POA: Pre-training Once for Models of All Sizes

Yingying Zhang, Xin Guo, Jiangwei Lao, Lei Yu, Lixiang Ru, Jian Wang, Guo Ye, Huimei He, Jingdong Chen, and Ming Yang

Ant Group.
qichu.zyy@antgroup.com

Abstract. Large-scale self-supervised pre-training has paved the way for one foundation model to handle many different vision tasks. Most pre-training methodologies train a single model of a certain size at one time. Nevertheless, various computation or storage constraints in real-world scenarios require substantial efforts to develop a series of models with different sizes to deploy. Thus, in this study, we propose a novel tri-branch self-supervised training framework, termed as POA (**P**re-training **O**nce for **A**ll), to tackle this aforementioned issue. Our approach introduces an innovative elastic student branch into a modern self-distillation paradigm. At each pre-training step, we randomly sample a sub-network from the original student to form the elastic student and train all branches in a self-distilling fashion. Once pre-trained, POA allows the extraction of pre-trained models of diverse sizes for downstream tasks. Remarkably, the elastic student facilitates the simultaneous pre-training of multiple models with different sizes, which also acts as an additional ensemble of models of various sizes to enhance representation learning. Extensive experiments, including k-nearest neighbors, linear probing evaluation and assessments on multiple downstream tasks demonstrate the effectiveness and advantages of our POA. It achieves state-of-the-art performance using ViT, Swin Transformer and ResNet backbones, producing around a hundred models with different sizes through a single pre-training session. The code is available at: `https://github.com/Qichuzyy/POA`.

Keywords: Self-supervised Learning · Pre-training Once for All

1 Introduction

Learning generalizable visual representations in a large model by self-supervised learning has delivered superior performance across a wide range of visual tasks [10, 13, 20, 46, 49, 53] in recent years. Nevertheless, when deployed to real-world applications, a large model has to be adapted to diverse resource limitations in terms of computation, storage, or power consumption, *etc*. For example, a well-engineered AI product typically comprises a suite of models tailored for varying scenarios, such as Gemini Nano, Pro and Ultra [37]. Given a large pre-trained model, common solutions to deploy it to multiple application scenarios with different resource constraints include additional weight pruning [28, 42, 51], knowledge distillation [22, 24], or even re-training a small network from scratch, which all require substantial development efforts. Consequently, this issue prompts a critical question: is it possible to pre-train once to produce multiple models with different sizes simultaneously, each delivering sufficiently good representations?

To address this challenge, we introduce a new paradigm of self-supervised learning, called POA (**P**re-training **O**nce for **A**ll). POA is built upon the prevalent teacher-student self-distillation framework [5, 33, 55], with an additional innovative elastic student branch. The elastic student branch embeds a series of sub-networks through parameter sharing, upon the observation that smaller-sized models are sub-networks of a larger-sized one for modern network structures [14, 21, 30]. Moreover, the parameters of this branch are shared with the original, or intact student. At each pre-training step, we randomly sample a subset of parameters from the intact student to form the corresponding elastic student. Both the original intact student and the elastic student are trained to emulate the output of the teacher network. The teacher itself is continually refined via an exponential moving average (EMA) of the student's parameters, including the sampled elastic student. The elastic student facilitates effective and efficient pre-training on different subsets of parameters, leading to the successful extraction of high-performance sub-networks from the pre-trained teacher for subsequent downstream scenarios. It also acts as a form of training regularization by enforcing the outputs to match between the teacher and various sub-networks, contributing to a stable training process. Additionally, by serving as an ensemble of different sub-networks across different pre-training steps, the elastic student improves the representation learning [44].

To the best of our knowledge, POA represents the first self-supervised learning method capable of training multiple-sized models concurrently, each obtaining high-quality representations for different resource constraints without further pre-training. Figure 1 displays the k-nearest neighbors (k-NN) evaluation results for 143 sub-networks extracted from a ViT-L model [14] pre-trained by POA. By choosing different elastic widths and depths, which will be explained in Sec. 3.1, the pre-trained teacher model can generate a sufficient number of candidate sub-networks for the selection of the suitable model tailored for downstream applications according to the computational resources available. Notably, each sub-network is well-trained and exhibits superior performance, thanks to our meticulous design on same-view distillation, as validated in Sec. 4.4. In particular, the ViT-S, ViT-B and ViT-

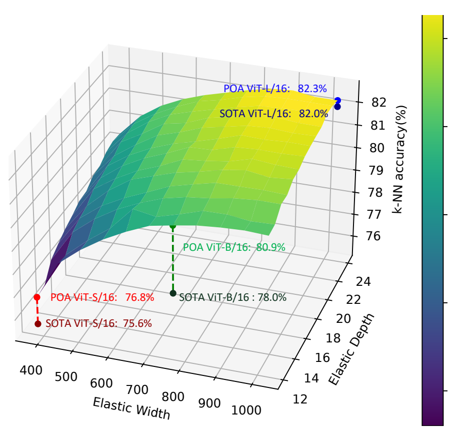

Fig. 1: The k-NN evaluation accuracy of 143 elastic ViTs derived from the ViT-L/16 teacher model pre-trained with POA.

L models set new benchmarks, achieving the state-of-the-art (SOTA) results compared with those pre-trained by existing methods [33, 34, 55].

To rigorously evaluate the efficacy of our approach, we conduct extensive experiments using three widely-used backbone architectures, *i.e.*, ViT [14], Swin Transformer [30], and ResNet [21]. Each backbone is pre-trained on ImageNet-1K dataset and assessed using k-NN and linear probing classification evaluation, as well as downstream dense prediction tasks such as object detection and semantic segmentation. Our method achieves SOTA accuracy across various model sizes with a single pre-training session. The technical contributions of this work are summarized as follows:

- To the best of our knowledge, POA is the first pre-training paradigm that integrates unsupervised representation learning and once-for-all model generation within a single pre-training session. It tackles the ***Pre-training-Once-for-All*** challenge which has been seldomly explored by the community but is of great practical significance for real-world deployment that usually requires a suite of models.
- We propose a novel and elegant component called ***Elastic Student***, featuring a range of elastic operators that enable POA to be compatible with popular backbone structures including ViT, Swin Transformer and ResNet. It provides the capability to generate models of diverse sizes. Furthermore, it serves as a model ensemble to smooth training process and improve learned representations.
- Through thorough assessments using k-NN, linear probing and downstream dense task evaluation, our approach exhibits superior performance over existing state-of-the-art pre-training methods across multiple metrics. Moreover, we compare our POA with Self-Supervised Distillation (SEED) [15], a knowledge distillation method designed especially for self-supervised learning, further validating POA's effectiveness.

2 Related Work

2.1 Self-supervised Learning

Self-supervised learning (SSL) frameworks commonly fall into two categories, generative and discriminative SSL. Most generative SSL approaches [7, 9, 18, 23, 25, 26, 38, 48, 52] focus on learning image representations directly in pixel space. On the other hand, discriminative SSL methods [4, 6, 19, 27, 36, 41, 43, 47] aim to learn representations by pulling those of different views of the same image closer while separating the representations of views from different images.

Contrastive learning (CL) with the InfoNCE loss [32] has emerged as a popular approach for discriminative SSL methods, attracting significant research interests in recent years. Although CL methods prevent the collapse of network representations through the use of negative samples, they still suffer from the dimensional collapse, where representations tend to collapse into a low-dimensional manifold. Grill *et al.* introduced a distillation-based asymmetric framework known as BYOL [17], which circumvents collapse without self-labeling or contrastive loss relying on negative samples. Following this work, distillation-based frameworks have become a prevailing trend in self-supervised learning. These frameworks [8, 16, 47] often merge with others to enhance overall performance. DINO [5] presented a simple self-distillation framework and has demonstrated impressive results in ViT pre-training. Subsequent works [33, 55] further improved the pre-training performance via masked token or the novel clustering design. Given the substantial benefits of distillation-based methods over other SSL techniques, we have developed our POA SSL framework upon these successful methodologies.

2.2 Dynamic Architecture.

Recently, Chen *et al*. proposed AutoFormer [45], which trained a supernet to support the effective search of optimal sub-network under some specific parameter number constraints. On the basis of [45], MaskTAS [50] introduced a self-supervised transformer architecture search method. Cai *et al*. [2] trained a network that accommodates various architectural configurations to reduce the training expense. Their methodologies enable the extraction of a specialized sub-network from the trained supernet. The design of the elastic student in our POA SSL is inspired by the weight-sharing strategy employed in these neural architecture search (NAS) methods. However, our implementation differs significantly due to the distinct purpose from NAS. Specifically, NAS aims to discover the optimal architecture within certain parameter constraints, which typically involves a huge number of sub-networks (more than 10^{16}) in the search space. Given the limitations on the number of training iterations and the network parameter capacity, it is challenging to ensure high performance across all sub-networks. After training the supernet, NAS requires a subsequent phase of searching and re-training to finalize the output model. In contrast, our POA SSL defines a sufficiently large yet compact set of sub-networks with elastic depths and widths for the purpose of pre-training models of various sizes via a single training session. Additional design on the same-view distillation guarantees that all elastic sub-networks within our framework are adequately and efficiently trained. As a result, our POA can readily extract a range of sub-networks from the teacher model without the need for extra pre-training, facilitating an easy selection of a suitable sub-network for different computational contexts. The design of elastic student is somewhat akin to the supervised learning method, Cosub [40]. The main difference is that Cosub simply skips blocks, making only the depth elastic.

3 POA Self-supervised Learning Framework

Our primary goal is to pre-train models of multiple sizes via a single self-supervised pre-training session. To this end, we propose a novel SSL framework named POA, inspired by the latest progress in self-distillation techniques [5, 33, 55]. The architecture of POA is illustrated in Figure 2, encompassing a

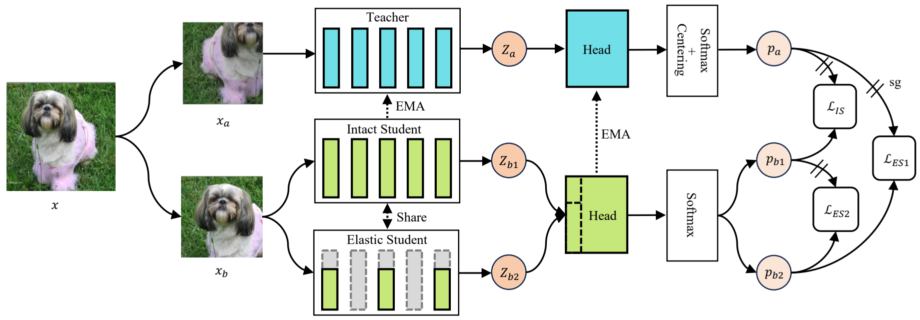

Fig. 2: Overview of the POA SSL: Given an image x, two augmented views x_a and x_b are generated. These views are input into three branches: a teacher, an intact student, and an elastic student, the latter being derived from the intact student. POA optimizes distillation losses in a twofold manner: the intact and the elastic students are distilled from the teacher using the cross-view data respectively, and additionally, the elastic student is distilled from the intact student using the same-view data.

teacher, an intact student, an elastic student and their corresponding heads. The teacher is updated with an EMA of the students. The elastic student is a derivative of the intact one, with both the backbones and heads' parameters shared. We leverage distillation in two aspects: both the intact and elastic students are distilled from the teacher using different views of the same image, while the elastic student additionally learns from the intact student using the identical views. The cross-view distillation works as a form of representation learning, as introduced in [5, 33, 55]. Notably, in addition to the regular EMA update with only the intact student as [5, 33, 55], the elastic student provides a randomly-sampled sub-network at each pre-training step to participate in the teacher's EMA refinement. This procedure actually simulates an ensemble of multiple sub-networks, which is also proven to be beneficial in the realm of supervised learning [44]. The same-view distillation is a standard knowledge distillation between the intact and elastic students, promoting the quality of the elastic one. Experiments in Sec. 4.4 validate our design in details.

3.1 Design of Elastic Student

The elastic student is a sub-network whose parameters are extracted from the intact student. In the context of a transformer backbone, the width refers to the dimensionality of the tokens, whereas for a convolutional backbone, the width indicates the number of channels. We denote the depth as the number of basic blocks in either a transformer or a convolutional network. Given the value of the width and depth, it yields a certain network structure. For simplicity, we focus on detailing the elastic design of the ViT in this section. For the similar elastic design of the Swin Transformer and ResNet, please refer to Appendix A.

A basic block of ViT mainly consists of a multi-head self-attention (MSA) module and a multi-layer perceptron (MLP) module. Layer Normalization (LN) [1] is applied before each module, with residual connections after each module.

As shown in the left part of Figure 3, an elastic block refers to a stack of elastic MSA, MLP, and LN after adjusting the width in the original basic block in ViT. In our approach, the elastic student branch is constructed by assembling a specific number of these elastic blocks at each training iteration.

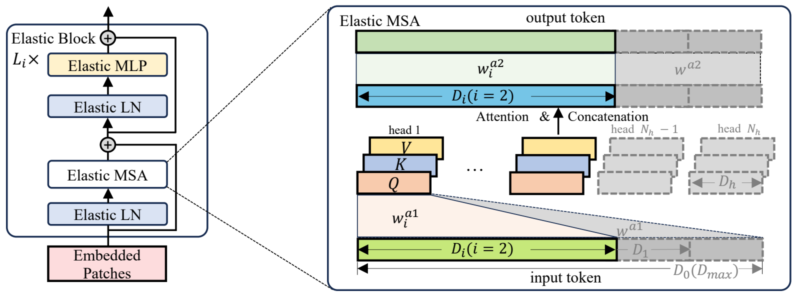

Fig. 3: Illustration of the elastic MSA in an elastic ViT block. To be concise, we simply exclude the projection layers that correspond to K and V in each head.

Elastic MSA An original, or intact MSA module consists of three major components, *i.e.*, input projection layers, an operator contains attention & concatenation, and an output projection layer. We define a projection layer as (w^*, b^*), where w^* is the linear transformation weight, b^* denotes the corresponding bias, and $*$ indicates the name of the layer. As shown in the right part of Figure 3, given a token dimension of $D_{max} = N_h \cdot D_h$, where N_h is the number of attention heads and D_h is the head dimension, an input sequence $z \in \mathbb{R}^{T\times D_{max}}$ with length T is initially projected to form the queries $Q \in \mathbb{R}^{T\times D_h}$, keys $K \in \mathbb{R}^{T\times D_h}$, and values $V \in \mathbb{R}^{T\times D_h}$ by the intact MSA. To generate elastic MSA, we define M + 1 elastic widths, including D_{max}, spaced at intervals of D_h as follows:

$$D_i = (N_h - i)\cdot D_h, \quad \forall i \in \{0, 1, ..., M\}, \quad M < N_h. \tag{1}$$

For each elastic width D_i, the weight $w_i^{a1} \in \mathbb{R}^{D_h\times D_i}$ and bias $b_i^{a1} \in \mathbb{R}^{D_h}$ that generate Q, K, and V of each head are extracted from the corresponding input projection layer (w^{a1}, b^{a1}) in the intact MSA, as $w_i^{a1} = w^{a1}[:, : D_i]\cdot\alpha_i$ and $b_i^{a1} = b^{a1}$. Here, α_i represents the scaling factor to cope with the reduction of the input dimension, calculated as $\alpha_i = D_{max}/D_i$. As the reduction of width, the number of attention heads in the elastic MAS is naturally reduced to $N_h - i$. Similarly, for the output projection layer (w^{a2}, b^{a2}), the weight $w_i^{a2} \in \mathbb{R}^{D_i\times D_i}$ and bias $b_i^{a2} \in \mathbb{R}^{D_i}$ are extracted as:

$$w_i^{a2} = w^{a2}[: D_i, : D_i]\cdot\alpha_i \qquad b_i^{a2} = b^{a2}[: D_i]. \tag{2}$$

Elastic MLP An original, or intact MLP module in the ViT block contains two projection layers. The first layer (w^{m1}, b^{m1}) expands the dimension of embedding by a factor of s, which is generally set to 4 in the ViT structure. The second layer (w^{m2}, b^{m2}) then projects it back to the original dimension. The parameters for both layers of the elastic MLP are extracted in a manner analogous to that described in Equation 2 as follows:

$$\begin{aligned} w_i^{m1} &= w^{m1}[: D_i \cdot s, : D_i] \cdot \alpha_i \qquad b_i^{m1} = b^{m1}[: D_i \cdot s] \\ w_i^{m2} &= w^{m2}[: D_i, : D_i \cdot s] \cdot \alpha_i \qquad b_i^{m2} = b^{m2}[: D_i]. \end{aligned} \tag{3}$$

Elastic LN For elastic LN, we directly use the first D_i elements of the parameter inside the original LN, akin to the bias extraction in Equation 2.

To create a sub-network with L_i elastic blocks from the intact ViT comprising L_{max} blocks, we introduce a set of $N+1$ elastic depths, defined as $L_i = L_{max} - i, \ \forall i \in \{0, 1, ..., N\}, \ N < L_{max}$. For a specific depth L_i, we select the corresponding blocks based on their block IDs at equal intervals. Each block ID $BID_j^{L_i}$ that is activated for depth L_i can be formulated as:

$$BID_j^{L_i} = \left\lfloor \frac{(L_{max} - 1) \cdot j}{L_i - 1} \right\rfloor, \quad \forall j \in \{0, 1, ..., L_i - 1\}. \tag{4}$$

Consequently, by combining elastic widths and depths, we can generate a total of $(N+1) \cdot (M+1)$ distinct sub-networks. For instance, by setting the elastic width to 384 and the elastic depth to 12, we can directly extract a ViT-S from the intact network such as ViT-L. During each iteration of the pre-training, we randomly select one of these sub-networks to serve as the elastic student branch.

3.2 Distillation between Views

POA performs distillation across its three branches accordingly. Given a pair of globally augmented views of an input image x, denoted as x_a and x_b, the teacher encoder E_T extracts the feature $Z_a = E_T(x_a)$ using x_a as input. Simultaneously, x_b is fed into both the intact student encoder E_{IS} and the elastic student encoder E_{ES}, producing the features $Z_{b1} = E_{IS}(x_b)$ and $Z_{b2} = E_{ES}(x_b)$, respectively. The feature output from the teacher encoder, Z_a, is then processed by the teacher head H_T, followed by centering with the Sinkhorn-Knopp (SK) [11] algorithm and normalization using a temperature-scaled softmax to generate the probability p_a as follows:

$$l_a = SK(H_T(Z_a)), \ l_a \in \mathbb{R}^P \qquad p_a^i = \frac{\exp(l_a^i/\tau)}{\sum_{k=0}^{P-1} \exp(l_a^k/\tau)}, \ \forall i \in \{0, ..., P-1\}, \tag{5}$$

where P is the number of prototypes and $\tau > 0$ is a temperature parameter. In a similar fashion, we compute the probabilities p_{b1}^i and p_{b2}^i for the intact and elastic student encoders, respectively, by processing the outputs through the student heads H_{IS} and H_{ES}. These outputs are then passed through a temperature-scaled softmax function using a temperature parameter τ' tailored for the student. It should be noted that H_{IS} and H_{ES} share the identical parameters, except that the first projection layer of H_{ES} is adapted similarly as in Equation 2 to conform to align with the corresponding dimensionality. For simplicity, we omit the explicit expressions for p_{b1}^i and p_{b2}^i, as they follow a similar calculation to Equation 5. For the intact student branch, we perform distillation from the teacher using the cross-view data as follows:

$$\mathcal{L}_{IS}^{g} = -p_a \log(p_{b1}). \tag{6}$$

The elastic student branch plays a pivotal role in our POA framework. To ensure adequate training of this branch, we employ a dual distillation from the teacher and intact student branch. The first distillation involves the teacher model, which utilizes the cross-view data to guide the learning of representations. The second is a distillation process with the intact student model, which uses the same-view data. This same-view distillation is responsible for transferring the representations learned by the intact student to the elastic student branch. The loss functions for this dual distillation process are formulated as follows:

$$\mathcal{L}_{ES1}^{g} = -p_a \log(p_{b2}), \quad \mathcal{L}_{ES2}^{g} = -p_{b1} \log(p_{b2}). \tag{7}$$

Note that in both loss functions, we sum over all prototypes to compute the cross-entropy loss between the respective probability distributions.

3.3 Overall Loss of POA

Following the SSL methods such as [5,33,55], we employ a multi-crop strategy [4] to create various distorted views from a single image. Apart from the two global views previously mentioned, we also generate v local views with lower resolution $x_{l_1}, x_{l_2}, ..., x_{l_v}$. These local views are processed by both students to promote *local-to-global* correspondence. The local distillation losses for the intact and elastic students are computed as follows:

$$\mathcal{L}_{IS}^{l} = -\frac{1}{v}\sum_{i=1}^{v} p_a \log(p_{l_i 1}), \tag{8}$$

$$\mathcal{L}_{ES1}^{l} = -\frac{1}{v}\sum_{i=1}^{v} p_a \log(p_{l_i 2}), \quad \mathcal{L}_{ES2}^{l} = -\frac{1}{v}\sum_{i=1}^{v} p_{l_i 1} \log(p_{l_i 2}), \tag{9}$$

where $p_{l_{i1}}$ and $p_{l_{i2}}$ are the probability produced by the intact and elastic student branches for the local view l_i, respectively. The total distillation loss of the intact and elastic student is calculated by summing them with the factor λ:

$$\begin{aligned} \mathcal{L}_{\mathcal{S}} &= \lambda(\mathcal{L}_{IS}^{g} + \mathcal{L}_{IS}^{l}) + (1-\lambda)((\mathcal{L}_{ES1}^{g} + \mathcal{L}_{ES1}^{l}) + (\mathcal{L}_{ES2}^{g} + \mathcal{L}_{ES2}^{l})) \\ &= \lambda\mathcal{L}_{IS} + (1-\lambda)(\mathcal{L}_{ES1} + \mathcal{L}_{ES2}). \end{aligned} \tag{10}$$

To ensure sufficient training of each sub-network from the elastic student, we introduce multiple projection heads (MPH) positioned after the backbone network. Each projection head has exactly the same structure, except for a different number of prototypes. For each head, the distillation loss $\mathcal{L}_{\mathcal{S}i}$ for both the intact and elastic student is calculated with Equation. 10. Finally, the overall loss function in the POA framework with H projection heads is formulated as: $\mathcal{L} = \frac{1}{H}\sum_{i=1}^{H} \mathcal{L}_{\mathcal{S}i}$.

4 Experiments

4.1 Implementation Details

Backbones. We have trained our POA using ViT, Swin Transformer and ResNet backbones, respectively. For the ViT, we configure the patch size to 16 and the dimension of each head in the MSA to 64. This aligns with the configurations typically used in existing SSL methods [5,34,35,55]. For the smallest and largest elastic networks of ViT, we choose the ViT-S and ViT-L, respectively. This leads to 11 elastic widths and 13 elastic depths, yielding a total of 143 ViT sub-networks. In the case of the Swin Transformer, we set the smallest and largest elastic networks as Swin-T and Swin-B, respectively. This configuration yields a total of 39 Swin sub-networks by multiplying the number of widths and depths as 3×13. For the ResNet, we establish the smallest and largest elastic network configurations as ResNet-50 and ResNet-152. Consequently, this setup accounts for a total of 465 ResNet sub-networks, which is the product of 3×155.

Pre-Training Setup. We pre-train all models on the ImageNet-1K dataset [12] without the labels. The process employs the AdamW optimizer [31] with a batch size of 1600, which is distributed across 32 A100 GPUs when employing a ViT backbone. We adopt a learning rate schedule that begins with a linear warm-up during the first 10 epochs, reaching a base value that is scaled proportionally to the total batch size: $\text{lr} = 0.004 \times \sqrt{\text{batch size}/1024}$, in line with [33]. Following this warm-up period, the learning rate is decayed with a cosine schedule. Similarly, the weight decay follows a cosine schedule, starting at 0.04 and increasing to 0.4. The student network's temperature τ' is fixed at 0.1, whereas the teacher temperature τ changes with a linear warm-up from 0.04 to 0.07 over the first 30 epochs. For further details, please refer to Appendix C.

4.2 Experiments on ImageNet-1K

After unsupervised pre-training, we assess the model's performance using two widely-recognized evaluation protocols in SSL domain on ImageNet-1K dataset, *i.e.*, k-NN and linear probing. To ensure a fair comparison between SSL methods that employ different numbers of crop views for data augmentation, Zhou *et al.* [55] introduced the effective training epoch as a measure to quantify the extent of a method's pre-training. We report the effective training epochs of the SSL methods for comparison. For additional evaluations including fine-tuning, semi-supervised and unsupervised learning, please refer to Appendix D.

k-NN and Linear Probing. To assess the quality of pre-trained features, we employ a k-NN classifier and a linear classifier on the frozen representations. For both the k-NN and linear probing (LP) evaluation, we follow the evaluation protocols established in [5, 33, 55]. The performance of our method when being trained using ResNet, Swin Transformer and ViT backbones is reported in Table 1. Our POA SSL achieves the SOTA k-NN accuracy of **82.3%** and LP accuracy

Table 1: Comparison results of k-NN and linear probing classification accuracy (%) on the ImageNet-1K dataset. "Param." refers to the number of parameters. "Epo." represents the number of effective training epochs following [55]. "/16" denotes patch size of 16. "/W7" means the window size of 7. "$*$" indicates our implementation based on official codebase. "†" denotes reproduced results using the released code.

Method	Epo.	k-NN	LP
ResNet-50(Param. 23M)			
VICReg	2000	-	73.2
SwAV	2400	65.7	75.3
DINO	3200	67.5	75.3
UniGrad	2400	-	75.5
SCFS	3200	68.5	75.7
ReLICv2	4000	-	**77.1**
POA	0	**73.4**	76.9
ResNet-101(Param. 41M)			
ReLICv2	4000	-	78.7
POA	0	**75.7**	**79.1**
ResNet-152(Param. 56M)			
ReLICv2	4000	-	79.3
POA	2400	**76.4**	**79.9**

(a) ResNet backbone.

Method	Epo.	k-NN	LP
Swin-T/W7(Param. 28M)			
SMoG	1200	-	77.7
iBOT	1200	75.3	78.6
DINOv2*	1200	75.4	78.0
EsViT	1200	75.7	78.1
POA	0	**77.5**	**78.9**
Swin-S/W7(Param. 49M)			
DINOv2*	1200	76.1	79.8
EsViT	1200	77.7	79.5
POA	0	**79.3**	**81.3**
Swin-B/W7(Param. 87M)			
DINOv2*	1200	76.9	80.9
EsViT	1200	78.9	80.4
POA	1200	**79.6**	**82.0**

(b) Swin backbone.

Method	Epo.	k-NN	LP
ViT-S/16(Param. 21M)			
DINO	3200	74.5	77.0
iBOT	3200	75.2	**77.9**
ENT	3200	75.3	77.7
POA	0	**76.8**	77.6
ViT-B/16(Param. 85M)			
DINO	1600	76.1	78.2
ENT	1600	77.1	79.1
iBOT	1600	77.1	79.5
POA	0	**80.9**	**81.7**
ViT-L/16(Param. 307M)			
iBOT	1200	78.0	81.0
DINOv2†	1200	82.0	83.3
POA	1200	**82.3**	**83.6**

(c) ViT backbone.

of **83.6%** when using the ViT-L/16 backbone. By employing the sub-network extraction approach outlined in Section 3.1, we derive the sub-networks ViT-S/16 and ViT-B/16 from the teacher ViT-L/16 without any additional pre-training. Thus, the number of effective training epochs of them is 0. Notably, the extracted ViT-S/16 achieves the SOTA k-NN accuracy of **76.8%** and LP accuracy on par with the previous SOTA of 77.9% reported by iBOT [55]. Our derived ViT-B/16 model sets new benchmarks for both k-NN and LP, with accuracy of **80.9%** and **81.7%**, respectively. For the models using Swin Transformer and ResNet backbones, POA also reaches SOTA performance in k-NN and LP accuracy. The only exception is the LP accuracy of ResNet-50, which is competitive with that of ReLICv2 [39], despite the latter being trained over a much longer period (4000 epochs). The superior performance achieved across a range of backbone architectures confirms the effectiveness and versatility of our method. For additional detailed comparison with **27** existing methods, please see Appendix D.1.

4.3 Evaluation on Downstream Visual Tasks

Object Detection and Instance Segmentation on COCO Dataset. For a fair comparison, we utilize the Cascade Mask R-CNN framework [3,20], which generates both bounding boxes and instance masks, in line with previous approach [55], on the COCO dataset [29]. We benchmark our results against existing SSL methods that generate pre-trained ViT-S/16 and ViT-B/16 backbones. As shown in Table 2, POA boosts the bounding box average precision (AP^b) for

ViT-S/16 from 49.4 to **50.6** and the mask average precision (AP^m) from 42.6 to **43.8**. When applied to ViT-B/16, POA attains an AP^b of **52.4** and an AP^m of **45.4**, which represents a remarkable step forward over previous SOTA.

Semantic Segmentation on ADE20K Dataset. For the semantic segmentation task, we primarily focus on two settings on the ADE20K dataset [54], following [55]. First, akin to the linear evaluation protocol in classification, we evaluate the quality of representations by keeping the patch features fixed and only fine-tuning one linear layer. This approach offers a clear comparison of representation quality. Second, we employ the UPerNet [46] as the task head and fine-tune the entire network. As depicted in Table 2, our POA significantly outperforms the supervised baseline using the ViT-S/16 backbone, achieving a substantial increase of **2.2** in mean Intersection over Union (mIoU), and surpassing iBOT by **1.3** mIoU. With the ViT-B/16 backbone, POA exceeds the previously best-performing method, iBOT, by **0.4** mIoU when utilizing UPerNet. Furthermore, under the assessment using solely a linear head, POA obtains an impressive improvement of **2.0** mIoU over iBOT as the performance is largely determiend by the quality of the pre-trained representation.

Table 2: Evaluation results on downstream detection and segmentation tasks. Seg.† indicates the use of a linear head for semantic segmentation.

Method	Arch.	Det. AP^b	ISeg. AP^m	Seg. mIoU	Arch.	Det. AP^b	ISeg. AP^m	Seg.† mIoU	Seg. mIoU
Sup.	ViT-S/16	46.2	40.1	44.5	ViT-B/16	49.8	43.2	35.4	46.6
BEiT	ViT-S/16	-	-	-	ViT-B/16	50.1	43.5	27.4	45.8
DINO	ViT-S/16	-	-	-	ViT-B/16	50.1	43.4	34.5	46.8
iBOT	ViT-S/16	49.4	42.6	45.4	ViT-B/16	51.2	44.2	38.3	50.0
POA	ViT-S/16	**50.6**	**43.8**	**46.7**	ViT-B/16	**52.4**	**45.4**	**40.3**	**50.4**

4.4 Ablations and Discussions

In this section, we conduct an empirical analysis of POA using ViT as backbone. Our investigation includes the impact of the loss functions $\mathcal{L}_{ES1}$ and $\mathcal{L}_{ES2}$, in addition with the effectiveness of multiple projection heads. Moreover, we compare our POA with knowledge distillation techniques for self-supervised learning to demonstrate the advantages of combining once-for-all model generation with self-distillation in a unified framework. For more ablation studies, we direct readers to the Appendix D.8. In addition, we further contrast our POA with three variants tailored for elastic pre-training to showcase POA's superiority. Finally, we discuss how the elastic student facilitates the pre-training.

Importance of Each Component. We evaluate the contributions of the components on a ViT backbone. Table 3 shows the performance of different component combinations. First, we note that employing multiple projection heads (MPH) enhances the learned representations for each elastic sub-network, particularly for smaller ones. The design consideration behind MPH is that for each

Table 3: Contributions of each component in POA framework. We conduct assessments with the k-NN and linear probing (LP) evaluations. MPH denotes the multiple projection heads with different numbers of prototypes.

MPH	$\mathcal{L}_{ES1}$	$\mathcal{L}_{ES2}$	k-NN ViT-S/16	k-NN ViT-B/16	k-NN ViT-L/16	LP ViT-S/16	LP ViT-B/16	LP ViT-L/16
✓	✓	✓	**76.8**	**80.9**	**82.3**	**77.6**	**81.7**	**83.6**
	✓	✓	76.2	80.7	82.2	77.3	81.6	83.4
		✓	75.1	80.2	82.2	75.8	81.0	83.4
	✓		72.8	79.1	82.1	75.3	80.8	83.3

pre-training iteration, the sub-network is chosen randomly, leading to a relatively insufficient optimization. MPH introduces different sets of prototypes, which act as multiple semantic spaces for representation learning, enabling the teacher to distill various aspects of learned knowledge into the sub-network. Furthermore, we ascertain that the same-view distillation loss $\mathcal{L}_{ES2}$ is crucial for the representation quality of elastic sub-networks. Omitting $\mathcal{L}_{ES2}$ causes a significant drop in k-NN accuracy, by 3.4% for ViT-S/16 and 1.6% for ViT-B/16. In addition, $\mathcal{L}_{ES2}$ is more important than cross-view distillation $\mathcal{L}_{ES1}$ in terms of sub-networks' representation quality. The underlying reason is that the cross-view distillation is for the unsupervised representation learning, while the same-view distillation improves the sub-networks by distilling previously learned representations from the intact student. Table 4 confirms that distillation from already-good representations is more effective than representation learning, especially for smaller networks. It also explains why employing three different views in our POA is unnecessary.

Table 4: Comparison with knowledge distillation for self-supervised learning. The teacher name "ViT-L/16-600" denotes a teacher model (ViT-L/16) that has been pre-trained with 600 effective epochs. The student name "ViT-S/16-600" refers to a student model (ViT-S/16) that has undergone distillation from the pre-trained teacher with 600 effective epochs. The number of the left side of "→" indicates the performance of the teacher model when pre-trained individually. The number of the right side denotes the performance achieved after distillation using the SEED [15].

Method	Teacher	Student	Total Epochs	k-NN	LP
DINOv2	ViT-S/16		1200	<u>72.2</u>	<u>73.1</u>
DINOv2+SEED	ViT-L/16-600	ViT-S/16-600	1200	81.3 → <u>74.0</u>	82.4 → <u>75.2</u>
DINOv2+SEED	ViT-L/16-1200	ViT-S/16-1200	2400	82.0 → <u>75.5</u>	83.3 → <u>76.2</u>
POA	ViT-S/16		1200	<u>**76.8**</u>	<u>**77.6**</u>
DINOv2	ViT-B/16		1200	<u>77.4</u>	<u>78.5</u>
DINOv2+SEED	ViT-L/16-600	ViT-B/16-600	1200	81.3 → <u>78.8</u>	82.4 → <u>80.0</u>
DINOv2+SEED	ViT-L/16-1200	ViT-B/16-1200	2400	82.0 → <u>79.7</u>	83.3 → <u>80.9</u>
POA	ViT-B/16		1200	<u>**80.9**</u>	<u>**81.7**</u>

Comparison with Knowledge Distillation. Knowledge distillation (KD) is a proven strategy for enhancing small networks by leveraging the knowledge of a well-trained, larger network. We compare our method with a self-supervised KD method SEED [15] that employs a pre-trained network from the previous SOTA DINOv2 as the teacher for distilling knowledge into smaller networks. We observe that SEED significantly boosts the performance of ViT-S/16 and ViT-B/16 when a pre-trained ViT-L/16 serves as the teacher. Specifically, with the same number of training epochs, SEED yields k-NN accuracy improvements of 1.8% and 1.4% for ViT-S/16 and ViT-B/16 respectively, compared with a learn-from-scratch setup. Our POA further outperforms SEED, delivering superior k-NN accuracy gains of **2.8%** (76.8% vs. 74.0%) and **2.1%** (80.9% vs. 78.8%) on ViT-S/16 and ViT-B/16. Moreover, the ViT-S/16 and ViT-B/16 derived directly from the pre-trained teacher from our POA, without additional pre-training, perform better than those enhanced by SEED, despite the latter undergoing two times of training epochs. These results demonstrate the effectiveness of our unified design, which integrates pre-training with once-for-all model generation. Additional comparison on training cost can be found in Appendix D.11.

Alternative Designs to Elastic Pre-training. Upon the proposed POA, we investigate three alternatives for concurrently pre-training multi-sized models within a single pre-training session. As illustrated in Figure 4, the first variant, POA-V1, eliminates the intact student and modifies the teacher to be elastic which aligns with the architecture of the elastic student. The second variant, POA-V2, only removes the intact student. However, in this model, the intact teacher is directly updated via an EMA of the elastic students. These two variants bear resemblance to current teacher-student self-distillation paradigm, with the primary differences in the structure of the teacher or student network. For the POA-V3, we introduce an additional elastic teacher branch that mirrors the architecture of the elastic student, enabling further cross-view distillation. For more details of the three variants, please refer to the Appendix D.9.

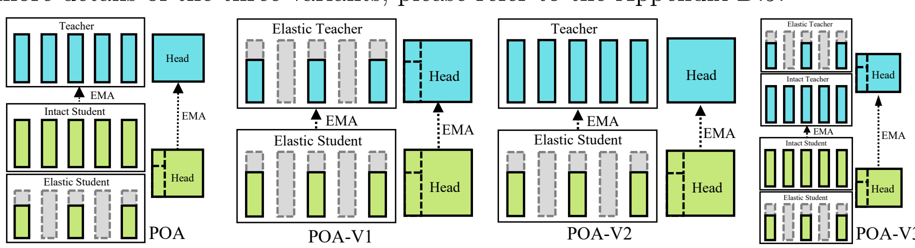

Fig. 4: Illustration of different variants of POA.

We compare the k-NN and linear probing accuracy of each variant in Table. 5. The results indicate that the performance drops dramatically for the variants with two branches. This decline can be attributed to two reasons: 1) in each training iteration, only a subset of the student's parameters are updated, leading to relatively insufficient overall pre-training; and 2) each elastic network undergoes training exclusively with the cross-view distillation, missing out the standard distillation guidance from a larger network, which our ablation study suggests is crucial for the efficacy of elastic networks. Although the POA-V3, featuring an additional elastic teacher branch, performs slightly better than our POA, it does so at the expense of increased model complexity and computational costs, making it a less appealing scheme than POA.

Table 5: k-NN and linear probing (LP) comparison of different variants of POA.

Method	k-NN			LP		
	ViT-S/16	ViT-B/16	ViT-L/16	ViT-S/16	ViT-B/16	ViT-L/16
POA	76.8	80.9	**82.3**	77.6	81.7	**83.6**
POA-V1	73.7	78.5	79.2	74.5	79.4	80.9
POA-V2	74.3	79.1	80.9	75.1	80.0	82.3
POA-V3	**76.9**	**81.0**	**82.3**	**77.8**	**81.8**	**83.6**

How Does Elastic Student Facilitate Pre-training? The elastic student branch enables the derivation of diverse-sized models directly from a pre-trained teacher, while simultaneously enhancing the learned representation. The elastic branch plays a dual role. First, it acts as a training regularization to stabilize the training progress. In our preliminary experiments, we observe that pre-training loss of a ResNet backbone easily diverges to NaN without the elastic branch. Conversely, the inclusion of the elastic student yields a highly stable pre-training. Furthermore, the elastic student provides hundreds of sub-network candidates that will be assembled to the teacher during pre-training. Unlike existing self-distillation methods, the teacher in the POA SSL integrates a series of sub-networks through an EMA update. Wortsman *et al.* [44] have shown that averaging the weights of multiple models typically improves accuracy and robustness. Similarly, the integration of different sub-networks for the teacher is effective in improving the representation quality. At the same time, an improved teacher promotes the learning of students in return, fostering a positive feedback loop for pre-training. For the visualization of representations, please refer to Appendix E.2.

5 Conclusion

In this study, we tackle the challenge of efficiently and effectively pre-training models of various sizes within a single self-supervised learning session, facilitating model deployment given different resource constraints. We propose a novel self-supervised learning paradigm, termed POA, which features an integrated design combining self-distillation and once-for-all model generation. It allows for the simultaneous pre-training of models of multiple sizes through an elastic branch design. POA enables the direct generation of varied-sized models from a pre-trained teacher, which are ready for downstream tasks without additional pre-training. This advantage significantly improves the deployment flexibility and facilitates our pre-trained model to achieve SOTA results across various vision tasks. Looking forward, we plan to extend POA to Multimodal Large Language Models, tapping into its vast potential for real-world AI product deployment.

References

1. Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. ArXiv **abs/1607.06450** (2016)
2. Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once for all: Train one network and specialize it for efficient deployment. In: International Conference on Learning Representations (2020)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**, 1483–1498 (2019)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 9912–9924. Curran Associates, Inc. (2020)
5. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9630–9640 (2021)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. International Conference on Machine Learning (2021)
7. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12146–12155 (2018)
8. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
9. Chen, Y., Liu, Y., Jiang, D., Zhang, X., Dai, W., Xiong, H., Tian, Q.: Sdae: Self-distillated masked autoencoder. In: European Conference on Computer Vision (2022)
10. Cheng, X., Chen, J., Wang, R.: Modified dual attention triplet-supervised hashing network for image retrieval. Signal, Image and Video Processing **18**, 1939–1948 (2023)
11. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Neural Information Processing Systems (2013)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR 2009 (2009)
13. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 295–307 (2014)
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR) (2021)
15. Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., Liu, Z.: Seed: Self-supervised distillation for visual representation. International Conference on Learning Representations (2021)
16. Gao, Y., Zhuang, J.X., Lin, S., Cheng, H., Sun, X., Li, K., Shen, C.: Disco: Remedying self-supervised learning on lightweight models with distilled contrastive learning. In: European Conference on Computer Vision (ECCV) (2022)
17. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own

latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
18. He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 15979–15988 (2021)
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9726–9735 (2019)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask r-cnn. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 2980–2988 (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2015)
22. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. ArXiv **abs/1503.02531** (2015)
23. Hou, L., Shen, H., Cao, Q., Cheng, X.: Self-supervised gans with label augmentation. ArXiv **abs/2106.08601** (2021)
24. Hu, C., Li, X., Liu, D., Wu, H., Chen, X., Wang, J., Liu, X.: Teacher-student architecture for knowledge distillation: A survey. ArXiv **abs/2308.04268** (2023)
25. Huang, Z., Jin, X., Lu, C., Hou, Q., Cheng, M.M., Fu, D., Shen, X., Feng, J.: Contrastive masked autoencoders are stronger vision learners. IEEE transactions on pattern analysis and machine intelligence **PP** (2022)
26. Li, J., Wang, Y., ZHANG, X., Chen, Y., Jiang, D., Dai, W., Li, C., Xiong, H., Tian, Q.: Progressively compressed auto-encoder for self-supervised representation learning. In: The Eleventh International Conference on Learning Representations (2023)
27. Li, Z., Zhu, Y., Yang, F., Li, W., Zhao, C., Chen, Y., Chen, Z., Xie, J., Wu, L., Zhao, R., Tang, M., Wang, J.: Univip: A unified framework for self-supervised visual pre-training. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 14607–14616 (2022)
28. Lin, J., Rao, Y., Lu, J., Zhou, J.: Runtime neural pruning. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
29. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (2014)
30. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9992–10002 (2021)
31. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. ArXiv **abs/1711.05101** (2017)
32. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. ArXiv **abs/1807.03748** (2018)
33. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024)

34. Ruan, Y., Singh, S., Morningstar, W.R., Alemi, A.A., Ioffe, S., Fischer, I., Dillon, J.V.: Weighted ensemble self-supervised learning. In: The Eleventh International Conference on Learning Representations (2023)
35. Song, K., Xie, J., Zhang, S., Luo, Z.: Multi-mode online knowledge distillation for self-supervised visual representation learning. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11848–11857 (2023)
36. Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., Dai, J.: Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 14411–14420 (2021)
37. Team, G.G.: Introducing gemini: our largest and most capable ai model. Google (2023)
38. Tian, K., Jiang, Y., qishuai diao, Lin, C., Wang, L., Yuan, Z.: Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. In: The Eleventh International Conference on Learning Representations (2023)
39. Tomasev, N., Bica, I., McWilliams, B., Buesing, L., Pascanu, R., Blundell, C., Mitrovic, J.: Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? arXiv preprint arXiv:2201.05119 (2022)
40. Touvron, H., Cord, M., Oquab, M., Bojanowski, P., Verbeek, J., Jégou, H.: Co-training 2l submodels for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11701–11710 (June 2023)
41. Wang, X., Huang, Y., Zeng, D., Qi, G.J.: Caco: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**, 10718–10730 (2022)
42. Wang, X., Yu, F., Dou, Z.Y., Gonzalez, J.: Skipnet: Learning dynamic routing in convolutional networks. ArXiv **abs/1711.09485** (2017)
43. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3023–3032 (2020)
44. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., Schmidt, L.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International Conference on Machine Learning (ICML) (2022)
45. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Neural Information Processing Systems (2021)
46. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. ArXiv **abs/1807.10221** (2018)
47. Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., Hu, H.: Self-supervised learning with swin transformers. ArXiv **abs/2105.04553** (2021)
48. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: a simple framework for masked image modeling. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9643–9653 (2021)
49. Yan, B., Jiang, Y., Sun, P., Wang, D., Yuan, Z., Luo, P., Lu, H.: Towards grand unification of object tracking. In: European Conference on Computer Vision (2022)
50. Yan, C., Chang, X., Li, Z., Yao, L., Luo, M., Zheng, Q.: Masked distillation advances self-supervised transformer architecture search. In: The Twelfth Interna-

tional Conference on Learning Representations (2024), https://openreview.net/forum?id=LUpC8KTvdV

51. Yu, F., Huang, K., Wang, M., Cheng, Y., Chu, W., Cui, L.: Width & depth pruning for vision transformers. In: AAAI Conference on Artificial Intelligence (2022)
52. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved VQGAN. In: International Conference on Learning Representations (2022)
53. Zhang, Y., Zhong, Q., Ma, L., Xie, D., Pu, S.: Learning incremental triplet margin for person re-identification. AAAI Conference on Artificial Intelligence (2019)
54. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5122–5130 (2017)
55. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. International Conference on Learning Representations (ICLR) (2022)