naive multi-task learning method, we use a task-specific decoder for each type of body pose estimation task. Technically, we take one transformer block as an example to illustrate the proposed ViTPose++. As shown in Fig. 4.4, given the output feature F^{attn} of MHSA, it is processed by the first linear layer of FFN, which is shared by the MoE, *i.e.*,

$$F^{FFN} = ReLU(Linear(F^{attn})). \tag{4.10}$$

Then, the output feature $F^{FFN} \in \mathcal{R}^{N \times \gamma C}$ is fed into separate linear layers (*i.e.*, task-agnostic expert and task-specific expert), where N represents the number of tokens and γ is the expansion ratio of FFN, which is set to 4 by default. The two kinds of experts project F^{FFN}

to F^{shared} and $F^{specific}$ with a channel dimension of $(1-\alpha)C$ and αC, respectively, *i.e.*,

$$\begin{aligned} F^{shared} &= Linear_{shared}^{\gamma C \to (1-\alpha)C}(F^{FFN}), \\ F^{specific} &= Linear_{specific}^{\gamma C \to \alpha C}(F^{FFN}), \end{aligned} \tag{4.11}$$

where α is the partition ratio used to balance the shared and task-specific experts and is set to 0.25 by default. Note that the parameters of the shared expert are trained using all the data, while the parameters of the task-specific expert are trained only using the data for the corresponding task. Then, F^{shared} and $F^{specific}$ are concatenated along the channel dimension to form the output of the transformer block. Given an input image from the training set for a specific pose estimation task, after getting the encoded feature from the transformer backbone described above, it is fed into the corresponding decoder to regress the heatmaps.

During inference for each type of pose estimation task, the shared and task-specific linear layers are merged into a single layer for parallel computation. In this way, ViTPose++ brings no extra parameters and computational costs compared with the ViTPose model while being able to serve as a foundation model for generic body pose estimation.

TABLE 4.1. Hyper-parameter settings for training ViTPose on the MS COCO dataset and the multiple datasets. The hyper-parameters before and after the slash are for ViTPose and ViTPose++, respectively.

Model	Bz	Lr	Wd	Lrd	Dpr
ViTPose-S/ViTPose++-S	512/1024	5e-4/1e-3	0.1	0.80	0.10
ViTPose-B/ViTPose++-B	512/1024	5e-4/1e-3	0.1	0.75	0.30
ViTPose-L/ViTPose++-L	512/1024	5e-4/1e-3	0.1	0.80	0.50
ViTPose-H/ViTPose++-H	512/1024	5e-4/1e-3	0.1	0.80	0.55
ViTPose-G	512	5e-4	0.1	0.85	0.55

4.4 Experiments

4.4.1 Datasets and evaluation metrics

Datasets. We adopt the MS COCO dataset to evaluate the performance of ViTPose and use several different pose estimation datasets to evaluate the proposed ViTPose++, *e.g.*, MS COCO [100], AIC [182], and MPII [4] for human pose estimation, COCO-W [73] for

all body keypoint detection including those of face, hands, and feet, AP-10K [206] and APT-36K [202] for animal pose estimation. Interhand2.6M [115] is also used to evaluate our models' data efficiency in transfer learning. OCHuman [222] dataset is only used for evaluation to measure the performance of different models in dealing with occluded persons. Specifically, **MS COCO** contains 118K images and 150K human instances with at most 17 keypoint annotations for each instance. The dataset is under the CC-BY-4.0 license. **COCO-W** adopts the images from the MS COCO dataset but provides additional annotations on the face, feet, and hands of each person instance, resulting in at most 133 keypoints for each instance. **MPII** is under the BSD license and contains 25K images and over 40K human instances. There are at most 16 human keypoints for each instance annotated in this dataset. **AIC** is much bigger than other datasets and contains over 200K training images and 350 human instances, with at most 14 keypoints for each annotated instance, including shoulder, elbow, wrist, hip, knee, ankle, and head. **OCHuman** contains human instances with heavy occlusions and is only used for evaluation. It consists of 4K images and 8K instances. **AP-10K** dataset follows the CC-BY-4.0 license and is used for animal pose estimation. It contains 10K images with 54 different animal categories. Similarly, **APT-36K** dataset contains 36K images belonging to 30 different animal categories. 17 different keypoints are annotated in AP-10K and APT-36K datasets for each animal instance. **Interhand2.6M** contains 2.6M labeled hand frames generated from different subjects with both 2D and 3D annotations. We focus on 2D hand pose estimation temporally and leave 3D pose estimation [149] as our future work.

Metrics. We adopt the average precision (AP) as the primary evaluation metric on most of the datasets, including COCO [100], AIC [182], COCO-W [73], OCHuman [222], and AP-10K [206], AP-36K [202]. It is calculated by evaluating the object keypoint similarity (OKS) using different thresholds from 0.5 to 0.95 [100]. A loose metric AP_{50} and a strict metric AP_{75} are also utilized by setting the threshold to 0.5 and 0.75, respectively. PCKh is adopted as the evaluation metric on the MPII [4] dataset following the common practice, which evaluates the accuracy of each keypoint with a matching threshold related to the head segment length.

TABLE 4.2. Ablation study of the backbone and decoder in ViTPose on the MS COCO val set.

Backbone	ResNet-50		ResNet-152		ViTPose-S		ViTPose-B		ViTPose-L		ViTPose-H	
Decoder	Classic	Simple	Classic	Simple	Classic	Simple	Classic	Simple	Classic	Simple	Classic	Simple
AP	71.8	53.1	73.5	55.3	73.8	73.5	75.8	75.5	78.3	78.2	79.1	78.9
AP_{50}	89.8	86.9	90.5	87.9	90.1	90.0	90.7	90.6	91.4	91.4	91.7	91.6
AR	77.3	62.0	79.0	63.8	79.2	78.9	81.1	80.9	83.5	83.4	84.1	84.0
AR_{50}	93.7	92.1	94.3	92.9	94.0	94.0	94.6	94.6	95.3	95.3	95.4	95.4

TABLE 4.3. Ablation study of the decoder in ViTPose on the MS COCO val set. Classic-FP denotes using a randomly initialized and fixed predictor within the Classic decoder (Fig. 4.3).

	Classic	Classic-FP	Simple	Minimal
AP	75.8	75.6	75.5	75.4
AP_{50}	90.7	90.5	90.6	90.6
AR	81.1	80.9	80.9	80.7
AR_{50}	94.6	94.3	94.6	94.5

TABLE 4.4. The performance of ViTPose-B using different pre-training data on the MS COCO val set .

Pre-training Dataset	Dataset Volume	AP	AP_{50}	AP_{75}
ImageNet-1k	1M	75.8	90.7	83.2
COCO (cropping)	150K	74.5	90.5	81.9
COCO+AIC (cropping)	500K	75.8	90.8	83.0
COCO+AIC (no cropping)	300K	75.8	90.5	83.0

4.4.2 Implementation details

ViTPose. In the **top-down paradigm**, ViTPose follows the standard top-down setting for human pose estimation, *i.e.*, a person detector is used to detect person instances and ViTPose is employed to estimate the keypoints for each of the detected instances. The detection results from SimpleBaseline [184] are utilized to evaluate ViTPose's performance on the MS COCO Keypoint val set. We use ViT-S, ViT-B, ViT-L, ViT-H [45], and ViTAE-G [217] as the backbone networks and denote the corresponding models as ViTPose-S, ViTPose-B, ViTPose-L, ViTPose-H, and ViTPose-G, respectively. The backbones are initialized with MAE [57] pre-trained weights, and the models are trained on 8 A100 GPUs based on the MMPose codebase [34]. We make no modifications on the backbone networks' structure since we do not want to introduce extra bias during the training. The default training setting in MMPose is adopted for training the ViTPose models, *i.e.*, we use the 256×192 input resolution and an

AdamW [139] optimizer with a learning rate of 5e-4. We generate the human-centric images using the groundtruth bounding boxes in the annotations, with random shifts and scaling as augmentations. Udp [69] is used for post-processing. The models are trained for 210 epochs, and the learning rate is reduced by multiplying 0.1 at the 170th and 200th epoch, respectively. We sweep each model's layer-wise learning rate decay and stochastic drop path ratio and provide the optimal settings in Table 4.1. In the **Bottom-up paradigm**, we use the whole image as input and resize it to 512×512 during training, following the common practice in HigherHRNet [31]. The models are trained for 300 epochs with an initial learning rate of 1.5e-3. An AdamW [139] optimizer is adopted during training. The learning rate is reduced by a factor of 10 at the 200th and 260th epoch, respectively.

ViTPose++. We also use ViT-S, ViT-B, ViT-L, and ViT-H [45] as the backbone networks and denote the corresponding models as ViTPose++-S, ViTPose++-B, ViTPose++-L, and ViTPose++-H, respectively. The backbones are initialized with MAE [57] pre-trained weights and the models are trained on 8 A100 GPUs based on the MMPose codebase [34]. The models are trained with the AdamW optimizer [139] for 210 epochs. A linear warmup of 500 iterations is also incorporated during training. The learning rate is reduced by multiplying 0.1 at the 170th and 200th epoch, respectively. We randomly sample 1,024 images from all the used training datasets to construct each mini-batch.

4.4.3 Ablation studies of ViTPose and analysis

The structural simplicity and scalability. We train ViTPose with the classic decoder and simple decoder as described in Section 4.3.1, respectively. We also train SimpleBaseline [184] with the ResNet [61] backbones using the two kinds of decoders for reference. Table 4.2 shows the results. It can be observed that using the simple decoder in SimpleBaseline can lead to about 18 AP drops for both ResNet-50 and ResNet-152. However, ViTPose with a plain vision transformer as the backbone and the simple decoder performs well, *i.e.*, only marginal performance drops less than 0.3 AP are observed for either small or large ViTPose models. It indicates that the feature representation extracted by CNN is far from satisfactory, no matter for the small-scale model ResNet-50 or large scale model ResNet-152. Such failure may be

caused by the limited reception field of convolution neural networks, while the transformer-based models can obtain better performance by jointly modeling the short- and long-range dependency. Such a good property can help the model better deal with occlusion and simplify the requirement for the decoder design, as discussed in the following part. For the AP_{50} and AR_{50} metrics, ViTPose obtains similar great performance regarding both decoders, further showing that the plain vision transformer has a strong representation ability to encode the linearly separable features and can thus relieve the necessity of complex decoders. Such a good feature extraction ability can also aid the network in scaling, *i.e.*, it can also be concluded from the table that the performance of ViTPose improves consistently with the increase of the model size.

The influence of simpler decoders. To further investigate the representation capabilities of vision transformer backbones, we incorporate three simplified decoder designs, namely classic-FP, simple, and minimal decoders, into the ViTPose framework. The performance of the ViTPose with these simplified decoder designs is summarized in Table 4.3. Surprisingly, even with the frozen projection layer in the classic-FP or the minimal decoder, ViTPose still achieves an impressive performance of over 75.4 AP. It suggests that a linear layer is enough to act as the task-specific module for pose estimation, demonstrating the excellent representation capability of vision transformers in learning linearly separable features from complex visual data. This emphasizes the notion that *fundamentals speak simply*.

The influence of pre-training data. To evaluate whether or not ImageNet data are necessary for pose estimation tasks, we pre-train the transformer backbone on different datasets, *i.e.*, ImageNet-1k [40], MS COCO [100], and a combination of MS COCO and AIC [182], respectively. Since images in the ImageNet-1k dataset are iconic, we also try to crop the person instances from the MS COCO and AIC training set to form new training data for pre-training. The models are pre-trained for 1,600 epochs on the three datasets, respectively, and then fine-tuned on the MS COCO dataset with pose annotations for 210 epochs. The results are summarized in Table 4.4. It can be seen that with the combination of MS COCO and AIC data for pre-training, ViTPose achieves comparable performance compared with that using ImageNet-1k, while the dataset volume is only half of the ImageNet-1k. It implies

that pre-training on the data from downstream tasks has better data efficiency, validating the flexibility of ViTPose in choosing pre-training data. Nevertheless, the AP decreases by 1.3 if only MS COCO data are used for pre-training. It may be caused by the limited volume of the MS COCO dataset, *i.e.*, the number of instances in MS COCO is on 1/3 of the combination of MS COCO and AIC. Besides, there is no obvious benefit of using cropping on the pre-training images by comparing the results in the last two rows, although the larger dataset volume after cropping means a higher training cost. These results further validate that ViTPose has better data efficiency in the pre-training stage when the data come from the same type of tasks as the fine-tuning stage.

TABLE 4.5. The performance of ViTPose-B using different pre-training methods on the MS COCO val set.

	Random	DeiT [167]	MoCov3 [29]	MAE [57]
AP	72.8	72.7	72.1	75.8

TABLE 4.6. The performance of ViTPose-B regarding different input resolutions on the MS COCO val set.

	224x224	256x192	256x256	384x288	384x384	576x432
AP	74.9	75.8	75.8	76.9	77.1	77.8
AR	80.4	81.1	81.1	81.9	82.0	82.6

TABLE 4.7. The performance of ViTPose-B with 1/8 feature size on the MS COCO val set. "*" means fp16 is used during training due to the limit of hardware memory. For the combination of vanilla self-attention (Full) and window-based attention (Window), we follow ViTDet [92] and use full attention every 1/4 of all the layers. "Shift" and "Poll" denote the two strategies described in Section 4.3.3.

Full	Window	Shift	Pool	Window Size	Training Memory (M)	GFLOPs	AP	AP_{50}	AR	AR_{50}
✓				N/A	36,141*	76.59	77.4	91.0	82.4	94.9
	✓			(8, 8)	21,161	66.31	66.4	87.7	72.9	91.9
	✓	✓		(8, 8)	21,161	66.31	76.4	90.9	81.6	94.5
	✓		✓	(8, 8)	22,893	66.39	76.4	90.6	81.6	94.6
	✓	✓	✓	(8, 8)	22,893	66.39	76.8	90.8	81.9	94.8
✓	✓			(8, 8)	28,594	69.94	76.9	90.8	82.1	94.7
	✓	✓	✓	(16, 12)	26,778	68.46	77.1	91.0	82.2	94.8

The influence of pre-training methods. We also conduct a comprehensive examination of various pre-training methods to investigate their impact on pose estimation, encompassing random initialization, supervised pre-training, contrastive self-supervised pre-training, and

masked image pre-training. Specifically, we employed DeiT [167] and MoCov3 [29] as the supervised and contrastive self-supervised pre-training methods, respectively, both applied to the ImageNet-1K dataset. The results in Table 4.5 show that random initialization, supervised pre-training, and contrastive self-supervised pre-training achieve comparable performance on the MS COCO dataset, with random initialization yielding slightly superior performance. This suggests that vision transformers pre-trained using supervised or contrastive self-supervised methods may learn classification-related features that generalize poorly to the pose estimation task. In contrast, using masked image pre-training, *e.g.*, MAE [57], significantly enhances the performance.

The influence of input resolution. To evaluate whether or not ViTPose can adapt well to different input resolutions, we train ViTPose with different input sizes and summarize the results in Table 4.6. The performance of ViTPose-B improves with the increase in the input size. It is also noted that the squared input only brings marginal or even no gains over the rectangular one, *e.g.*, 256×256 v.s. 256×192. The reason may be that the average aspect ratio of human instances in MS COCO is about 4:3, and the squared input size does not fit the statistics well.

The influence of attention type. As demonstrated in HRNet [154] and HRFormer [211], high-resolution feature maps are beneficial for pose estimation. ViTPose can easily generate high-resolution features by varying the down-sampling ratio of the patching embedding layer, *i.e.*, from 1/16 to 1/8. Besides, to alleviate the out-of-memory issue caused by the quadratic computational complexity of the vanilla self-attention, window-based attention with the shifted window and pooling window strategies described in Section 4.3.3 is used. The results are presented in Table 4.7. Directly using full attention with 1/8 feature size obtains the best 77.4 AP on the MS COCO val set while suffering from a large memory footprint even under the mixed-precision training mode. Window-based attention can alleviate the memory issue while at the cost of performance drop due to lacking global context modeling, *e.g.*, from 77.4 AP to 66.4 AP. The shifted window and pooling window strategies both promote cross-window information exchange for global context modeling and thus significantly improve the performance by 10 AP with less than 10% memory increase. When applying the two

mechanisms together, *i.e.*, the 5th row, the performance further increases to 76.8 AP, which is comparable to the strategy proposed in ViTDet [92] that jointly uses full and window attention (the 6th row), while having less memory footprint, *i.e.*, 76.8 AP v.s. 76.9 AP and 22.9G memory v.s. 28.6G memory. Comparing the 5th and last row in Table 4.7, we also note that the performance can be further improved from 76.8 AP to 77.1 AP by enlarging the window size from 8×8 to 16×12, which outperforms the ViTDet setting and is comparable with the full attention setting in the first row while having fewer computations and less memory footprint.

TABLE 4.8. The performance of ViTPose-B at three partially fine-tuning settings on the MS COCO val set.

FFN	MHSA	Memory (M)	GFLOPs	AP	AP_{50}	AR	AR_{50}
✓	✓	14,090	17.1	75.8	90.7	81.1	94.6
✓		11,052	10.9	75.1	90.5	80.3	94.4
	✓	10,941	6.2	72.8	89.8	78.3	93.8

The influence of partially fine-tuning. To assess whether or not vision transformers can still perform well for pose estimation via partially fine-tuning, we fine-tune the ViTPose-B model at three settings, *i.e.*, fully fine-tuning, freezing the MHSA and freezing the FFN. As shown in Table 4.8, with the MHSA frozen, the performance drops moderately compared with the fully fine-tuning setting, *i.e.*, 75.1 AP v.s. 75.8 AP. The AP_{50} metric is almost the same for the two settings. However, there is a significant drop of 3.0 AP when freezing the FFN. This finding implies that the FFN of transformers is more responsible for task-specific modeling. In contrast, the MHSA is insensitive to different tasks, *e.g.*, probably being only responsible for modeling the relationship of tokens based on feature similarity no matter in the MIM pre-training task or the down-stream pose estimation task.

TABLE 4.9. The performance of knowledge distillation from ViTPose-L to ViTPose-B on the MS COCO val set.

Heatmap	Token	Memory (M)	AP	AP_{50}	AR	AR_{50}
-	-	14,090	75.8	90.7	81.1	94.6
	✓	14,203	76.0	90.7	81.3	94.8
✓		15,458	76.3	90.8	81.5	94.8
✓	✓	15,565	76.6	90.9	81.8	94.9

The analysis of transferability. To evaluate the transferability of ViTPose, we use both the classic output distillation method and the proposed knowledge token distillation method to

transfer the knowledge from ViTPose-L to ViTPose-B. The results are listed in Table 4.9. As can be seen, the token-based distillation brings a gain of 0.2 AP with a marginal cost of extra memory footprint. In comparison, the output distillation brings a gain of 0.5 AP with a moderate cost of extra memory footprint. It is noteworthy that the proposed token-based distillation does not require the teacher model to be served during the training of the student model, thereby bringing fewer computations and less memory footprint compared with the output distillation method. The two distillation methods are complementary, and using them together obtains 76.6 AP, validating the excellent transferability of ViTPose models.

4.4.4 Ablation studies of ViTPose++ and analysis

4.4.4.1 Different settings of ViTPose++

The baseline method. Since the decoder in ViTPose is rather simple and lightweight, we can easily extend ViTPose to deal with multiple types of body pose estimation tasks by using a shared backbone and individual decoder for each task. Specifically, images from each pose estimation task are randomly selected and organized into a batch. These images are then fed into a shared encoder network for feature extraction. The extracted features are subsequently passed to the respective task-specific decoder, which is responsible for predicting the keypoints specific to the given task. This straightforward extension serves as the baseline method and is illustrated in Fig. 4.2 (d). We gradually introduce various datasets such as MS COCO [100], AIC [182], MPII [4], COCO-W [73], AP-10K [206], and APT-36K [202] to formulate the training data. The results on the MS COCO val set are listed in Table 4.10. Note that we directly use the models after multi-task training for evaluation without further fine-tuning. It can be observed that the performance of ViTPose increases consistently from 75.8 AP to 77.1 AP by using human pose estimation datasets (MS COCO, AIC, and MPII) for training. Although the dataset volume of MPII is much smaller than the combination of MS COCO and AIC (40K v.s. 500K), using MPII for training still brings a 0.1 AP increase, showing that ViTPose can well harness the diverse data in different datasets. With the same data but novel annotations in COCO-W for training, there is no performance gain or drop, showing that ViTPose can well encode the human body feature representations and mitigate

the side effect of allocating the representation capacity for those distinct keypoints in the face, hands, and feet. However, with the animal datasets (AP-10K and APT-36K) involved, the performance of the baseline method drops from 77.0 to 76.7, probably due to the conflict between different tasks (*i.e.*, humans and animals).

TABLE 4.10. The performance of ViTPose-B at the multi-task training setting on the MS COCO val set.

COCO	AIC	MPII	COOC-W	AP10K	APT36K	AP	AP_{50}	AR	AR_{50}
✓	-	-	-	-	-	75.8	90.7	81.1	94.6
✓	✓	-	-	-	-	77.0	90.8	82.2	94.9
✓	✓	✓	-	-	-	77.1	90.8	82.2	94.7
✓	✓	✓	✓	-	-	77.1	90.8	82.2	94.8
✓	✓	-	-	✓	-	76.8	90.8	82.0	94.7
✓	✓	-	-	✓	✓	76.7	90.7	81.8	94.6

ViTPose++. We consider three variants of ViTPose++ by changing the configurations of the experts in FFN. **1) Independent FFN (I-FFN)**: To mitigate the conflict between different tasks, we use an independent for each task in each transformer block. In this way, each FFN can only process the images of the dataset corresponding to a specific pose estimation task. All these independent FFNs are initialized with the weights of the original FFN from the MAE pre-trained models. **2) Independent and Shared FFN (IS-FFN)**: Although there may exist conflicts between different tasks, they may also share some common knowledge of body poses, *e.g.*, the locations of eyes are symmetric for both human beings and most kinds of animals. To encode such common knowledge, we introduce a shared FFN in each transformer block in addition to the independent ones. In this way, each image will be processed by not only a task-specific FFN but also the shared FFN. The output features from a task-specific FFN and the shared FFN are then summed together and used as the input to the next transformer layer. We initialize the shared FFN with the weights from the MAE pre-trained models and the independent FFNs with zero. **3) Partially Shared FFN (PS-FFN)**: We also explore another design choice to jointly encode the common knowledge and task-specific knowledge of body poses, *i.e.*, the default setting of ViTPose++. As described in Section 4.3.5, we split the last linear layer of each FFN into a shared part and an independent part along the channel dimension and leave the first linear layer shared for all tasks. We duplicate the independent part for all the tasks while not sharing their weights during training. We initialize the shared

and independent parts with the weights from the MAE pre-trained models. The features from the first linear layer are processed by the shared part and the corresponding independent part of the second layer, respectively. The output features are then concatenated along the channel dimension and used as the input to the next transformer layer.

TABLE 4.11. The performance of ViTPose++-B at different settings on the MS COCO val set. ViTPose-B is only trained on the MS COCO dataset. "MT Baseline" denotes the multi-task training baseline method (Section 4.4.4.1) based on ViTPose-B on the combination of MS COCO, AIC, and AP-10K datasets. The ViTPose++-B at the rest settings are described in Section 4.4.4.1 and trained on the same datasets as "MT Baseline". The number of parameters during inference is also listed in the second column.

Model	Params (M)	AP	AP_{50}	AR	AR_{50}
ViTPose-B	86	75.8	90.7	81.1	94.6
MT Baseline	86	76.8	90.8	82.0	94.7
I-FFN	86	75.8	90.5	81.1	94.4
IS-FFN	143	77.0	90.8	82.2	94.8
PS-FFN ($\alpha = 1/6$)	86	77.0	90.8	82.0	94.7
PS-FFN ($\alpha = 1/4$)	**86**	**77.0**	**90.9**	**82.2**	**94.8**
PS-FFN ($\alpha = 1/3$)	86	77.0	90.8	82.1	94.7

We use the combination of MS COCO, AIC, and AP-10K datasets to train the multi-task baseline model and ViTPose++ at the above three settings. The results are summarized in Table 4.11. It can be observed that without modeling the common knowledge between different tasks, the performance of ViTPose++ with I-FFN obtains 75.8 AP, which is only comparable to the results of ViTPose trained on MS COCO and much worse than the results of the multi-task baseline method (76.8 AP). Although the MHSA layers are shared by all datasets in ViTPose++ with I-FFN, they are almost task-agnostic, as also evidenced in Table 4.8, thus explaining the unsatisfactory result of I-FFN. With PS-FFN, ViTPose++ obtains 77.0 AP, which is the same as the performance of using human datasets only for training (2nd row in Table 4.10). The results demonstrate that a proper design for encoding the common and task-specific knowledge of the body poses matters for addressing the conflict between different tasks. Nevertheless, it introduces extra computations and parameters compared with single-task ViTPose and the multi-task baseline since there are additional FFNs used for each task. With a proper partition ratio α in the PS-FFN setting, *i.e.*, 0.25, ViTPose++ with PS-FFN achieves a better trade-off between performance and model complexity, *i.e.*, obtaining 77.0

AP without extra computations and parameters. Considering that ViTPose++ can be easily extended to handle more types of pose estimation tasks, the results show its flexibility and potential for building a foundation model towards generic body pose estimation.

4.4.5 Comparison with SOTA methods

4.4.5.1 The performance on MS COCO

TABLE 4.12. Comparison of ViTPose and ViTPose++ with SOTA methods on the MS COCO val set.

Model	Backbone	Params (M)	Speed (fps)	FLOPs (G)	Mem. (M)	Input Res.	Feature Res.	COCO val AP	AP_M	AP_L	AR
SimpleBaseline [184]	ResNet-152	60	829	15.7	3827	256x192	1/32	73.5	69.9	80.2	79.0
HRNet [154]	HRNet-W32	29	428	16.0	7049	384x288	1/4	75.8	71.9	82.8	81.0
HRNet [154]	HRNet-W48	64	309	32.9	7339	384x288	1/4	76.3	72.3	83.4	81.2
UDP [69]	HRNet-W48	64	309	32.9	7339	384x288	1/4	77.2	73.2	84.4	82.0
FastPose [47]	ResNet-152	60	653	16.0	6013	256x192	1/32	73.3	-	-	-
TokenPose-L/D24 [94]	HRNet-W48	28	602	11.0	3477	256x192	1/4	75.8	72.3	82.7	80.9
TransPose-H/A6 [201]	HRNet-W48	18	309	21.8	-	256x192	1/4	75.8	76.4	87.2	80.8
HRFormer-B [211]	HRFormer-B	43	158	12.2	4287	256x192	1/4	75.6	71.7	82.6	80.8
HRFormer-B [211]	HRFormer-B	43	78	26.8	7859	384x288	1/4	77.2	73.2	84.2	82.0
ViTPose-S	ViT-S	22	1439	5.3	3438	256x192	1/16	73.8	70.5	80.4	79.2
ViTPose-B	ViT-B	86	944	17.1	4589	256x192	1/16	75.8	72.1	82.2	81.1
ViTPose-L	ViT-L	307	411	59.8	5587	256x192	1/16	78.3	74.5	85.4	83.5
ViTPose-H	ViT-H	632	241	122.9	7293	256x192	1/16	79.1	75.3	86.0	84.1
ViTPose++-S	ViT-S	22	1439	5.3	3438	256x192	1/16	75.8	72.3	82.6	81.0
ViTPose++-B	ViT-B	86	944	17.1	4589	256x192	1/16	77.0	73.4	84.0	82.6
ViTPose++-L	ViT-L	307	411	59.8	5587	256x192	1/16	78.6	75.2	85.6	84.1
ViTPose++-H	ViT-H	632	241	122.9	7293	256x192	1/16	79.4	75.8	86.5	84.8

Since MS COCO [100] is the most popular and representative dataset among all the body pose estimation tasks, we first compare the performance of ViTPose and ViTPose++ with SOTA methods on the MS COCO dataset. Specifically, we report their results at both the top-down and bottom-up paradigms. Then, we further compare their performance on other body pose estimation datasets.

Top-down paradigm. Based on the previous analysis, we use the input resolution of 256×192 during training and report the results on the MS COCO val set as shown in Table 4.12. The speed of all methods is recorded on a single A100 GPU with a batch size of 64. The ViTPose++ models are trained on the combination of MS COCO, AIC, MPII, COCO-W, AP-10K,

and APT-36K datasets. It can be observed that ViTPose achieves a better trade-off between throughput and accuracy, showing that the plain vision transformer has a strong representation ability and is computationally friendly to modern hardware devices. For example, the ViTPose-S model with fewer parameters and fewer memory footprint obtains similar performance compared to SimpleBaseline [184] and FastPose [47] using a ResNet-152 [61] backbone. Besides, ViTPose performs better with much larger backbones, demonstrating the good scalability of ViTPose. For example, ViTPose-L obtains much better performance than ViTPose-S and ViTPose-B, *i.e.*, 78.3 AP v.s. 73.8 AP and 75.8 AP. ViTPose-L also outperforms previous SOTA methods based on CNN and transformers (*e.g.*, UDP [69] and TokenPose [94]) by a large margin while keeping a similar inference speed. Furthermore, in comparison to the strong transformer-based baseline method HRFormer-B [211], the proposed plain vision transformer-based ViTPose delivers comparable performance. For instance, ViTPose-H (16th row) achieves slightly better performance and faster inference speed compared to HRFormer-B (10th row), with 79.1 AP v.s. 75.6 AP and 241 fps v.s. 158 fps, respectively. Even when compared to HRFormer-B with a larger input resolution (11th row), ViTPose-H still outperforms it while consuming less memory. These results highlight that despite having a higher number of parameters, the plain vision transformer structure, with its simple matrix multiplication operations, exhibits excellent compatibility with modern hardware designs. It also has the potential to achieve better performance with reduced memory consumption and faster speed, pointing towards a new design paradigm for future works. While using multiple body pose estimation datasets for training, the performance of ViTPose++ further increases. For example, the small model ViTPose++-S obtains 75.8 AP with 1,439 fps, which is comparable to the larger model ViTPose-B but has a much faster speed. The results show the good scalability and flexibility of ViTPose regarding both model structures and training data.

We then build a much stronger ViTPose-G model using the ViTAE-G [217] backbone, which has about 1B parameters and larger input resolution (576×432) and trained on the combination of MS COCO, AIC, and MPII datasets. A more powerful detector from Bigdet [12] is also used to provide person detection results (68.5 AP on the person class of COCO dataset). As shown in Table 4.13, ViTPose-G with the ViTAE-G backbone outperforms all previous SOTA

TABLE 4.13. Comparison with SOTA methods on the MS COCO test-dev set. "◇" means model ensemble. "†", "‡", and "*" denote the champions of the 2018, 2019, and 2020 MS COCO Human Keypoint Detection Challenge, respectively.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_M	AP_L	AR
Baseline$^{\diamond}$ [184]	ResNet-152	76.5	92.4	84.0	73.0	82.7	81.5
RMPE [48]	PyraNet	68.8	87.5	75.9	64.6	75.1	73.6
CPN+ [30]	ResNet-Inception	73.0	91.7	80.9	69.5	78.1	79.0
HRNet [154]	HRNet-w48	77.0	92.7	84.5	73.4	83.1	82.0
HRFormer [211]	HRFormer-B	76.2	92.7	83.8	72.5	82.3	81.2
MSPN$^{\diamond\dagger}$ [91]	4xResNet-50	78.1	94.1	85.9	74.5	83.3	83.1
SwinV2 [189]	SwinV2-L	77.2	-	-	-	-	-
DARK [214]	HRNet-w48	77.4	92.6	84.6	73.6	83.7	82.3
RSN$^{\diamond\ddagger}$ [13]	4xRSN-50	79.2	94.4	87.1	76.1	83.8	84.1
CCM$^{\diamond}$ [215]	HRNet-w48	78.9	93.8	86.0	75.0	84.5	83.6
UDP++$^{\diamond *}$ [69]	HRNet-w48plus	80.8	94.9	88.1	77.4	85.7	85.3
ViTPose-B	ViT-B	75.1	92.5	83.1	72.0	80.7	80.3
ViTPose-L	ViT-L	77.3	93.1	85.3	74.0	83.1	82.4
ViTPose-H	ViT-H	78.1	93.3	85.7	74.9	83.8	83.1
ViTPose-G	ViTAE-G	80.9	94.8	88.1	77.5	85.9	85.4
ViTPose-G$^{\diamond}$	ViTAE-G	81.1	95.0	88.2	77.8	86.0	85.6

methods on the MS COCO test-dev set at 80.9 AP, where the previous best method UDP++ uses 17 models and a slightly better detector (68.6 AP on the person class of COCO dataset) and only obtains 80.8 AP. Our ensemble model ViTPose-G$^{\diamond}$ with only three models further achieves the best 81.1 AP.

TABLE 4.14. Comparison of ViTPose and SOTA methods in the bottom-up paradigm on the MS COCO val set.

Model	Backbone	COCO Val			
		AP	AP_{50}	AR	AR_{50}
Associate Embedding [117]	Hourglass	61.3	83.3	65.9	85.0
Associate Embedding [117]	ResNet-50	46.6	74.2	56.6	81.0
Associate Embedding [117]	ResNet-101	55.4	80.7	62.2	84.1
Associate Embedding [117]	ResNet-152	59.5	82.9	65.1	85.6
Pifpaf [78]	ResNet-50	62.6	-	-	-
HigherHRNet-w32 [31]	HRNet-w32	67.7	87.0	72.3	89.0
HigherHRNet-w48 [31]	HRNet-w48	68.6	87.3	73.1	89.2
ViTPose-B	ViT-B	68.5	87.3	72.9	89.2
ViTPose-L	ViT-L	70.1	88.3	74.5	90.2

Bottom-up paradigm. We use images of size 512×512 in MS COCO to train the ViTPose models in the bottom-up paradigm and report the performance on the MS COCO val set. 1/8 feature resolutions with full window-based attention are adopted. The results are listed

in Table 4.14. Without delicate designs, the plain vision transformer obtains 68.5 AP with ViTPose-B and 70.1 AP with ViTPose-L, demonstrating the good scalability of ViTPose in the bottom-up paradigm. Compared with previous SOTA methods, *e.g.*, HigherHRNet with the HRNet-w48 backbone, ViTPose-L model achieves better performance and sets a new SOTA, *i.e.*, 68.6 AP v.s. 70.1 AP. These results validate the flexibility of plain vision transformers for different keypoint detection paradigms.

TABLE 4.15. Comparison of ViTPose++ and SOTA methods on the OCHuman [222] val and test set with ground truth bounding boxes.

Model	Backbone	Resolution	Val Set				Test Set			
			AP	AP_{50}	AR	AR_{50}	AP	AP_{50}	AR	AR_{50}
SimpleBaseline [184]	ResNet-152	384x288	58.8	72.7	63.1	75.7	58.2	72.3	62.7	75.2
HRNet [154]	HRNet-w32	384x288	60.9	76.0	65.1	78.2	60.6	74.8	64.7	77.6
HRNet [154]	HRNet-w48	384x288	62.1	76.1	65.9	78.2	61.6	74.9	65.3	77.3
MIPNet [75]	HRNet-w48	384x288	74.1	89.7	81.0	-	-	-	-	-
HRFormer [211]	HRFormer-S	384x288	53.1	73.1	59.6	76.9	52.8	72.8	59.1	76.6
HRFormer [211]	HRFormer-B	384x288	50.4	71.5	58.8	76.6	49.7	71.6	58.2	76.0
ViTPose++-S	ViT-S	256x192	79.4	90.7	81.6	91.4	78.4	90.6	80.6	91.0
ViTPose++-B	ViT-B	256x192	83.7	91.8	85.4	92.9	82.6	91.7	84.6	92.4
ViTPose++-L	ViT-L	256x192	87.4	93.7	88.8	94.2	85.7	92.8	87.5	93.4
ViTPose++-H	ViT-H	256x192	86.8	92.8	88.3	93.7	85.7	92.8	87.4	93.8

TABLE 4.16. Comparison (PCKh) of ViTPose++ and SOTA methods on the MPII [4] val set with ground truth bounding boxes.

Model	Backbone	Resolution	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
SimpleBaseline [184]	ResNet-152	256x256	86.9	95.4	89.4	84.0	88.0	84.6	82.1	89.0
HRNet [154]	HRNet-w32	256x256	96.9	85.9	90.5	85.9	89.1	86.1	82.5	90.0
HRNet [154]	HRNet-w48	256x256	97.1	95.8	90.7	85.6	89.0	86.8	82.1	90.1
CFA [153]	ResNet-101	384x384	95.9	95.4	91.0	86.9	89.8	87.6	83.9	90.1
ASDA [9]	HRNet-w48	256x256	97.3	96.5	91.7	87.9	90.8	88.2	84.2	91.4
TransPose-H-A6 [201]	HRNet-w48	256x256	-	-	-	-	-	-	-	92.3
OKDHP [96]	Hourglass	256x256	97.3	96.1	91.2	86.8	89.9	86.9	83.1	90.6
HRFormer [211]	HRFormer-S	256x256	97.1	95.8	90.5	85.9	88.7	85.7	82.1	89.9
HRFormer [211]	HRFormer-B	256x256	96.8	96.1	90.4	85.9	89.0	87.3	84.1	90.4
ViTPose++-S	ViT-S	256x192	97.4	97.2	92.9	89.0	92.3	90.4	86.8	92.7
ViTPose++-B	ViT-B	256x192	97.3	97.2	93.3	89.7	91.5	90.7	87.2	92.8
ViTPose++-L	ViT-L	256x192	98.0	97.6	94.3	90.9	92.9	92.6	89.5	94.0
ViTPose++-H	ViT-H	256x192	97.8	97.6	94.4	91.5	93.2	92.8	90.2	94.2

4.4.5.2 The performance on other datasets

To evaluate the performance of ViTPose comprehensively, apart from the results on the MS COCO val and test-dev set, we also report the performance of ViTPose++-S, ViTPose++-B,

TABLE 4.17. Comparison of ViTPose++ and SOTA methods on the AIC [182] val set with ground truth bounding boxes.

Method	Backbone	Resolution	AP	AP_{50}	AP_{75}	AR	AR_{50}
SimpleBaseline [184]	ResNet-50	256x192	28.0	71.6	15.8	32.1	74.1
SimpleBaseline [184]	ResNet-101	256x192	29.4	73.6	17.4	33.7	76.3
SimpleBaseline [184]	ResNet-152	256x192	29.9	73.8	18.3	34.3	76.9
HRNet [154]	HRNet-w32	256x192	32.3	76.2	21.9	36.6	78.9
HRNet [154]	HRNet-w48	256x192	33.5	78.0	23.6	37.9	80.0
HRFormer [211]	HRFomer-S	256x192	31.6	75.9	20.9	35.8	78.0
HRFormer [211]	HRFomer-B	256x192	34.4	78.3	24.8	38.7	80.9
ViTPose++-S	ViT-S	256x192	29.7	74.6	17.6	34.3	77.1
ViTPose++-B	ViT-B	256x192	31.8	76.7	20.6	36.3	79.0
ViTPose++-L	ViT-L	256x192	34.3	79.1	24.1	38.9	81.8
ViTPose++-H	ViT-H	256x192	34.8	80.2	24.5	39.1	82.2

ViTPose++-L, and ViTPose++-H on the OCHuman [222] val and test set, MPII [4] val set, AIC [182] val set, COCO-W [73] val set, AP-10K [206] test set, and APT-36K [202] test set, respectively. Please note that the ViTPose++ models are trained with the combination of all the datasets and directly tested on the target dataset without further fine-tuning, which keeps the whole pipeline as simple as possible. For each dataset, we use the corresponding FFN, decoder, and prediction head in ViTPose for prediction.

OCHuman val and test set. To evaluate the performance of different methods on the human instances with heavy occlusions, we compare ViTPose++ and SOTA methods on the OCHuman val and test set. We adopt the ground truth bounding boxes instead of those obtained by a person detector to isolate the effect of person detection because not all human instances are annotated in the OCHuman datasets, and the person detector may cause false positive or missing bounding boxes, which may obscure the true performance of pose estimation models. Specifically, we use the decoder and prediction head of ViTPose++ corresponding to the MS COCO dataset since the keypoint definition is the same in both the MS COCO and OCHuman datasets. The results are listed in Table 4.15. Compared with previous SOTA methods with complex structures, *e.g.*, MIPNet [75], ViTPose++ obtains a gain of over 10 AP on the OCHuman val set, although there is no special structural design to deal with occlusions, implying the strong feature representation ability of plain vision transformer. It should also be noted that HRFormer [211] experiences large performance drops from MS COCO to OCHuman, and its small model even beats the base model, *i.e.*,

53.1 AP v.s 50.4 AP on the OCHuman val set. Such phenomena imply that HRFormer may overfit the MS COCO dataset, especially for larger models. By contrast, ViTPose++ generally delivers performance gains with the increase of model size and significantly pushes forward the frontier of the keypoint detection performance on OCHuman, *i.e.*, 87.4 AP on the val set and 85.7 AP on the test set.

MPII val set. We evaluate the performance of ViTPose++ and representative models on the MPII val set with the ground truth bounding boxes. Following the default settings of MPII, we use PCKh as the metric for performance evaluation. It can be observed that the previous representative methods have obtained well performance on the MPII dataset, *i.e.*, the strong transformer-based method HRFormer-B obtains over 90 PCKh. Nevertheless, ViTPose demonstrates the potential to further improve performance due to its remarkable scalability. As shown in Table 4.16, ViTPose++ outperforms previous methods in terms of both single joint evaluation and average evaluation, *e.g.*, ViTPose++-S, ViTPose++-B, ViTPose++-L, and ViTPose++-H achieve 92.7, 92.8, 94.0, and 94.2 average PCKh with a smaller input resolution.

AIC val set. We also evaluate the performance of ViTPose++ on the AIC val set. As listed in Table 4.17, compared to representative CNN-based and transformer-based models, our ViTPose++ obtains better performance, *i.e.*, 34.8 AP by ViTPose-H v.s. 33.5 AP by HRNet-w48 and 34.4 AP by HRFromer-B. Nevertheless, the score is still not high enough on the AIC set, indicating that more efforts may be needed to improve the performance further.

TABLE 4.18. Comparison of ViTPose and SOTA methods on the COCO-W [73] val set with detected boxes.

Method	Backbone	Resolution	Body	Foot	Face	Hand	WholeBody
SimpleBaseline [184]	ResNet-50	256x192	65.2	61.4	60.8	46.0	52.0
SimpleBaseline [184]	ResNet-101	256x192	67.0	64.0	61.1	46.3	53.3
HRNet [154]	HRNet-w32	256x192	70.0	56.7	63.7	47.3	55.3
HRNet [154]	HRNet-w48	256x192	70.0	67.2	65.6	53.4	57.9
ViTPose++-S	ViT-S	256x192	71.6	72.1	55.9	45.3	54.4
ViTPose++-B	ViT-B	256x192	73.3	74.2	60.1	50.2	57.4
ViTPose++-L	ViT-L	256x192	75.3	77.1	63.0	54.2	60.6
ViTPose++-H	ViT-H	256x192	75.9	77.9	63.3	54.7	61.2

TABLE 4.19. Comparison of ViTPose and SOTA methods on the AP-10K [206] test set with ground truth bounding boxes.

Method	Backbone	Resolution	AP	AP_{50}	AP_{75}	AP_M	AP_L
SimpleBaseline [184]	ResNet-50	256x256	68.1	92.3	74.0	51.0	68.8
SimpleBaseline [184]	ResNet-101	256x256	68.1	92.2	74.2	53.4	68.8
HRNet [154]	HRNet-w32	256x256	72.2	93.9	78.7	55.5	73.0
HRNet [154]	HRNet-w48	256x256	73.1	93.7	80.4	57.4	73.8
ViTPose++-S	ViT-S	256x192	71.4	93.3	78.4	47.6	71.8
ViTPose++-B	ViT-B	256x192	74.5	94.9	82.2	46.8	75.0
ViTPose++-L	ViT-L	256x192	80.4	97.6	88.5	52.7	80.8
ViTPose++-H	ViT-H	256x192	82.4	98.2	89.5	59.1	82.8

TABLE 4.20. Comparison of ViTPose and SOTA methods on the APT-36K [202] val set with ground truth bounding boxes.

Method	Backbone	Resolution	AP	AP_{50}	AP_{75}	AR	AR_{50}
SimpleBaseline [184]	ResNet-50	256x256	69.4	-	-	-	-
SimpleBaseline [184]	ResNet-101	256x256	69.6	-	-	-	-
HRNet [154]	HRNet-w32	256x256	74.2	-	-	-	-
HRNet [154]	HRNet-w48	256x256	74.1	-	-	-	-
HRFormer [154]	HRFormer-S	256x256	71.3	-	-	-	-
HRFormer [154]	HRFormer-B	256x256	74.2	-	-	-	-
ViTPose++-S	ViT-S	256x192	74.2	94.9	82.3	77.6	95.6
ViTPose++-B	ViT-B	256x192	75.9	95.4	83.7	79.2	96.0
ViTPose++-L	ViT-L	256x192	80.8	97.4	88.5	83.9	97.9
ViTPose++-H	ViT-H	256x192	82.3	97.7	90.6	85.5	98.3

COCO-W val set. Different from the previous datasets, COCO-W contains not only the annotations for typical human body keypoints, but also provides fine-grained annotations on the faces, hands, and feet. Similar to the MS COCO val set, we use the person detection results from SimpleBaseline [184] for evaluation. As shown in Table 4.18, our ViTPose++-B model obtains competitive performance at 57.4 AP on the COCO-W val set. With the model becoming larger, the performance of ViTPose++ further increases, and ViTPose++-H sets a new SOTA, *e.g.*, 61.2 AP, demonstrating the excellent scalability of ViTPose++.

AP-10K val set. To evaluate the performance of ViTPose++ for animal pose estimation, we adopt the AP-10K and APT-36K datasets. We use an input size of 256×192 for ViTPose++ during training, while other methods take a larger size of 256×256 by default. The results are listed in Table 4.19. As can be seen, ViTPose++ performs well not only for human pose estimation but also for animal pose estimation. For example, ViTPose++-B outperforms previous methods by obtaining 74.5 AP. In comparison, the performance of the largest model

ViTPose++-H further increases to 82.4 AP, setting a new record on the animal pose estimation task and demonstrating the potential of ViTPose in dealing with various types of body pose estimation tasks.

APT-36K val set. We also evaluate the performance of ViTPose++ on the APT-36K dataset, which contains a larger number of instances. As shown in Table 4.20, ViTPose++ also achieves better performance on the APT-36K datasets than previous methods. Note that ViTPose++-S delivers a much higher AP on APT-36K than AP-10K, *i.e.*, 74.2 AP v.s. 71.4 AP, probably due to the more balanced data distribution of APT-36K than that of AP-10K.

FIGURE 4.5. Visual results of ViTPose++ on some test images from the MS COCO dataset.

FIGURE 4.6. Visual results of ViTPose++ on some test images from the AIC dataset.

4.4.6 Subjective results

We also provide some visual pose estimation results for subjective evaluation. We show the results of ViTPose++-H on the MS COCO (Figure 4.5), AIC (Figure 4.6), OCHuman (Figure 4.7), MPII (Figure 4.8), COCO-W(Figure 4.9), AP-10K (Figure 4.10), and APT-36K (Figure 4.20) datasets, respectively. As can be seen, ViTPose++ is good at dealing with challenging cases such as occlusions, blur, scale changes, appearance variance, odd body postures, and complex backgrounds, owing to its strong representation ability and flexibility in encoding pose-relevant knowledge from multiple types of body pose estimation datasets.

FIGURE 4.7. Visual results of ViTPose++ on some test images from the OCHuman dataset.

4.4.7 Data efficiency analysis

A critical property of the foundation model is high data efficiency for transfer learning, *i.e.*, performing well on a target domain after fine-tuning on only a small fraction of labeled data. To evaluate the data efficiency of ViTPose under different settings, we first train them with only the MS COCO dataset and the combination of MS COCO, MPII, AIC, and COCO-W datasets, respectively. Then, we demonstrate their generalization ability by further fine-tuning them on two different datasets, *i.e.*, InterHand2.6M [115] and AP-10K [206], to make a comprehensive evaluation of the generalization ability on datasets with different domain shifts, *i.e.*, human body $\rightarrow$ human hand and human body $\rightarrow$ animal body. Specifically, we use the 10%, 20%, 40%, 60%, 80%, and 100% percentage of the training data for fine-tuning.

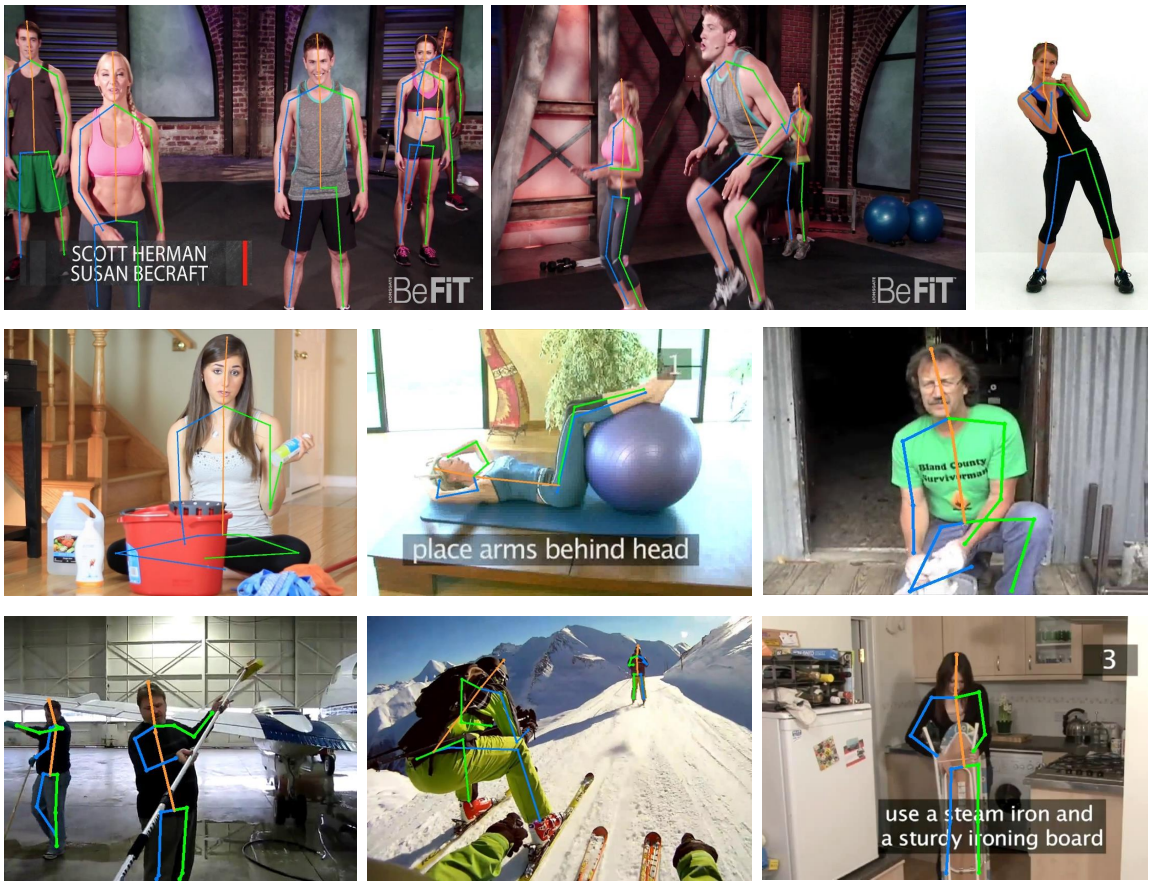


FIGURE 4.8. Visual results of ViTPose++ on some test images from the MPII dataset.

During fine-tuning, we ensure the models are fine-tuned with the same amount of data by keeping the same number of iterations.

InterHand2.6M. We first conduct the data efficiency experiments using ViTPose-B, ViTPose-L, and ViTPose-H on the InterHand2.6M datasets. The results are plotted in Fig. 4.12, where we also provide the results of representative small models trained with 100% training data for reference. It can be observed that the data efficiency is improved with the increase of the model size, *i.e.*, with the same percentage of data for training. ViTPose with the ViT-H backbone always has better performance than that with smaller ones. Moreover, the transformer-based methods obtain better performance gains compared with the CNN-based method, *i.e.*, increasing the transfer data from 10% to 100%, SimpleBaseline with a ResNet-50 backbone only experiences 0.8 AUC increase while ViTPose-B obtains 1.3 AUC improvement.

FIGURE 4.9. Visual results of ViTPose++ on some test images from the COCO-W dataset.

Besides, multi-task training on multiple types of body pose estimation datasets also improve data efficiency. For example, compared with single-task training, the multi-task training helps ViTPose-B/L/H achieve a gain of 0.4/0.3/0.2 AUC when using 10% data for training. In addition, compared with the small models using 100% data for training, ViTPose-H obtains better performance with fewer training data, *e.g.*, ViTPose-H using 20% training data (87.2 AUC) outperforms ViTPose-S (86.5 AUC) and SimpleBaseline with ResNet-50 (85.1 AUC) using 100% training data, showing the high data efficiency of large models. Besides, with the model size increasing, the benefit of using multiple types of pre-training data becomes lower, *i.e.*, the gap between the red and blue curves becomes narrower. It implies that the model size plays a more important role in improving the data efficiency of transfer learning, *i.e.*, suggesting the notion that *more is different*.

FIGURE 4.10. Visual results of ViTPose++ on some test images from the AP-10K dataset.

AP-10K. Similarly, we evaluate the data efficiency of the previous best model ViTPose-H on the AP-10K dataset. The results are shown in Fig. 4.13. It can be observed that with only 10% data for training, ViTPose-H outperforms the representative methods with small models, *e.g.*, ResNet-50, HRNet-32, and HRNet-48, as well as the ViTPose-B and ViTPose-S models, further validating the good data efficiency of the large model.

4.4.8 Visualization and analysis

Visualization of feature maps. To inspect the learning ability of ViTPose, we visualize the feature maps of ViTPose++-B on the normal and heavily occluded images, *i.e.*, we mask half of the images. As shown in Fig. 4.14 (a), ViTPose++-B gradually focuses on the human body from the shallow layers to deep layers in the backbone network. The up-sampling layer

FIGURE 4.11. Visual results of ViTPose++ on some test images from the APT-36K dataset.

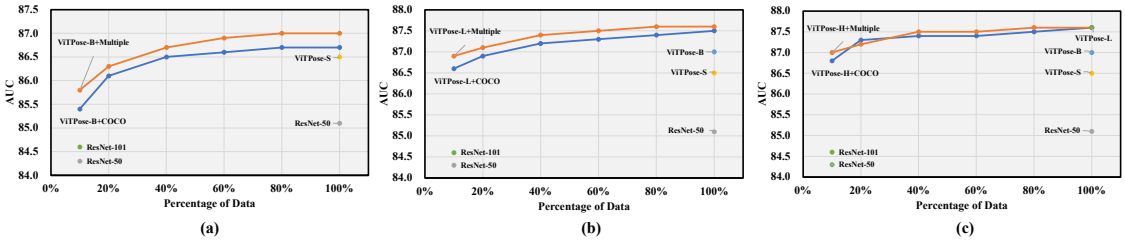


FIGURE 4.12. The transfer learning results of ViTPose-B (a), ViTPose-L (b), and ViTPose-H (c) on the Interhand2.6M dataset using different percentages of training data. The red and blue curves denote the results of ViTPose at the multi-task and single-task training settings. We also plot the results of small pose estimation models, *e.g.*, SimpleBaseline [184] with ResNet-50 and ResNet-101, for reference.

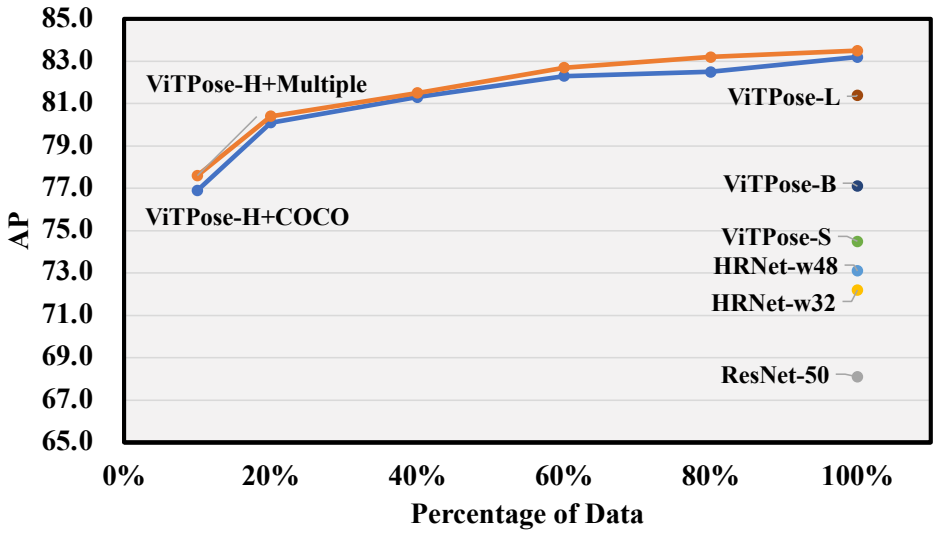


FIGURE 4.13. The transfer learning results of of ViTPose-H on the AP-10K [206] using different percentages of training data. We also plot the results of smaller models for comparison.

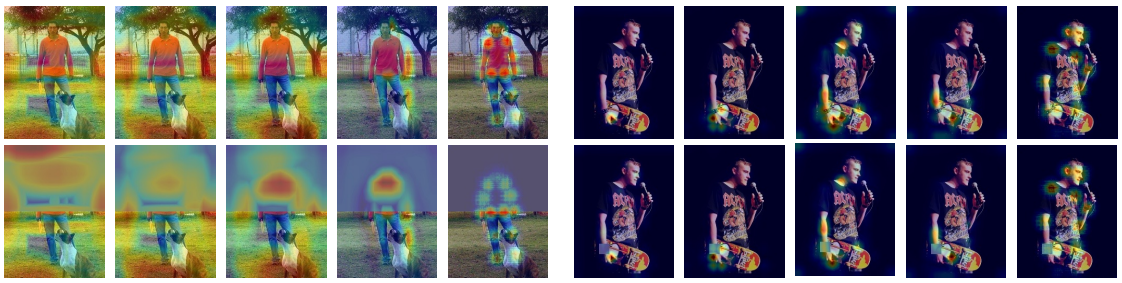


FIGURE 4.14. Visualization of the (a) feature maps and (b) attention maps from ViTPose++-B on two test images w/ and w/o manual masks.

further mitigates the interference of the background and focuses on the keypoint locations. When comparing the feature maps extracted from the normal images (1st row) and masked images (2nd row), the features are almost the same in the visible parts. In contrast, ViTPose gradually guesses the locations of those invisible joints for the masked parts. For example, although the heads of the human are masked, ViTPose can still guess the locations of the head and the keypoints based on the visible parts, as demonstrated in the upper parts of the feature maps from the 9th and 12th layers.

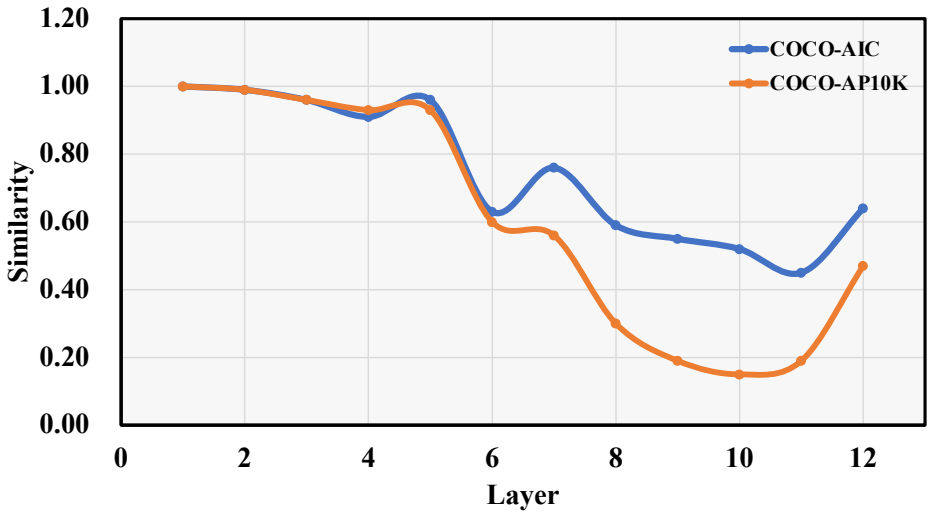
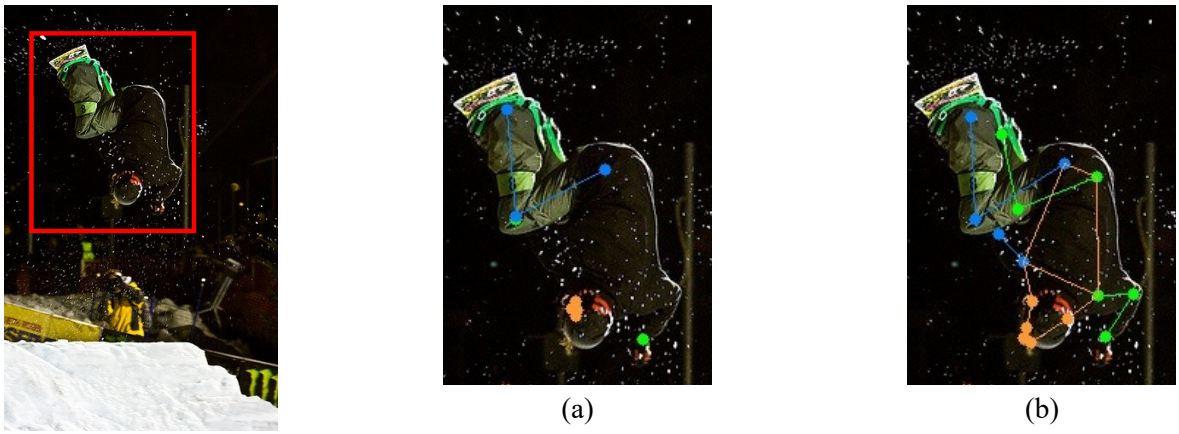


FIGURE 4.15. The similarity of task-specific FFN weights of ViTPose++-B, *i.e.*, COCO v.s. AIC and COCO v.s. AP-10K.

Visualization of attention maps To further inspect how ViTPose models the relationships between different keypoints, we visualize the attention maps regarding the right wrist from the 3rd, 6th, 9th, and 12th layers of ViTPose++-B. We test on both normal images (1st row) and images with the right wrist masked (2nd row). The results are shown in Fig. 4.14 (b). It can be observed that the attention map is rather similar no matter whether the right wrist is masked or not, especially in the deep layers. For example, in the 12th layer, ViTPose pays attention to the person's right arm to help localize the locations of the right wrist in both cases. These visualization results present an intuitive explanation of how ViTPose leverages the learned relationship between different keypoints to facilitate the prediction of a specific (invisible) keypoint.

Task correlation. We further evaluate the correlation between different types of body pose estimation tasks by calculating the similarity between the weights of the task-specific FFN parts. The weight similarity is defined as the cosine similarity between the weights of the FFN layer corresponding to each dataset, respectively. We analyze the per-layer similarity of ViTPose++-B model. The results are plotted in Fig. 4.15. It can be observed that the task-specific FFNs for human pose estimation have a larger similarity than that of the task-specific FFNs for body pose estimation of different species, *i.e.*, the similarity between MS COCO and AIC is larger than that between MS COCO and AP-10K, especially in the deeper layers,



(a) (b)

FIGURE 4.16. Failure case analysis on a test image with an extreme pose. (a) Result of ViTPose-S. (b) Results of ViTPose++-H.

implying that there probably exist conflicts between different types of body pose estimation tasks. Nevertheless, for the shallow layers, the weights of FFN layers are rather similar in both cases, implying that FFNs in the deeper layers are more task-specific than those in the shallower layers, and thus we may only adopt MoE in the deeper FFN layers.

4.4.9 Failure case analysis

Despite the superior performance of ViTPose on various pose estimation tasks, there are still challenges when it comes to handling extreme postures, such as skiing as in Fig. 4.16. Fortunately, due to the scalability of ViTPose, these limitations can be mitigated by using a larger backbone and incorporating more data during training. For instance, the ViTPose++-H model, trained with an expanded dataset containing a greater variety of pose data, can estimate the pose correctly.

4.5 Limitations and discussion

Despite the good properties and performance of our ViTPose and ViTPose++ for body pose estimation, even without elaborate structural designs, there is still room to improve them further. First, we only focus on the pose estimation task in this study, while designing a unified foundation model for simultaneous object detection (tracking) and pose estimation

is more appealing for generic body pose estimation and tracking. Besides, we only use the training data in the visual modality in this study, while leveraging language knowledge of body keypoints to help body pose estimation, especially for zero-shot generalization on unseen species or keypoints, is also worth further exploration.

4.6 Conclusion

This paper presents ViTPose and ViTPose++ as the simple baseline for body pose estimation. They demonstrate good properties, including simplicity, scalability, flexibility, and transferability, which have been well justified through extensive experiments on the representative benchmarks, including MS COCO, AIC, MPII, OCHuman, COCO-W, AP-10K, and APT-36K. New performance records on these datasets have also been set by the proposed models. We hope this work could provide useful insights to the community and inspire more future studies on the potential of developing vision transformers towards a foundation model for generic pose estimation and beyond.

CHAPTER 5

Future Research and Conclusion

5.1 Open Challenges and Future Research

This thesis introduces the infusion of inductive bias into vision transformers by means of innovative architectural designs and refined training methodologies. As a starting point, a foundational baseline model is presented, showcasing the prowess of vision transformers in the realm of pose estimation tasks. However, it is important to acknowledge that several open challenges remain on the horizon, ripe for future exploration and investigation.

5.1.1 More kinds of inductive bias

The core emphasis of this thesis primarily revolves around two distinct forms of inductive bias: locality and scale invariance. Through comprehensive exploration, it substantiates their scalability and notable efficacy within the context of vision transformers via the proposed ViTAE model. Nevertheless, the landscape of inductive bias in vision transformers remains rich and ripe for further investigation. For instance, as illuminated in a recent study [189], transformer models pre-trained using different methodologies exhibit a diverse spectrum of attention distance distributions. This phenomenon serves as an indicator of various forms of inductive bias. To illustrate, models trained via supervised learning tend to prioritize neighboring regions within shallow layers, while expanding their focus to distant regions in deeper layers. This implies that inductive bias towards locality plays a pivotal role in shallow layers. Conversely, when employing self-supervised pre-training methods like masked image modeling [56], attention distance distribution evens out across all layers. This suggests that the capability to concurrently model objects of differing scales becomes particularly pertinent

for models trained through self-supervised learning. In light of these findings, a critical imperative emerges: the identification of the most paramount form of inductive bias for specific vision tasks executed by vision transformers. This underscores the need to discern which bias holds the utmost significance in guiding the performance of vision transformers in distinct vision-based assignments. For example, for tasks related to remote sensing, the inductive bias to explicitly model rotational angles will play an important role in improving performance due to the greater flexibility in capturing viewpoints, whereas simply injecting localization or scale invariance may not be optimal.

5.1.2 More efficient pre-training methods

The paradigm of pre-training followed by fine-tuning has become the de facto approach in deploying vision models, largely due to the immense computational demands associated with training large-scale models from scratch. In this thesis, we exemplify that through meticulous pre-training, models imbued with acquired inductive bias can seamlessly transition to diverse downstream vision tasks. This illuminates the pivotal role played by pre-training methods in unlocking the potential inherent within vision transformers. In the initial stages, early methods predominantly employed fully supervised learning to steer model learning. However, the simplicity of this approach struggles to accommodate the resource-intensive nature of large-scale data annotation for pre-training. Contrastive learning, in contrast, alleviates data requisites and has been explored via diverse avenues. These encompass techniques such as freezing the first layer within MoCov3 [29] and implementing centering operations as observed in DINO [18]. More recently, taking inspiration from masked language modeling pre-training in the realm of natural language, the masked image modeling (MIM) pre-training method has been devised [56]. This methodology involves randomly obscuring portions of images through masking, prompting the network to reconstruct the image. To optimize training efficiency, diverse masking strategies have been investigated, including random masking and attention-guided masking. Furthermore, the reconstruction target has emerged as a crucial area of study, with various alternatives being employed [6, 179, 224]. These encompass VAE tokenizers, HOG features, and signals from pre-trained models. The exploration of more

efficient reconstruction targets during the pre-training of vision transformers stands as an imperative avenue for further research.

5.1.3 Extension to multi-modality learning

It has been observed that vision transformers have obtained superior performance on a series of vision tasks like the pose estimation task. However, they are difficult to generalize to unseen classes due to the use of a pre-training and fine-tuning pipeline, where task-specific data is combined with a task-specific design (*e.g.*, the task-specific decoder in ViTPose). Utilizing the information from another modality is a possible strategy to enhance the model's generalization ability. It is important to note that in contrast to CNN models tailored specifically for visual tasks, transformer architectures inherently possess the adaptability to accommodate diverse modality inputs, *e.g.*, spanning both visual and language domains. Historically, strategies such as those outlined in [134, 44] have been employed to reconcile different modalities. This often involves utilizing distinct backbones for each modality and aligning them through loss functions. However, these methods tend to overlook granular information since each modality is modeled in isolation. To enhance inter-modality interaction, contemporary techniques integrate both modalities by jointly inputting them into transformer models for feature extraction. Various tokenizers are employed to map data from disparate modalities into a shared token space, and the design of these tokenizers constitutes a critical facet in multi-modality pre-training. Early approaches [76] leverage simple structures like a single tokenizer layer, often comprising a fully-connected or convolutional layer. Nonetheless, learning meaningful interactions in shallow layers proves challenging due to the relatively limited semantic information available in both modalities. To address this, recent methodologies, exemplified by CoCa [207], introduce stacked transformer layers as tokenizers. This enables embedding both modalities into feature representations, subsequently leveraging cross-modality transformers to further align these features. With the emergence of large-language models (LLMs), novel techniques like those found in [102] endeavor to incorporate visual features into LLMs by projecting them from pre-trained vision models to the language feature space via projection

layers. Examples include the Q-Former utilized in BLIP-2 [88] and cross-attention mechanisms employed in Flamingo [3]. A key challenge moving forward is the design of innovative structures adept at extracting features from disparate modalities, coupled with the formulation of novel loss functions to effectively align these features.

5.1.4 Unified structures for different vision tasks

Different from natural language processing where versatile tasks can be tackled using pre-trained language models by manipulating prompts, distinct decoders are typically essential to adapt pre-trained vision models to various vision tasks. These decoders, such as classifier layers for classification tasks or feature pyramid networks for detection and segmentation, necessitate random initialization and updating using task-specific training sets. Could it be possible to establish a unified model structure that seamlessly accommodates multiple vision tasks, leveraging pre-trained models directly? This aspiration proves challenging for CNN models, as they lack the capacity to accommodate auxiliary input information like language descriptions or exemplars. However, the inherent flexibility of transformer models offers a potential avenue for such unification. Certain efforts have ventured into using exemplars to guide network learning, aiming to harmonize diverse vision tasks through language-image or image-image exemplars. Crafting these methods necessitates meticulous attention to training strategies and exemplar selection. An alternative approach leverages language models to process task-specific instructions. For instance, LLaVA [102] harnesses pre-trained language models to guide visual token processing with instructions, enabling the network to undertake various image comprehension tasks. Similar strategies have been explored in [219, 221, 230]. However, this strategy hinges on generating sets of task-specific descriptions, where prompts play a pivotal role, and the design of the projection layer warrants investigation. In these methodologies, pre-trained vision transformers serve as vision encoders, with no further fine-tuning specific to the tasks at hand. Yet, the efficacy of this seemingly straightforward approach in furnishing task-specific features for a diverse range of vision tasks is uncertain, especially considering the noticeable disparity between the zero-shot classification performance of these models and their fully fine-tuned counterparts.

Consequently, it becomes imperative to delve into improved strategies that can more effectively adapt pre-trained vision models to an array of distinct vision tasks.

5.2 Conclusion

In this thesis, we advance vision transformers by introducing inductive bias into them via both structural design and training methods. Specifically, a novel backbone model ViTAE (Chapter 2) is proposed with a parallel convolution branch for modeling locality and a pyramid reduction module for capturing scale-invariance inductive bias. For the training method, a region-swapping strategy is discussed in RegionCL (Chapter 3) to further promote the locality information in contrastive pre-training. The performance of the pre-trained vision transformers is demonstrated in pose estimation tasks via a novel baseline model ViTPose (Chapter 4), where the ViTPose-G model with ViTAE-G backbone obtains the champion in the challenging MS COCO, MPII, and OCHuman datasets without ensemble. Extensive experiments and ablation studies are conducted to demonstrate the effectiveness of the proposed methods. We hope this thesis can provide some insights for the community to emphasize the inductive bias in vision transformers and help the community to further develop transformers in diverse vision tasks.

References

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3686–3693, 2014.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[8] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

[9] Yanrui Bin, Xuan Cao, Xinya Chen, Yanhao Ge, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Changxin Gao, and Nong Sang. Adversarial semantic data augmentation for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–622, 2020.

[10] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[11] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987.

[12] Likun Cai, Zhi Zhang, Yi Zhu, Li Zhang, Mu Li, and Xiangyang Xue. Bigdetection: A large-scale benchmark for improved object detector pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4777–4787, 2022.

[13] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xinyu Zhou, Erjin Zhou, Xiangyu Zhang, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[14] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 194–203, 2021.

[15] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[16] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[17] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[18] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[19] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021.

[20] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[21] Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7546–7554, 2021.

[22] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[23] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, volume 1, pages 1597–1607. PMLR, 2020.

[25] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020.

[26] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[27] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[28] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[29] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[30] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7103–7112, 2018.

[31] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5386–5395, 2020.

[32] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021.

[33] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[34] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[35] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[36] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[38] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[39] Hasan Demirel and Gholamreza Anbarjafari. Image resolution enhancement by using discrete and stationary wavelet decomposition. *IEEE transactions on image processing*, 20(5):1458–1460, 2010.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Unsupervised pretraining for object detection by patch reidentification. *arXiv preprint arXiv:2103.04814*, 2021.

[43] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv preprint arXiv:2105.01883*, 2021.

[44] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10995–11005, 2023.

[45] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[46] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *arXiv preprint*

arXiv:2106.09681, 2021.

[47] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[48] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017.

[49] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[50] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[51] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

[52] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[53] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358*, 2021.

[54] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006.

[55] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

[56] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*,

2021.

[57] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[58] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

[59] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[62] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[63] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

[64] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

[65] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*, 2021.

[66] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[67] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[68] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[69] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[70] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

[71] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[72] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[73] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 196–214, 2020.

[74] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 2, pages II–II. IEEE, 2004.

[75] Rawal Khirodkar, Visesh Chari, Amit Agrawal, and Ambrish Tyagi. Multi-instance pose networks: Rethinking top-down pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3122–3131, 2021.

[76] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[77] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[78] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11977–11986, 2019.

[79] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[81] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[82] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[83] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 577–593. Springer, 2016.

[84] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[86] Chunyuan Li, Xiujun Li, Lei Zhang, Baolin Peng, Mingyuan Zhou, and Jianfeng Gao. Self-supervised pre-training with hard examples improves visual representations. *arXiv preprint arXiv:2012.13493*, 2020.

[87] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10863–10872, 2019.

[88] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[89] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2020.

[90] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1944–1953, June 2021.

[91] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[92] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[93] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[94] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[95] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

[96] Zheng Li, Jingwen Ye, Mingli Song, Ying Huang, and Zhigeng Pan. Online knowledge distillation for efficient pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11740–11750, 2021.

[97] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

[98] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[99] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[100] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[101] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Rui Qian, Tao Wang, Ning Xu, Hongkai Xiong, Guo-Jun Qi, and Nicu Sebe. Human in events: A large-scale benchmark for human-centric video analysis in complex events. *arXiv preprint arXiv:2005.04490*, 2020.

[102] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[103] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2020.

[104] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[105] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.

[106] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, 2022.

[107] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[108] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[109] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[110] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[111] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, volume 29, pages 4898–4906, 2016.

[112] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[113] Luke Melas-Kyriazi. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv: Computer Vision and Pattern Recognition*, 2021.

[114] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6707–6717, 2020.

[115] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 548–564, 2020.

[116] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307, 2017.

[117] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, 2017.

[118] Alejandro Newell, Zhiao Huang, and Jia Deng. Segvit: Semantic segmentation with plain vision transformers. In *Zhang, Bowen and Tian, Zhi and Tang, Quan and Chu, Xiangxiang and Wei, Xiaolin and Shen, Chunhua and Liu, Yifan*, 2022.

[119] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[120] Pauline C Ng and Steven Henikoff. Sift: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003.

[121] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6951–6960, 2019.

[122] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[123] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–84. Springer, 2016.

[124] Hannu Olkkonen and Peitsa Pesola. Gaussian pyramid wavelet transform for multiresolution analysis of images. *Graphical Models and Image Processing*, 58(4):394–398, 1996.

[125] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[126] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4903–4911, 2017.

[127] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[128] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.

[129] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[130] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. *arXiv preprint arXiv:2105.03889*, 2021.

[131] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, June 2021.

[132] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmalek, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. *arXiv preprint arXiv:2011.05499*, 2020.

[133] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937, 2016.

[134] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[135] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[136] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[137] Umer Rafi, Andreas Doering, Bastian Leibe, and Juergen Gall. Self-supervised keypoint correspondences for multi-person pose estimation and tracking in videos. In *Proceedings of the European conference on computer vision (ECCV)*, 2020.

[138] Umer Rafi, Bastian Leibe, Juergen Gall, and Ilya Kostrikov. An efficient convolutional network for human pose estimation. In *BMVC*, volume 1, page 2, 2016.

[139] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[140] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1144–1153, 2021.

[141] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE international conference on computer vision*, pages 2564–2571. Ieee, 2011.

[142] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[143] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[144] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on*

computer vision, pages 618–626, 2017.

[145] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[146] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2016.

[147] Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. *arXiv preprint arXiv:2003.05438*, 2020.

[148] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.

[149] Leonid Sigal. Human pose estimation. In *Computer Vision: A Reference Guide*, pages 573–592. Springer, 2021.

[150] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[151] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.

[152] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7262–7272, 2021.

[153] Zhihui Su, Ming Ye, Guohui Zhang, Lei Dai, and Jianda Sheng. Cascade feature aggregation for human pose estimation. *arXiv preprint arXiv:1902.07837*, 2019.

[154] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5693–5703, 2019.

[155] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[156] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[157] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[158] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[159] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[160] Chenxin Tao, Honghui Wang, Xizhou Zhu, Jiahua Dong, Shiji Song, Gao Huang, and Jifeng Dai. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. *arXiv preprint arXiv:2112.05141*, 2021.

[161] Ajinkya Tejankar, Soroush Abbasi Koohpayegani, Vipin Pillai, Paolo Favaro, and Hamed Pirsiavash. Isd: Self-supervised learning by iterative similarity distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9609–9618, 2021.

[162] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 776–794. Springer, 2020.

[163] Yuandong Tian. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022.

[164] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and

Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[165] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014.

[166] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

[167] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[168] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[169] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Grafit: Learning fine-grained image representations with coarse labels. *arXiv preprint arXiv:2011.12982*, 2020.

[170] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

[171] Chaoyue Wang, Chang Xu, and Dacheng Tao. Self-supervised pose adaptation for cross-domain image animation. *IEEE Transactions on Artificial Intelligence*, 1(1):34–46, 2020.

[172] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[173] Pichao Wang, Xue Wang, Hao Luo, Jingkai Zhou, Zhipeng Zhou, Fan Wang, Hao Li, and Rong Jin. Scaled relu matters for training vision transformers. In *Proceedings of*

the AAAI Conference on Artificial Intelligence, volume 36, pages 2495–2503, 2022.

[174] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[175] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 568–578, 2021.

[176] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations*, 2022.

[177] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[178] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3024–3033, 2021.

[179] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[180] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[181] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[182] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017.

[183] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2018.

[184] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[185] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[186] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8392–8401, 2021.

[187] Jiangtao Xie, Ruiren Zeng, Qilong Wang, Ziqi Zhou, and Peihua Li. So-vit: Mind visual tokens for vision transformer. *arXiv preprint arXiv:2104.10935*, 2021.

[188] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[189] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[190] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[191] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16684–16693, 2021.

[192] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[193] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9981–9990, October 2021.

[194] Yufei Xu, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Dut: Learning video stabilization by simply watching unstable videos. *IEEE Transactions on Image Processing*, 2022.

[195] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 2022.

[196] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021.

[197] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Regioncl: Exploring contrastive region pairs for self-supervised representation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[198] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint arXiv:2104.13497*, 2021.

[199] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3987–3996, 2021.

[200] Junwei Yang, Ke Zhang, Zhaolin Cui, Jinming Su, Junfeng Luo, and Xiaolin Wei. Inscon: Instance consistency feature representation via self-supervised learning. *arXiv preprint arXiv:2203.07688*, 2022.

[201] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on*

Computer Vision (ICCV), 2021.

[202] Yuxiang Yang, Junjie Yang, Yufei Xu, Jing Zhang, Long Lan, and Dacheng Tao. Apt-36k: A large-scale benchmark for animal pose estimation and tracking. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[203] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[204] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR 2016 : International Conference on Learning Representations 2016*, 2016.

[205] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[206] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[207] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[208] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

[209] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[210] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[211] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *Advances in Neural Information Processing Systems*, 2021.

[212] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[213] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[214] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7093–7102, 2020.

[215] Jing Zhang, Zhe Chen, and Dacheng Tao. Towards high performance human keypoint detection. *International Journal of Computer Vision*, 129(9):2639–2662, 2021.

[216] Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020.

[217] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022.

[218] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: Learning varied-size window attention in vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[219] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

[220] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 649–666. Springer, 2016.

[221] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023.

[222] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–898, 2019.

[223] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[224] Xinyu Zhang, Jiahui Chen, Junkun Yuan, Qiang Chen, Jian Wang, Xiaodi Wang, Shumin Han, Xiaokang Chen, Jimin Pi, Kun Yao, et al. Cae v2: Context autoencoder with clip target. *arXiv preprint arXiv:2211.09799*, 2022.

[225] Yuexi Zhang, Yin Wang, Octavia Camps, and Mario Sznaier. Key frame proposal network for efficient pose estimation in videos. In *EProceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[226] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[227] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10160–10169, 2021.

[228] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[229] Jingkai Zhou, Pichao Wang, Fan Wang, Qiong Liu, Hao Li, and Rong Jin. Elsa: Enhanced local self-attention for vision transformer. *arXiv preprint arXiv:2112.12786*, 2021.

[230] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[231] Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems*, 2022.

[232] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.