International Journal of Computer Vision (2024) 132:2929–2942
https://doi.org/10.1007/s11263-024-01988-x

Training Object Detectors from Scratch: An Empirical Study in the Era of Vision Transformer

Weixiang Hong[1,2] · Wang Ren[1] · Jiangwei Lao[1] · Lele Xie[1] · Liheng Zhong[1] · Jian Wang[1] · Jingdong Chen[1] · Honghai Liu[2,3] · Wei Chu[1]

Received: 6 May 2022 / Accepted: 1 January 2024 / Published online: 26 February 2024
© The Author(s) 2024

Abstract
Modeling in computer vision has long been dominated by convolutional neural networks (CNNs). Recently, in light of the excellent performance of self-attention mechanism in the language field, transformers tailored for visual data have drawn significant attention and triumphed over CNNs in various vision tasks. These vision transformers heavily rely on large-scale pre-training to achieve competitive accuracy, which not only hinders the freedom of architectural design in downstream tasks like object detection, but also causes learning bias and domain mismatch in the fine-tuning stages. To this end, we aim to get rid of the "pre-train and fine-tune" paradigm of vision transformer and train transformer based object detector from scratch. Some earlier works in the CNNs era have successfully trained CNNs based detectors without pre-training, unfortunately, their findings do not generalize well when the backbone is switched from CNNs to a vision transformer. Instead of proposing a specific vision transformer based detector, in this work, our goal is to reveal the insights of training vision transformer based detectors from scratch. In particular, we expect those insights to help other researchers and practitioners, and inspire more interesting research in other fields, such as remote sensing, visual-linguistic pre-training, etc. One of the key findings is that both architectural changes and more epochs play critical roles in training vision transformer based detectors from scratch. Experiments on the MS COCO dataset demonstrate that vision transformer based detectors trained from scratch can also achieve similar performance to their counterparts with ImageNet pre-training.

Keywords Vision transformer · Object detection · Training from scratch · Large-scale pre-training · Convolutional neural networks · Detection performance and efficiency

1 Introduction

The extraordinary performance of AlexNet (Krizhevsky et al., 2012) on the ImageNet image classification challenge has sparked the passion in convolutional neural networks (CNNs), and led to a variety of powerful CNN backbones through greater scale (He et al., 2016), more extensive connections (Huang et al., 2017), and more sophisticated forms of convolution (Dai et al., 2017). Consequently, modeling in computer vision has long been dominated by CNNs, until the Transformer architecture (Devlin et al., 2019) was recently

Communicated by Esa Rahtu.

✉ Weixiang Hong
hwx229374@antgroup.com; weixiang.hong@port.ac.uk

Wang Ren
renwang.rw@antgroup.com

Jiangwei Lao
wenshuo.ljw@antgroup.com

Lele Xie
yule.xll@antgroup.com

Liheng Zhong
zhongliheng.zlh@antgroup.com

Jian Wang
bobblair.wj@antgroup.com

Jingdong Chen
jingdongchen.cjd@antgroup.com

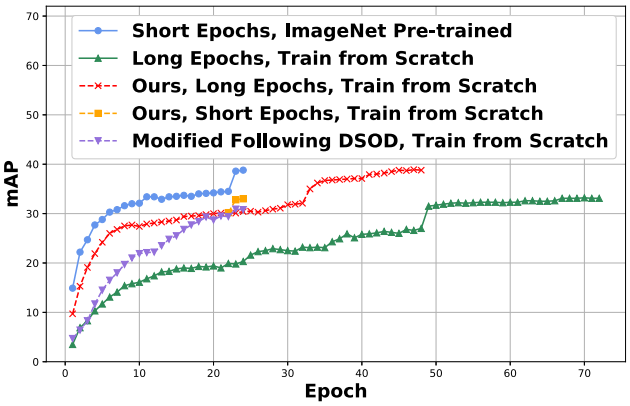
Honghai Liu
honghai.liu@port.ac.uk

Wei Chu
weichu.cw@antgroup.com

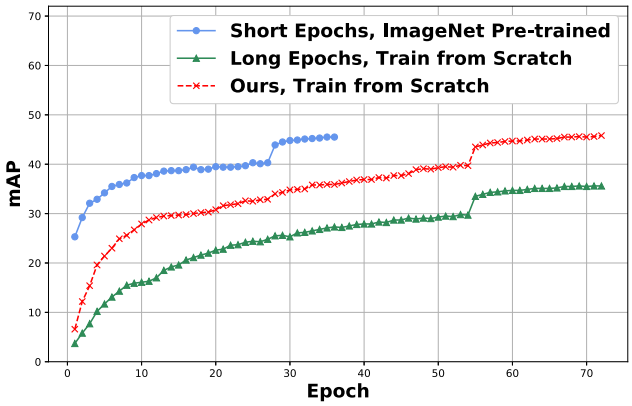
[1] Ant Group, Hangzhou, China

[2] University of Portsmouth, Portsmouth, UK

[3] Harbin Institute of Technology, Shenzhen, China



(a) Train Swin-T based FCOS [40].



(b) Swin-T based Faster R-CNN [32].

Fig. 1 We train and evaluate Swin-T (Liu et al., 2021) based detectors (FCOS (Tian et al., 2019) and faster R-CNN (Ren et al., 2015)) on the COCO dataset. We observe that: (1) Swin-T based detectors trained from scratch do not achieve comparable mAP to their ImageNet pre-trained counterpart, even if more epochs of training are conducted following He et al. (He et al., 2019). (2) The results of Swin-T based FCOS will increase if its architecture is modified following DSOD (Shen et al., 2017), which is originally proposed to boost the proposal-free CNNs based detector when pre-training is unavailable. However, the performance of "Swin-T + FCOS + DSOD" detector trained from scratch is still not as good as the ImageNet pre-trained one. (3) With suitable architectural changes and sufficient training epochs, the proposed vision transformer based detectors without pre-training demonstrate competitive mAP to their ImageNet pre-trained counterparts (Color figure online)

adapted from natural language processing (NLP) to vision community. A group of transformers tailored for visual data has triumphed over numerous CNN-based methods in many vision tasks (e.g. , image classification (Dosovitskiy et al., 2021), object detection (Carion et al., 2020), semantic segmentation (Cheng et al., 2021), etc). Among them, object detection is one of the fastest-moving areas due to its wide applications in surveillance, autonomous driving, etc.

Most of the advanced object detectors require initialization from large-scale pre-training to achieve good performance, regardless of whether their backbones are CNNs or vision transformers (Ren et al., 2015; Liu et al., 2021). Typically, these methods first pre-train the backbone model on ImageNet (Russakovsky et al., 2015) dataset, then fine-tune the pre-trained weights on the specific object detection task. Fine-tuning from pre-trained models has at least two advantages. First, it is convenient to reuse various state-of-the-art deep models that are publicly available. Second, fine-tuning can quickly generate the final model and requires much fewer annotated training samples than the classification task. The fine-tuning process can also be viewed as an instance of transfer learning (Pan & Yang, 2010).

However, there are also critical limitations when adopting pre-trained networks in object detection: (1) **Limited structure design space** (Shen et al., 2017). The pre-trained models are usually cumbersome (containing a huge number of parameters) for performing well on the ImageNet classification task. Existing object detectors directly adopt the pre-trained networks, resulting in little flexibility to control/adjust the network structures. The requirement of computing resources is also bounded by the complex pre-trained networks. (2) *Learning bias* (Xu et al., 2021). Both the loss functions and category distributions differ between classification and detection tasks, leading to different searching/optimization spaces. Thus, learning may be biased towards a local minimum for detection tasks. (3) *Domain mismatch* (Gupta et al., 2016). Though fine-tuning can mitigate the gap between different target category distributions, it is still a severe problem when the source domain (ImageNet) has a huge mismatch with the target domain such as satellite remote sensing, depth camera, medical images, etc.

Some earlier works have studied training CNNs based object detection networks from scratch (Shen et al., 2017; He et al., 2019). Specifically, DSOD, abbreviated for deeply supervised object detector (Shen et al., 2017), argues that only proposal-free detectors can be trained from scratch, though proposal-based methods like faster R-CNN (Ren et al., 2015) often have superior performance than proposal-free ones. In detail, DSOD (Shen et al., 2017) augments the original detector by deep supervision, stem block and dense prediction, etc., to achieve ideal detection performance. In contrast, He et al. (2019) point out that no architectural change is required for training from scratch. As long as sufficient training iterations are executed, detectors trained from scratch can converge to similar accuracy to their ImageNet pre-training counterparts.

Given the fact that vision transformers have outperformed CNNs in numerous computer vision tasks, we are motivated to raise the following two questions: (1). Do the findings (Shen et al., 2017; He et al., 2019) obtained on CNNs based detectors remain effective in the era of vision

transformer? (2). If not, is it still possible to train vision transformer based object detectors from scratch?

In this work, we experimentally answer the two questions above in Sect. 3 and Sect. 4. Specifically, we first show that naively applying the experiences from Shen et al. (2017) and He et al. (2019) to vision transformer is not enough. On the one hand, we modify the architecture of Swin-T (Liu et al., 2021) based FCOS following DSOD (Shen et al., 2017), and observe boosted detection performance (the purple curve in Fig. 1a). Unluckily, the modified FCOS is still not as good as the pre-trained vanilla one (the blue curve); On the other hand, we follow He et al. (He et al., 2019) to apply more training epochs to two Swin-T (Liu et al., 2021) based detectors FCOS and Faster R-CNN, and illustrate the performance of these two detectors by the green curves in Fig. 1a, b. Given sufficient training epochs, both two detectors trained from scratch cannot converge to comparable solutions to their pre-trained counterparts (blue curves). Based on the phenomena above, we conjecture that: if either architectural changes or more training epochs are solely applied, vision transformer based detectors that are trained from scratch will achieve inferior results compared to their pre-trained counterparts.

Thus, instead of proposing a specific vision transformer based detector, we aim to reveal the insights of training vision transformer based detectors from scratch. In particular, we find that both architectural changes and more epochs are important in training vision transformer based detectors from scratch. Together with several other techniques, we manage to train transformer based detectors from scratch and achieve competitive results to the ImageNet pre-trained counterpart, as shown by the red curves in Fig. 1a, b. We experimentally validate the generality of our findings to several advanced vision transformers for detection task, and anticipate that these insights will assist other researchers and practitioners, inspiring further research in fields such as remote sensing, visual-linguistic pre-training, etc.

Our main findings are summarized as follows:

1. *From RoIPooling to RoIAlign* We observe that proposal-based and proposal-free detectors exhibit distinct behavior when trained from scratch, that is, proposal-free detectors degrade less than proposal-based ones compared to their pre-trained counterparts. We find out this phenomenon is essentially caused by RoIPooling, i.e., it hinders the gradient from being smoothly back-propagated to backbone layers. We address this problem by replacing RoIPooling with RoIAlign, and achieve consistencies between proposal-based and proposal-free detectors when trained from scratch.
2. *From T-T-T-T to C-C-T-T* Recent studies have revealed that large-scale pre-training essentially makes lower attention layers learn inductive bias and "act like convolutions" (Raghu et al., 2021). Thus, we replace the first two stages of vision transformers with convolution blocks, namely, from T-T-T-T to C-C-T-T, where T and C stand for transformer and convolution block, respectively. Such a replacement directly introduces the inductive prior of convolution into the backbone model, making it less dependent on ImageNet pre-training.
3. **Gradient Calibration**. In C-C-T-T architecture, we observe that the convolution and self-attention layers exhibit significant differences in terms of the scale of gradient. Since it is better to adjust all of the layers a little rather than to adjust just a few layers a large amount (Yosinski et al., 2014), we propose to calibrate the gradients of our model, and achieve better convergence property.
4. *More training epochs* As argued by He et al. (He et al., 2019), it is unrealistic and unfair to expect models trained from random initialization to converge as fast as those initialized from ImageNet pre-training. Typical ImageNet pre-training can learn not only semantic information, but also low-level features (e.g., edges, textures) that do not need to be re-learned during fine-tuning. Therefore, models trained from scratch must be trained for more epochs than typical fine-tuning schedules.

2 Related Work

2.1 Vision Transformer

Convolutional neural networks have been the dominating architectures for many computer vision tasks Krizhevsky et al. (2012); He et al. (2016). Inspired by the recent success of self-attention mechanism (Vaswani et al., 2017) in natural language field, there is growing interest in exploiting transformer architecture for vision tasks. The pioneering work ViT (Dosovitskiy et al., 2021) directly applies a transformer architecture on non-overlapping image patches for image classification. This approach achieves an impressive speed-accuracy trade-off in image classification compared to CNNs. Later work such as (Touvron et al., 2021; Han et al., 2021; Liu et al., 2021) has made significant progress in modifying the ViT architecture for better performance. Particularly, Swin Tranformer (Liu et al., 2021) achieves state-of-the-art results on various tasks, including object detection, semantic segmentation, etc. Our analysis of training a vision transformer based detector will be based on Swin Transformer.

2.2 Combining Vision Transformer and Convolution

Generally speaking, convolutional layers tend to have faster converging rate thanks to their strong prior of inductive bias, while attention layers exhibit higher model capacity that can



Fig. 2 Qualitative comparisons between the detection results from faster R-CNN (Ren et al., 2015) models that are naively trained from scratch, and trained using our method. The bottom row (ours) demonstrates significantly better detection performance in terms of both recall and precision

benefit from large-scale pre-training (Raghu et al., 2021). To achieve the balance of inductive prior and model capacity, some pioneering works have attempted to combine convolutional and attention layers. For example, Conformer (Peng et al., 2021) proposes a feature coupling unit to fuse the features extracted by convolutional and self-attention layers, ConViT (d'Ascoli et al., 2021) introduces gated positional self-attention to equip vision transformer with a "soft" convolutional inductive bias. BotNet (Srinivas et al., 2021) replaces the spatial convolutions with global self-attention in the final three bottleneck blocks of a ResNet (He et al., 2016). CvT (Wu et al., 2021) designs a hierarchy of transformers containing a convolutional token embedding, and a convolutional self-attention block leveraging a convolutional projection.

It is worth noting that CMT (Guo et al., 2022) places a LPU (essentially depth-wise convolution plus residual connection) ahead of every multi-head self-attention layer, and manages to train image classifiers from scratch on ImageNet dataset (Russakovsky et al., 2015). Nevertheless, ImageNet dataset (Russakovsky et al., 2015) has more than 10× training images than COCO (Lin et al., 2014) (1.28M v.s. 118K), and object detection is regarded as a more challenging task than image classification (He et al., 2017). Thus, it is considerably harder to train object detector on COCO (Lin et al., 2014) than to train image classifier on ImageNet (Russakovsky et al., 2015), both from scratch. The fact that CMT (Guo et al., 2022) is trained from scratch on ImageNet dataset does not imply that our task is trivial.

2.3 Train Object Detection from Scratch

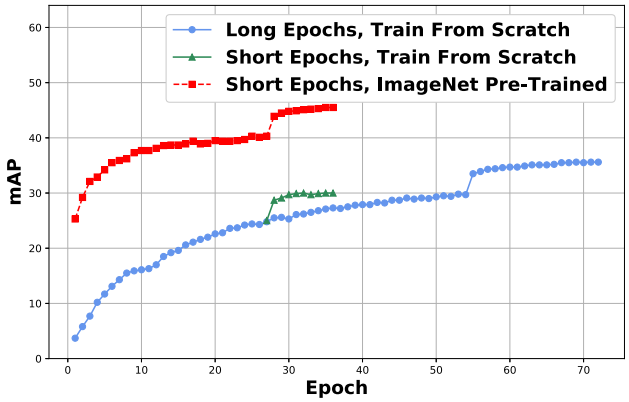
Earlier object detection methods were trained with no pre-training (Matan et al., 1992; Rowley et al., 1996; Szegedy et al., 2013). Given the success of pre-training in R-CNN (Girshick et al., 2014), the "pre-training and fine-tuning" paradigm has become a conventional wisdom in modern CNNs based detectors. Nevertheless, due to the limitations caused by pre-training, research efforts have been continuously devoted to training CNNs based detector from scratch (Shen et al., 2017; He et al., 2019; Li et al., 2018; Law & Deng, 2018). Specifically, DetNet (Li et al., 2018) and CornerNet (Law & Deng, 2018) concentrate on designing detection-specific architectures, which is not the focus of this work. DSOD (Shen et al., 2017) contributes a set of principles that enable detectors to be trained from scratch, but these principles only work for proposal-free methods. He et al. (He et al., 2019) do not require any specific architectural changes, instead, they advocate that training from scratch only requires more iterations to sufficiently converge.

3 Do the Findings Obtained on CNNs Based Detectors Remain Effective?

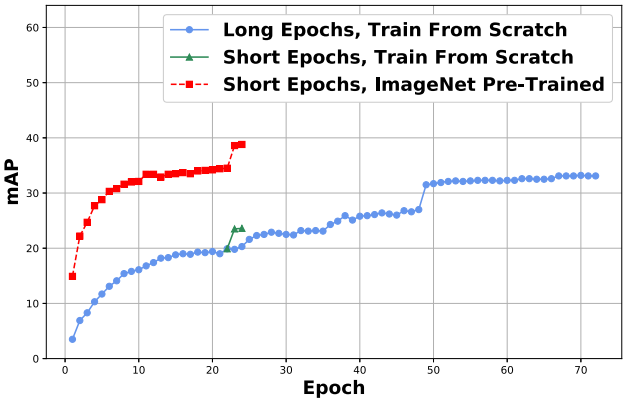
In this section, we experimentally investigate whether DSOD (Shen et al., 2017) and He et al. (He et al., 2019) generalize well to vision transformer based detectors.

Backbone Without loss of generality, we choose the representative work Swin Transformer (Liu et al., 2021) to investigate the generality of (Shen et al., 2017; He et al., 2019) to vision transformer. To be specific, we use Swin-T, an instance of Swin Transformer, as the backbone for all detectors in this section. The complexity of Swin-T is similar to that of ResNet-50 (Liu et al., 2021).

Detectors Modern object detectors can be roughly classified into two categories: proposal-based and proposal-free, depending on whether object proposals are utilized as intermediate results. Generally speaking, proposal-free detectors are more efficient owing to straightforward architectures, but the proposal-based detectors still take the lead in accuracy. In this work, we choose faster R-CNN (Ren et al., 2015) and



(a) Train Faster R-CNN following [18].



(b) Train FCOS following [18].

Fig. 3 Train Swin-T based detectors on COCO dataset following (He et al., 2019). We conduct experiments with Faster R-CNN (Ren et al., 2015) and FCOS (Tian et al., 2019). The green, red and blue curves stand for short epochs without pre-training, short epochs with pre-training and long epochs without pre-training. For both detectors, the extended epochs significantly boosts the detector trained from scratch. Unluckily, the final detection mAP is still inferior to the ImageNet pre-trained counterpart (Color figure online)

FCOS (Tian et al., 2019) as representative of proposal-based and proposal-free detectors. Faster R-CNN (Ren et al., 2015) is the seminal work that innovatively addresses object detection in an end-to-end manner. It first generates a set of region proposals based on pre-defined anchors, then classifies and refines those proposals to obtain final bounding boxes. Thus, faster R-CNN is also regarded as a two-stage detector. In contrast, FCOS (Tian et al., 2019) is a one-stage proposal-free method, which contributes a significantly simplified detection framework. The bounding boxes are directly regressed from the feature map, without involving anchors and proposals.

Dataset All experiments are conducted on the challenging MS COCO (Lin et al., 2014) dataset that includes 80 object classes. Following the common practice (Liu et al., 2016), all $115K$ images in the trainval $35k$ split are used for training, and all $5K$ images in the minival split are used as validation for analysis study.

Training and Inference During training, we resize the input images to keep their shorter side at 800 pixels and their longer side at or below $1,333$ pixels. The whole network is initialized with He method (He et al., 2015) and trained using AdamW (Loshchilov & Hutter, 2019) optimizer with batch size as 16. During the inference phase, we resize the input image in the same way as in the training phase, and forward it through the whole network to output the predicted bounding boxes with predicted classes. Then, the non-maximum suppression (NMS) (Girshick et al., 2014) is applied with the IoU threshold 0.6 per class to generate the final top 100 confident detections per image.

3.1 Train FCOS and Faster-RCNN Following (He et al., 2019)

He et al. (He et al., 2019) argues that training from scratch on target dataset is feasible without architectural changes, and the resulting detection performance is no worse than its ImageNet pre-training counterparts. Since there is no constraint on proposal-based or proposal-free detectors in (He et al., 2019), we extend the training iterations of both FCOS and Faster R-CNN, with Swin-T as their backbones.

Specifically, we train both two detectors with the initial learning rate and monitor the validation set mAP at each epoch. When the mAP reaches saturation, we decay the learning rate and continue to train it until convergence. The experimental results are shown in Fig. 3. The extended training epochs significantly boost the detector trained from scratch, unluckily, the final detection mAP is still inferior to the ImageNet pre-trained counterpart. Also, one can compare the gaps in final mAP between ImageNet pre-trained version and train-from-scratch one, and observe that Faster R-CNN degrades more than FCOS.

3.2 Train FCOS Following (Shen et al., 2017)

Different from He et al. (He et al., 2019), DSOD (Shen et al., 2017) advocates 4 principles for training detectors from scratch, i.e., (1) Proposal-free; (2) deep supervision; (3) stem block; (4) dense prediction. In line with these principles, we made the following modifications to our Swin-T + FCOS detector: (1) FCOS is naturally proposal-free; (2) we add dense connections between stages of Swin-T following (Shen et al., 2017); (3) we change the patchify stem to inception (Szegedy et al., 2016) style. Note that (Xiao et al.,

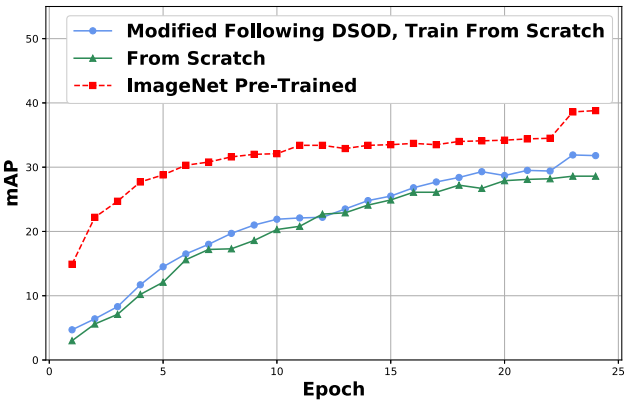


Fig. 4 Train FCOS (Tian et al., 2019) **following DSOD** (Shen et al., 2017). The green, red and blue curves stand for vanilla architecture without pre-training, vanilla architecture with pre-training and modified architecture following DSOD (Shen et al., 2017). The modifications do improve the detection performance when trained from scratch, but the gap to the pre-trained baseline is still significant (Color figure online)

2021) has also emphasized the importance of stem block in vision transformer; (4) For each scale, we only learn half of new feature maps and reuse the remaining half of the previous ones. Besides, we also train the vanilla Swin-T + FCOS, with and without initialization from ImageNet pre-training, so as to provide comparison baselines. All three models are trained for 24 epochs, with the learning rate decay once at the 22nd epoch following (Chen et al., 2019).

The experimental results are shown in Fig. 4. The green and red curves denote the vanilla Swin-T + FCOS, with/without ImageNet pre-training. As expected, the one with pre-training significantly outperforms the counterpart that is trained from scratch, in terms of both convergence rate and final detection mAP. Also, as shown by the blue curve in Fig. 4, the variant modified according to DSOD (Shen et al., 2017) demonstrated improved performance than the vanilla Swin-T + FCOS architecture. Unfortunately, it still has a large gap to the pre-trained version.

3.3 Discussion

The results in Figs. 3 and 4 indicate that the findings in CNNs era, either architectural changes (Shen et al., 2017) or long epochs (He et al., 2019), do not generalize well enough on vision transformer based detectors. However, given the improvement by solely applying (Shen et al., 2017) or (He et al., 2019), it is natural to consider combining the best of two worlds, as is elaborated in the next section.

4 Method

In this section, we present the step-by-step modifications to FCOS and Faster R-CNN, to train both proposal-based and proposal-free detectors from scratch.

4.1 From RoIPooling to RoIAlign

We first investigate the distinct behaviors of proposal-free and proposal-based detectors observed in Sect. 3.1, i.e., Faster R-CNN degrades more than FCOS when switched from "pre-train and fine-tune" to "Train from Scratch". We find that the unsatisfactory performance of Faster R-CNN (Ren et al., 2015) is essentially caused by the internal information loss in RoIPooling (Girshick, 2015). Specifically, RoIPooling involves max pooling on a region of feature maps. It requires the execution of quantization or padding if the coordinates of the RoI are floating-point numbers, or if the region's size cannot be exactly divided by the size of the RoIPooling operator. The quantization or padding inevitably causes information distortion (He et al., 2017), hence hinders the gradients from being smoothly back-propagated from region-level to backbone. The proposal-based methods work well with pre-trained network models because they are well initialized by pre-trained weights, while this is not true for training from scratch.

We empirically find that Faster R-CNN (Ren et al., 2015) can also converge well if we replace RoIPooing (Girshick, 2015) with RoIAlign (He et al., 2017), in which any quantizations of the RoI boundaries or bins are avoided. Instead, bilinear interpolation is exploited to compute the exact values of the output features. We train Swin-T based Faster R-CNN on the COCO dataset, and show the experimental results in Table 1. In the case of "Train from Scratch", RoIAlign achieves 3.7 points higher mAP than RoIPooling. While in the "Pre-Train & Fine-tune" setting, the improvement is relatively tiny, which validates our interpretations above.

Discussion RoIAlign is originally proposed to tackle pixel-based detection tasks in (He et al., 2017). Though it also helps box-based detection, the gains are actually limited, e.g., as shown by Row 3 and 6 of Table 3 in (He et al., 2017), RoIAlign increases AP_{box} by 1.1pt over RoIPooling (36.2 to 37.3). As a comparison, replacing RoIPooling with RoIAlign significantly boosts AP_{box} by 3.7pt when trained from scratch. The insights behind these improvements are fundamentally different. For pixel-based detection, RoIAlign works by conducting bilinear interpolation to avoid inaccurate segmentation boundaries. For our work, the involvement of RoIAlign is to enable smooth gradient propagation, and help the model converge to a better situation.

4.2 From T-T-T-T to C-C-T-T

The convolution operations inherently have the inductive bias towards local processing, which is replaced in vision transformers by global processing performed by multi-head self-attention (Vaswani et al., 2017). Intuitively, it seems not so necessary to conduct long-range attention modeling in pixel-level or lower stages of backbones. Recent studies have

Table 1 From RoIPooling to RoIAlign. RoIAlign enables smooth gradient back-propagation and boosts detection mAP by 3.7pt in "Train from Scratch" setting.

	Train from Scratch				Pre-train & Fine-tune			
	mAP	mAP_S	mAP_M	mAP_L	mAP	mAP_S	mAP_M	mAP_L
RoIPooling	26.6	13.0	29.0	37.7	42.1	21.3	40.2	49.2
RoIAlign	30.3	15.5	33.0	41.6	42.5	21.6	40.6	49.7

When it moves to the "Pre-train & Fine-tune" case where the weights are properly initialized, the improvement of RoIAlign is not so significant

Table 2 From T-T-T-T to C-C-T-T (All experiments are trained from scratch in this table).

	Faster R-CNN				FCOS			
	mAP	#Params	FLOPs	Memory	mAP	#Params	FLOPs	Memory
T2-T2-T6-T2	30.3	**68.93M**	**246.30G**	**15.1G**	23.6	**35.73M**	**211.56G**	**14.2G**
C2-C2-T6-T2	26.6	44.08M	188.31G	10.6G	18.8	24.67G	187.16G	9.8G
C2-C2-T9-T3	37.9	**68.62M**	**245.68G**	**14.8G**	29.5	**35.51M**	**207.15G**	**13.9G**

Thanks to the removal of high-resolution self-attention operators in lower layers, we are feasible to enhance the model capacity by heuristically stacking more self-attention blocks to the latter two T blocks. As highlighted by bold values, the resources consumption of ours and the baseline T2-T2-T6-T2 (i.e., Swin-T (Liu et al., 2021)) are similar

Algorithm 1: Gradient calibration.

1 Draw a batch of samples from training set
2 Compute the norm of layer gradient $C_i = \mathbb{E}_j[C_{i,j}]$
3 Compute the average ratio $\bar{C} = (\prod_i C_i)^{\frac{1}{N}}$
4 Compute the calibration multiplier $r_k = (C_k/\bar{C})^{\alpha}$
5 Calibrate the weights of each layer as $W_k \leftarrow r_k W_k$

also revealed that large-scale pre-training essentially makes lower attention layers learn inductive bias and "act like convolutions" (Raghu et al., 2021). Therefore, a natural idea is to replace early self-attention layers with convolution, so as to directly introduce the inductive prior of convolution into the model and mitigate the dependence on large-scale pre-training.

Similar to ResNet (He et al., 2016), Swin Transformer also has four stages, each of which consists of multiple stacked transformer blocks. We dub such an architecture as T-T-T-T, where T stands for transformer. Further, the typical Swin-T model is then denoted as T2-T2-T6-T2 (Row 1 in Table 2), meaning that the Swin-T model consists of 4 blocks, each of which is stacked by 2, 2, 6 and 2 transformer layers, respectively. To introduce the inductive prior of convolution into the model, we replace each one of the first two T block with a stack of residual convolutional units (termed as C) (He et al., 2016) and obtain the C2-C2-T6-T2 architecture (Row 2 in Table 2). Though such replacement is efficient in resource (e.g., parameters, FLOPs and memories), the resulting detection mAPs of both Faster R-CNN and FCOS degrade, possibly due to the reduced model capacity.

Fortunately, convolution operation consumes much fewer resources than transformer block, when the resolution of feature map is the same. By replacing transformer block with convolution in the former 2 stages (whose feature maps are large!), we can allocate more resources to increase the layers of transformer block in the latter 2 stages (whose feature maps are much smaller). As shown in Row 3 of Table 2, the C2-C2-T9-T3 architecture significantly boosts the detection mAP of both Faster R-CNN and FCOS when trained from scratch, without consuming more resources than the vanilla T2-T2-T6-T2 architecture[1]. More variants of architectures such as C-C-C-C and C-T-T-T are ablated in Sect. 5.1.

4.3 Gradient Calibration

The heterogeneous C-C-T-T architecture introduces the hybrid of convolution and self-attention layers. We observe that they exhibit significant differences in terms of the norm of layer gradient (defined below). The norm of layer gradient of self-attention layers can be up to *10* times as that of convolution layers. Existing research has found that it is better to adjust all of the layers a little rather than to adjust just a few layers a large amount (Yosinski et al., 2014). Therefore, we propose to calibrate the gradients of our model, so as to achieve better convergence property.

Definition 1 (Norm of gradient.) Given a N-layer neural network, we define $C_{i,j}$ to be the expected norm of the gradient

[1] Strictly speaking, the C-C-T-T based detector cannot be called a vision transformer based detector. However, for the simplicity of presentation, we do not explicitly distinguish C-C-T-T and T-T-T-T architectures in concept, and still refer the process of training both of them as training vision transformer based detectors.

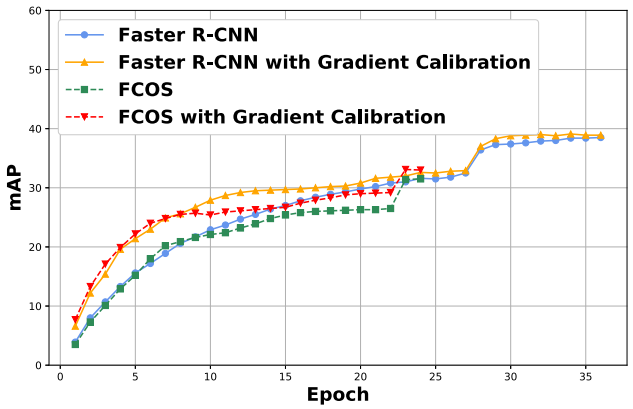


Fig. 5 Train Faster R-CNN (Ren et al., 2015) and FCOS (Tian et al., 2019) on COCO dataset, with and without gradient calibration. The proposed gradient calibration not only accelerates the convergence rate, but also improves the final mAP (Color figure online)

w.r.t. weights $W_i(j)$ of layer i:

$$C_{i,j} = \mathbb{E}_{z_0 \in D}\left[\left(z_{i-1}(j) y_i(j)\right)^2\right], \tag{1}$$

where D is the set of training data, z_{i-1} is the activation of layer $i-1$, and y_i is the backpropagated error of layer i.

Definition 2 (Norm of layer gradient.) Given norm of gradient, the norm of layer gradient is defined as:

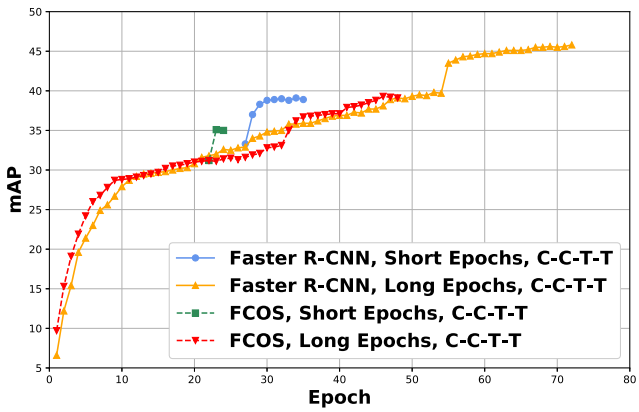
$$C_i = \mathbb{E}_j\left[C_{i,j}\right]. \tag{2}$$

The proposed gradient calibration works by adjusting the scale of weights in each layer in initialization, so that they are all equal to their geometric average. Specifically, we first compose a batch with randomly selected samples from the training set. Next, we forward and backward propagate this batch through our model to obtain the norm of layer gradient. Then, we compute the geometric average of all norms of layer gradient, and find out the scale correction multiplier of each layer. Finally, we multiply the weights with the scale correction multiplier so that they have the same norm of layer gradient. The entire process is summarized in Algorithm 1, where α in Line 4 is a hyper-parameter (0.25 in this work) against oscillatory behavior. Figure 5 illustrates the training curves of Faster R-CNN and FCOS with and without gradient calibration. The proposed gradient calibration not only accelerates the convergence rate, but also improves the final detection mAP.

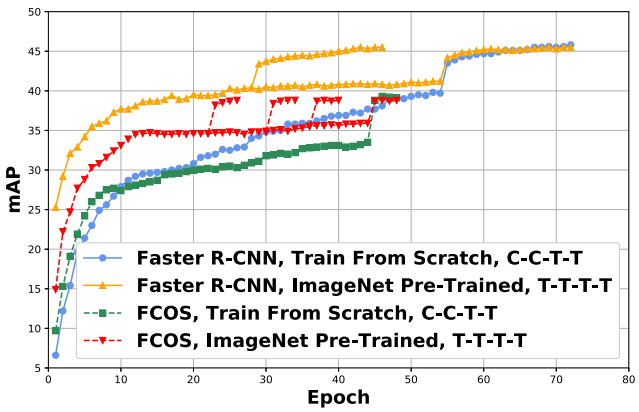
Here we also present another perspective to intuitively interpret the benefits of gradient calibration. Typically, Transformer models require a small learning rate to converge, for example, 0.0005 in BERT (Devlin et al., 2019), 0.001 in Swin Transformer (Liu et al., 2021). In contrast, the learning rate for CNNs is much larger, i.e., 0.1 for ResNet (He et al., 2016). Though the optimizers for vision transformer and CNNs are usually different (e.g., AdamW v.s. SGD), the significant gap in learning rate suggests that it might be sub-optimal to naively train a hybrid model of convolution and self-attention without any adjustment.

4.4 More Training Epochs

Though gradient calibration accelerates the convergence and improves final mAP, it is still unrealistic and unfair to expect models trained from random initialization like (He et al., 2015) to converge as well as those initialized from ImageNet pre-training. Typical ImageNet pre-training can learn not only semantic information, but also low-level features



(a) Short epochs *v.s.* long epochs.



(b) Trained from scratch *v.s.* pre-training.

Fig. 6 Experiments with various training epochs. **a**. All four detectors are equiped with C-C-T-T backbones and trained from scratch with short or long epochs. One can observe that detectors trained from scratch require more epochs than those with pre-trained weights to reach convergence. **b** All four detectors are trained with sufficient epochs. Under the long epochs training schedules, C-C-T-T architecture trained from scratch converges to a solution that is no worse than the pre-trained T-T-T-T counterpart (Color figure online)

Table 3 Different designs of backbone. With "pre-train and fine-tune", the full transformer network T-T-T-T demonstrates a better mAP at 45.5 than other four competitors; While in the case of training from scratch, the C-C-T-T architecture achieves the highest mAP at "45.8", which validates the rationale of our choice of C-C-T-T architecture

	Training setting	mAP	mAP_{50}	mAP_{75}	#params	FLOPs	FPS	Memory
T2-T2-T6-T2	Pre-train & Fine-tune	45.5	67.7	49.0	68.93M	246.30G	23.9	15.1G
	Train from Scratch	43.4	65.8	46.9				
C2-T2-T8-T2	Pre-train & Fine-tune	45.4	67.5	48.5	68.78M	245.92G	23.8	15.0G
	Train from Scratch	44.8	66.7	48.0				
C2-C2-T9-T3	Pre-train & Fine-tune	45.3	67.4	48.1	68.62M	245.68G	24.1	14.8G
	Train from Scratch	45.8	68.1	48.8				
C2-C2-C12-T4	Pre-train & Fine-tune	44.1	65.9	47.2	68.76M	246.17G	24.2	14.9G
	Train from Scratch	43.7	66.2	47.1				
C2-C2-C16-C4	Pre-train & Fine-tune	43.5	66.1	46.8	68.84M	245.88G	23.7	14.9G
	Train from Scratch	42.3	64.6	45.5				

(e.g., edges, textures) that do not need to be re-learned during fine-tuning.

Similar to the settings in Sect. 3.1, we train our detectors with the initial learning rate and monitor the validation set mAP at each epoch. When the mAP reaches saturation, we decay the learning rate and continue to train it until convergence. In consideration of the scale of the COCO and ImageNet dataset, the iterations of "more training epochs" setting are still much less than the "pre-train & fine-tune" pipeline (See Figure 2 of (He et al., 2019)).

The experimental results are shown in Fig. 6a. As expected, detectors trained from scratch require more epochs than those with pre-trained weights to reach convergence. Particularly, the final mAP of both Faster R-CNN and FCOS are 45.8 and 38.9, which is superior or similar to their ImageNet pre-trained counterpart, i.e., 42.5 and 38.8 as shown in Table 1 and Fig. 4, respectively.

Moreover, we train T-T-T-T models initialized by ImageNet pre-trained weights for long epochs, and explore different training schedules by varying the epochs at which the learning rate is reduced (where the mAP leaps). As illustrated in Fig. 6b, the C-C-T-T model trained from random initialization needs more iterations to converge, but the final mAP is no worse than that of the fine-tuning counterpart.

5 Experiments

We conduct experiments on the MS COCO dataset and measure detection performance by mean average precision (mAP).

5.1 Ablation Studies

All ablation studies are based on the Faster R-CNN detector (Ren et al., 2015). The performance achieved by different variants and backbones settings are reported in the following.

5.1.1 Different Design Choices of Architecture

Based on the findings that lower attention layers tend to learn inductive bias and "act like convolutions" (Raghu et al., 2021), we propose 4 variants with increasingly more Transformer stages, i.e., C-C-C-C, C-C-C-T, C-C-T-T and C-T-T-T, where C and T represent Convolution and Transformer respectively. For the purpose of conducting fair comparisons of the 4 designs, we will heuristically adjust the number of layers in each stage (listed in the first column of Table 3), to make each of them consume similar GPU memory to that of T2-T2-T6-T2 (roughly 16G in this ablation study).

To systematically study the design choices, we evaluate their performance in two different settings, i.e., "pre-train and fine-tune", "train from scratch". for "pre-train and fine-tune", we pre-train the model on ImageNet dataset, and fine-tune the weights on the COCO dataset for object detection, following the setting of Swin Transformer (Liu et al., 2021); For "train from scratch", we conducted the training following our proposed methods in Sect. 4.

The experimental results are shown in Table 3. On one hand, under the "Pre-train & Fine-tune" paradigm, the full transformer architecture T-T-T-T, which is exactly Swin-T (Liu et al., 2021), achieves the highest mAP at 45.5. Also, we can observe that the performance monotonically grows during the change from C-C-C-C to T-T-T-T, even if the total number of layers is decreasing. Such a phenomenon demonstrates the great modeling capacity of the self-attention operator. On the other hand, when it moves to "Train from Scratch" setting, C-C-T-T architecture shows

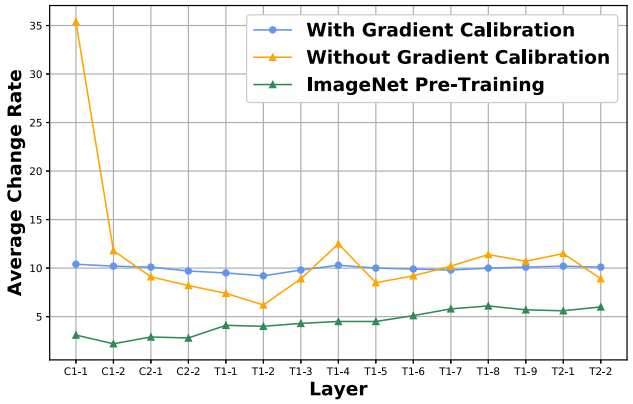


Fig. 7 Average change rate A flat curve is better, as all layers learn at the same rate. Random initialization without gradient calibration demonstrates large variance of average change rate between different layers, while initialization with ImageNet pre-training and our gradient calibration have relatively flatter curves (Color figure online)

the best detection performance at 45.8, which reveals the good trade-off between model capacity and inductive prior. Notably, under the same consumption of memory, the C-C-T-T architecture trained from scratch achieves even 0.3 point higher mAP than the T-T-T-T variant initialized from ImageNet pre-trained weights.

5.1.2 On the Effect of GC

To investigate the effect of the proposed gradient calibration algorithm, we monitor the norm of the layer gradient when training our C-C-T-T architecture. Specifically, we measure its relative change rate between before and after training.

The results are shown in Fig. 7. One can observe the following facts: (1) Random initialization (orange curve) exhibits large average change rate, while initialization with ImageNet pre-training and our proposed gradient calibration have flat curves, which indicates that all layers learn at a similar rate; (2) The curve of ImageNet pre-training has smaller average change rate than ours, especially in the lower layers, i.e., the convolutional layers.

We hypothesize the reason is that ImageNet pre-training provides a good initialization of lower layers, which are mainly responsible to extract low-level visual cues like corner, edge, etc. Therefore, these layers do not need to be significantly updated when switching from classification to detection.

5.2 Working with Variants of Swin Transformer

We study the generability of the proposed method to other Swin Transformer variants, namely, Swin-T, Swin-S, Swin-B and Swin-L. Similar to previous settings, we adjust the number of layers in the latter two stages, to make T-T-T-T and C-C-T-T architectures consume similar resources.

The experimental results are shown in Table 4. The #Channels denotes the channel number of the hidden layers in the first stage for T-T-T-T architecture, and the channel number of the residual unit in the first stage for C-C-T-T architecture. The proposed method, trained from scratch, consistently performs favorably against the vanilla Swin Transformer counterpart that is initialized with ImageNet pre-training, validating the efficacy of our work.

5.3 Generalizing to Other Vision Transformers

In this section, we apply our findings to other vision transformers, including PVT (Wang et al., 2021), BotNet (Srinivas et al., 2021) and MViT (Li et al., 2022). All of the three vision transformers have four stages, and each stage is composed by stacking multiple multi-head self-attention blocks. Following our proposed principles in Sect. 4, we modified these three vision transformers by the following steps: (1) Replace the multi-head self-attention blocks in the former two stages with residual convolution ops, (2) Increase the number of layers in the latter two stages so that they consume similar resources to their vanilla versions, (3) Initialize them with

Table 4 Different scales of backbones

	Training setting	mAP	mAP_S	mAP_M	mAP_L
T2-T2-T6-T2 (Swin-T)	Pre-train and fine-tune	45.5	30.0	49.0	58.7
C2-C2-T9-T3	Train from scratch	45.8	30.5	49.2	59.3
T2-T2-T18-T2, #Channel=96 (Swin-S)	Pre-train and fine-tune	48.2	32.9	52.2	62.2
C2-C2-T27-T3 (#Channel=96)	Train from scratch	48.6	33.4	52.9	62.8
T2-T2-T18-T2, #Channel=128 (Swin-B)	Pre-train and fine-tune	51.0	35.5	54.8	64.4
C2-C2-T27-T3 (#Channel=128)	Train from scratch	51.2	35.8	55.1	64.8
T2-T2-T18-T2, #Channel=192 (Swin-L)	Pre-train and fine-tune	52.9	37.0	56.7	66.5
C2-C2-T27-T3 (#Channel=192)	Train from scratch	53.0	37.2	56.9	66.8

The #Channels denotes the channel number of the hidden layers in the first stage for T-T-T-T architecture, and the channel number of the residual unit in the first stage for C-C-T-T architecture. Our proposed method works well for various instances of Swin Transformer (Liu et al., 2021)

Table 5 Generalizing to other vision transformers. The proposed method generalizes well with all of the three vision transformers when trained from scratch on COCO dataset

Method	Specifics	Training setting	mAP	mAP_{50}	mAP_{75}	#params	FLOPs	FPS
PVT	T3-T3-T6-T3 (PVT-Small)	Pre-train & Fine-tune	43.0	65.3	46.9	34M	212G	29.5
	C3-C3-T9-T3	Train from Scratch	42.8	65.0	46.5	33M	207G	28.1
BoTNet	T3-T4-T6-T3 (BoTNet-50)	Pre-train & Fine-tune	43.6	66.0	47.4	20M	145G	32.2
	C3-C4-T9-T3	Train from Scratch	43.5	65.7	47.2	20M	139G	32.9
MViT	T1-T2-T5-T2 (MViT-T)	Pre-train & Fine-tune	48.2	70.9	53.3	44M	279G	14.8
	C1-C2-T8-T3	Train from Scratch	48.1	70.9	53.1	42M	268G	16.2

Table 6 Working with sota detectors. The proposed method demonstrates promising results than both C-C-C-C and T-T-T-T

Detector	Backbone	Training setting	mAP	mAP_{50}	mAP_{75}	#params	FLOPs	FPS
Cascade Mask R-CNN	C3-C4-C6-C3 (ResNet-50)	Pre-train & Fine-tune	46.3	64.3	50.5	82M	739G	18.0
	T2-T2-T6-T2 (Swin-T)	Pre-train & Fine-tune	50.5	69.3	54.9	86M	745G	15.3
	C2-C2-T9-T3	Train from Scratch	51.0	69.8	55.3	88M	725G	19.2
ATSS	C3-C4-C6-C3 (ResNet-50)	Pre-train & Fine-tune	43.5	61.9	47.0	32M	205G	28.3
	T2-T2-T6-T2 (Swin-T)	Pre-train & Fine-tune	47.2	66.5	51.3	36M	215G	22.3
	C2-C2-T9-T3	Train from Scratch	47.5	66.7	51.6	35M	217G	26.1
RepPointsV2	C3-C4-C6-C3 (ResNet-50)	Pre-train & Fine-tune	46.5	64.6	50.3	42M	274G	13.6
	T2-T2-T6-T2 (Swin-T)	Pre-train & Fine-tune	50.0	68.5	54.2	45M	283G	12.0
	C2-C2-T9-T3	Train from Scratch	50.4	68.9	54.5	44M	279G	14.6
Sparse R-CNN	C3-C4-C6-C3 (ResNet-50)	Pre-train & Fine-tune	44.5	63.4	48.2	106M	166G	21.0
	T2-T2-T6-T2 (Swin-T)	Pre-train & Fine-tune	47.9	67.3	52.3	110M	172G	18.4
	C2-C2-T9-T3	Train from Scratch	48.2	67.4	52.6	108M	170G	22.3

the proposed gradient calibration method in Algorithm 1, (4) Train them from scratch with extended epochs.

The experimental results with Faster R-CNN as detectors are shown in Table 5. When trained from scratch, these modified vision transformers demonstrate competitive accuracy to their pre-trained counterparts, demonstrating the generalization effect of our proposed principles of modification.

5.4 Coupled with State-of-the-Art Detectors

We apply our findings to several state-of-the-art detectors and train them from scratch, to validate the generability of our methods. The detectors include Cascade Mask R-CNN (Cai & Vasconcelos, 2018), ATSS (Zhang et al., 2020), RepPointsV2 (Chen et al., 2020) and Sparse R-CNN (Sun et al., 2021), whose implementations are adopted from mmdetection (Chen et al., 2019).

Particularly, there are 3 types of backbones for each of the 4 detectors, i.e., (1) The C-C-C-C architecture, which is essentially ResNet-50 (He et al., 2016); (2) The T-T-T-T architecture, which is essentially Swin-T (Liu et al., 2021); (3) The proposed C-C-T-T architecture with gradient calibration. Note that the C-C-C-C and T-T-T-T models are pre-trained on ImageNet, while ours is randomly initialized. All combinations are trained on the COCO dataset with multi-scale learning (resizing the input such that the shorter side is between 480 and 800 while the longer side is at most 1333), AdamW (Loshchilov & Hutter, 2019) optimizer (initial learning rate of 0.0001, weight decay of 0.05, and batch size of 16) and sufficiently long training epochs. The results are shown in Table 6, the proposed C-C-T-T design with gradient calibration demonstrates competitive performance in all experiments.

We provide illustrative samples of our vision transformer based detector (Swin-T (Liu et al., 2021) + faster R-CNN (Ren et al., 2015)), trained from scratch. The detection results on the COCO dataset are shown in Fig. 8.

6 Conclusion

The domination of convolutional neural networks (CNNs) in vision tasks has recently been challenged by transformer models, which heavily depend on large-scale pre-training to achieve competitive accuracy. The dependence on pre-training not only hinders the freedom of architectural design in downstream tasks like object detection, but also causes learning bias and domain mismatch in the fine-tuning stages.



Fig. 8 Results obtained by our vision transformer based detector, trained from scratch on COCO dataset

In this work, we first show that naively applying the experiences from training CNNs based detectors to vision transformer based ones results in unsatisfactory performance. These experiments suggest that both architectural changes and more epochs play critical roles for this task; neither alone can train vision transformer-based detectors from scratch to achieve performance comparable to their pre-trained counterparts. Then, we demonstrate the feasibility of training vision transformer based detectors from scratch, and contribute a set of principles for realizing this goal. Particularly, the purpose of this work is not to propose a specific vision transformer based detector. Instead, we aim to uncover the insights of training vision transformer based detector from scratch, and expect those insights can help other researchers and practitioners, and inspire more interesting research in other fields, such as remote sensing, visual-linguistic pre-training, etc. By introducing a series of effective modifications such as C-C-T-T and gradient calibration, the proposed detectors demonstrate competitive mAP to their pre-trained variants, under the same long training epochs schedule. Extensive experiments demonstrate the merits and advantages of the proposed method.

Data Availability Statement The data that supports the findings of this study are available in MS COCO official website (https://cocodataset.org/#home) with public access.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *The European conference on computer vision (ECCV)*.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., & Lin, D. (2019). Mmdetection: Open MMLab detection toolbox and benchmark. arXiv:1906.07155.

Chen, Y., Zhang, Z., Cao, Y., Wang, L., Lin, S., & Hu, H. (2020). Reppoints v2: Verification meets regression for object detection. In *Advances in neural information processing systems (NIPS)*.

Cheng, B., Schwing, A.G., & Kirillov, A. (2021). Per-pixel classification is not all you need for semantic segmentation. In *Advances in neural information processing systems (NIPS)*.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *The IEEE international conference on computer vision (ICCV)*.

d'Ascoli, S., Touvron, H., Leavitt, M., Morcos, A., Biroli, G., & Sagun, L. (2021). Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning (ICML)*.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Girshick, R. (2015). Fast R-CNN. In *The IEEE international conference on computer vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Guo, J., Han, K., Wu. H., Tang, Y., Chen, X., Wang, Y., Xu, C. (2022). Cmt: Convolutional neural networks meet vision transformers. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Gupta, S., Hoffman, J., & Malik, J. (2016). Cross modal distillation for supervision transfer. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. In *Advances in neural information processing systems (NIPS)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

He, K., Gkioxari, G., Dollar, P.,& Girshick, R. (2017). Mask R-CNN. In *The IEEE international conference on computer vision (ICCV)*.

He, K., Girshick, R., & Dollar, P. (2019). Rethinking imagenet pre-training. In *The IEEE international conference on computer vision (ICCV)*.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K.Q. (2017). Densely connected convolutional networks. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*.

Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *The European conference on computer vision (ECCV)*.

Li, Y., Wu, C.Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., & Feichtenhofer, C. (2022). Mvitv2: Improved multiscale vision transformers for classification and detection. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2018). Detnet: A backbone network for object detection. In *The European conference on computer vision (ECCV)*.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *The European conference on computer vision (ECCV)*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., & Berg, A.C. (2016). SSD: Single shot multibox detector. In *The European conference on computer vision (ECCV)*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *The IEEE international conference on computer vision (ICCV)*.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *international conference on learning representations (ICLR)*.

Matan, O., Burges, C.J.C., LeCun, Y., & Denker, J. (1992). Multi-digit recognition using a space displacement neural network. In *Advances in neural information processing systems (NIPS)*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE TKDE, 22*, 1345–1359.

Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., & Ye, Q. (2021). Conformer: Local features coupling global representations for visual recognition. In *The IEEE international conference on computer vision (ICCV)*.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? In: Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW (Eds.) *Advances in neural information processing systems (NIPS)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NIPS)*.

Rowley, H., Baluja, S., & Kanade, T. (1996). Human face detection in visual scenes. In *Advances in neural information processing systems (NIPS)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*.

Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., & Xue, X. (2017). DSOD: Learning deeply supervised object detectors from scratch. In *The IEEE international conference on computer vision (ICCV)*.

Srinivas, A., Lin, T.Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., & Luo, P. (2021). Sparse R-CNN: end-to-end object detection with learnable proposals. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *Advances in neural information processing systems (NIPS)*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Tian, Z., Shen, C., Chen, H.,& He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *The IEEE international conference on computer vision (ICCV)*.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jegou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning (ICML)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Lu., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (NIPS)*.

Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile

backbone for dense prediction without convolutions. In *The IEEE international conference on computer vision (ICCV)*.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In *The IEEE international conference on computer vision (ICCV)*.

Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., & Girshick, R. (2021). Early convolutions help transformers see better. In *Advances in neural information processing systems (NIPS)*.

Xu, R., Luo, F., Zhang, Z., Tan, C., Chang, B., Huang, S., & Huang, F. (2021). Raise a child in large language model: Towards effective and generalizable fine-tuning. In *EMNLP*.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems (NIPS)*.

Zhang, S., Chi, C., Yao, Y., Lei, Z., & Li, S.Z. (2020). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.