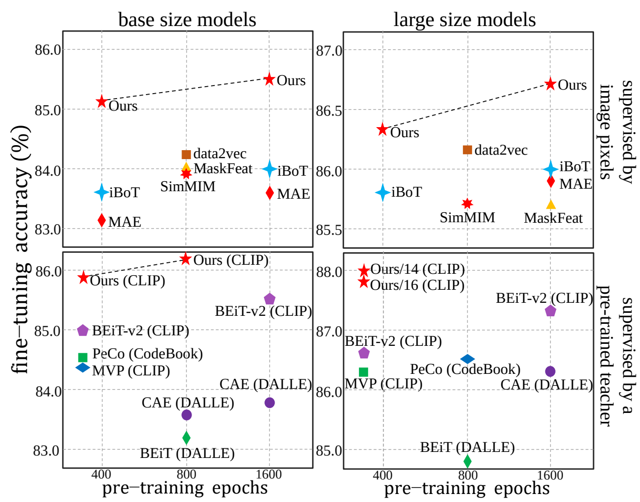
Integrally Pre-Trained Transformer Pyramid Networks

Yunjie Tian[1], Lingxi Xie[2], Zhaozhi Wang[1], Longhui Wei[2], Xiaopeng Zhang[2],
Jianbin Jiao[1]*, Yaowei Wang[3]*, Qi Tian[2]*, Qixiang Ye[1,3]*
[1]UCAS [2]Huawei Inc. [3]Pengcheng Lab.

tianyunjie19@mails.ucas.ac.cn 198808xc@gmail.com wangzhaozhi22@mails.ucas.ac.cn
weilonghui1@huawei.com zxphistory@gmail.com yaoweiwang@bit.edu.cn
jiaojb@ucas.ac.cn tian.qi1@huawei.com qxye@ucas.ac.cn

Abstract

*In this paper, we present an integral pre-training framework based on masked image modeling (MIM). We advocate for pre-training the backbone and neck jointly so that the transfer gap between MIM and downstream recognition tasks is minimal. We make two technical contributions. **First**, we unify the reconstruction and recognition necks by inserting a feature pyramid into the pre-training stage. **Second**, we complement mask image modeling (MIM) with masked feature modeling (MFM) that offers multi-stage supervision to the feature pyramid. The pre-trained models, termed integrally pre-trained transformer pyramid networks (iTPNs), serve as powerful foundation models for visual recognition. In particular, the base/large-level iTPN achieves an **86.2%/87.8%** top-1 accuracy on ImageNet-1K, a **53.2%/55.6%** box AP on COCO object detection with 1× training schedule using Mask-RCNN, and a **54.7%/57.7%** mIoU on ADE20K semantic segmentation using UPerHead – all these results set new records. Our work inspires the community to work on unifying upstream pre-training and downstream fine-tuning tasks. Code is available at github.com/sunsmarterjie/iTPN.*



base-level models	IN-1K cls. acc.	COCO (MR, 1×) det. AP	COCO (MR, 1×) seg. AP	ADE20K (UP) seg. mIoU
iTPN-B	**86.2**	**53.2**	**46.6**	**54.7**
prev. best	85.5 [50]	50.0 [13]	44.0 [13]	53.0 [50]
large-level models	**IN-1K cls. acc.**	**COCO (MR, 1×) det. AP**	**COCO (MR, 1×) seg. AP**	**ADE20K (UP) seg. mIoU**
iTPN-L	**87.8**	**55.6**	**48.6**	**57.7**
prev. best	87.3 [50]	54.5 [13]	47.6 [13]	56.7 [50]

Figure 1. **Top**: on ImageNet-1K classification, iTPN shows significant advantages over prior methods, either only using pixel supervision (top) or leveraging knowledge from a pre-trained teacher (bottom, in the parentheses lies the name of teacher model). **Bottom**: iTPN surpasses previous best results in terms of recognition accuracy (%) on several important benchmarks. Legends – IN-1K: ImageNet-1K, MR: Mask R-CNN [30], UP: UPerHead [66].

1. Introduction

Recent years have witnessed two major progresses in visual recognition, namely, the vision transformer architecture [22] as network backbone and masked image modeling (MIM) [3, 28, 68] for visual pre-training. Combining these two techniques yields a generalized pipeline that achieves state-of-the-arts in a wide range of visual recognition tasks, including image classification, object detection, and instance/semantic segmentation.

One of the key issues of the above pipeline is the transfer gap between upstream pre-training and downstream fine-tuning. From this point of view, we argue that downstream visual recognition, especially fine-scaled recognition (*e.g.*, detection and segmentation), requires hierarchical visual

*Corresponding author.

features. However, most existing pre-training tasks (*e.g.*, BEiT [3] and MAE [28]) were built upon plain vision transformers. Even if hierarchical vision transformers have been used (*e.g.*, in SimMIM [68], ConvMAE [25], and GreenMIM [33]), the pre-training task only affects the backbone but leaves the neck (*e.g.*, a feature pyramid) un-trained. This brings extra risks to downstream fine-tuning as the optimization starts with a randomly initialized neck which is not guaranteed to cooperate with the pre-trained backbone.

In this paper, we present an integral pre-training framework to alleviate the risk. We establish the baseline with HiViT [74], an MIM-friendly hierarchical vision transformer, and equip it with a feature pyramid. To jointly optimize the backbone (HiViT) and neck (feature pyramid), we make two-fold technical contributions. **First**, we unify the upstream and downstream necks by inserting a feature pyramid into the pre-training stage (for reconstruction) and reusing the weights in the fine-tuning stage (for recognition). **Second**, to better pre-train the feature pyramid, we propose a new masked feature modeling (MFM) task that (i) computes intermediate targets by feeding the original image into a moving-averaged backbone, and (ii) uses the output of each pyramid stage to reconstruct the intermediate targets. MFM is complementary to MIM and improves the accuracy of both reconstruction and recognition. MFM can also be adapted to absorb knowledge from a pre-trained teacher (*e.g.*, CLIP [52]) towards better performance.

The obtained models are named integrally pre-trained pyramid transformer networks (iTPNs). We evaluate them on standard visual recognition benchmarks. As highlighted in Figure 1, the iTPN series report the best known downstream recognition accuracy. **On COCO and ADE20K**, iTPN largely benefits from the pre-trained feature pyramid. For example, the base/large-level iTPN reports a $\mathbf{53.2}\%$/$\mathbf{55.6}\%$ box AP on COCO ($1\times$ schedule, Mask R-CNN) and a $\mathbf{54.7}\%$/$\mathbf{57.7}\%$ mIoU on ADE20K (UPerNet), surpassing all existing methods by large margins. **On ImageNet-1K**, iTPN also shows significant advantages, implying that the backbone itself becomes stronger during the joint optimization with neck. For example, the base/large-level iTPN reports an $\mathbf{86.2}\%$/$\mathbf{87.8}\%$ top-1 classification accuracy, beating the previous best record by 0.7%/0.5%, which is not small as it seems in such a fierce competition. In diagnostic experiments, we show that iTPN enjoys both (i) a lower reconstruction error in MIM pre-training and (ii) a faster convergence speed in downstream fine-tuning – this validates that shrinking the transfer gap benefits both upstream and downstream parts.

Overall, the key contribution of this paper lies in the integral pre-training framework that, beyond setting new state-of-the-arts, enlightens an important future research direction – unifying upstream pre-training and downstream fine-tuning to shrink the transfer gap between them.

2. Related Work

In the deep learning era [36], visual recognition algorithms are mostly built upon deep neural networks. There are two important network backbones in the past decade, namely, the **convolutional neural networks** [31,35,53] and the **vision transformers** [23,45,59,74]. This paper focuses on the vision transformers which were transplanted from the natural language processing field [58]. The core idea is to extract visual features by treating each image patch as a token and computing self-attentions among them.

The **vanilla vision transformers** appeared in a plain form [11,23,41,70,75] where, throughout the backbone, the number of tokens keeps a constant and the attention among these tokens are totally symmetric. To compensate the inductive priors in computer vision, the community designed **hierarchical vision transformers** [16,45,59,65,74] that allow the number of tokens to gradually decrease throughout the backbone, *i.e.*, similar as in convolutional neural networks. Other design principles were also inherited, such as introducing convolution into the transformer architecture so that the relationship between neighborhood tokens is better formulated [16,24,40,47,59,60], interacting between hybrid information [51], using window [20, 45, 74] or local [71] self-attentions to replace global self-attentions, adjusting the geometry for local-global interaction [69], decomposing self-attentions [57], and so on. It was shown that hierarchical vision transformers can offer high-quality, multi-level visual features that easily cooperate with a neck module (for feature aggregation, *e.g.*, a feature pyramid [42]) and benefit downstream visual recognition tasks.

The continuous growth of vision data calls for visual pre-training, in particular, **self-supervised learning** that learns generic visual representations from unlabeled vision data. At the core of self-supervised learning lies a pretext task, *i.e.*, an unsupervised learning objective that the model pursues. The community started with preliminary pretext tasks such as geometry-based tasks (*e.g.*, determining the relative position between image patches [19, 48, 62] or the transformation applied to an image [26]), and generation-based tasks (*e.g.*, recovering the removed contents [49] or attributes [55, 56, 72] of an image), but these methods suffer unsatisfying accuracy (*i.e.*, trailed by fully-supervised pre-training significantly) when transferred to downstream recognition tasks. The situation was changed when new pretext tasks were introduced, in particular, contrastive learning [7, 8, 12, 27, 27, 29, 67] and **masked image modeling** (MIM) [1, 3, 15, 28, 37, 38, 68], where the latter is yet another type of generation-based learning objective.

This paper focuses on MIM, which takes the advantage of vision transformers that formulate each image patch into a token. Hence, the tokens can be arbitrarily masked (discarded from the input data) and the learning objective is to recover the masked contents at the pixel level [28, 55, 68],

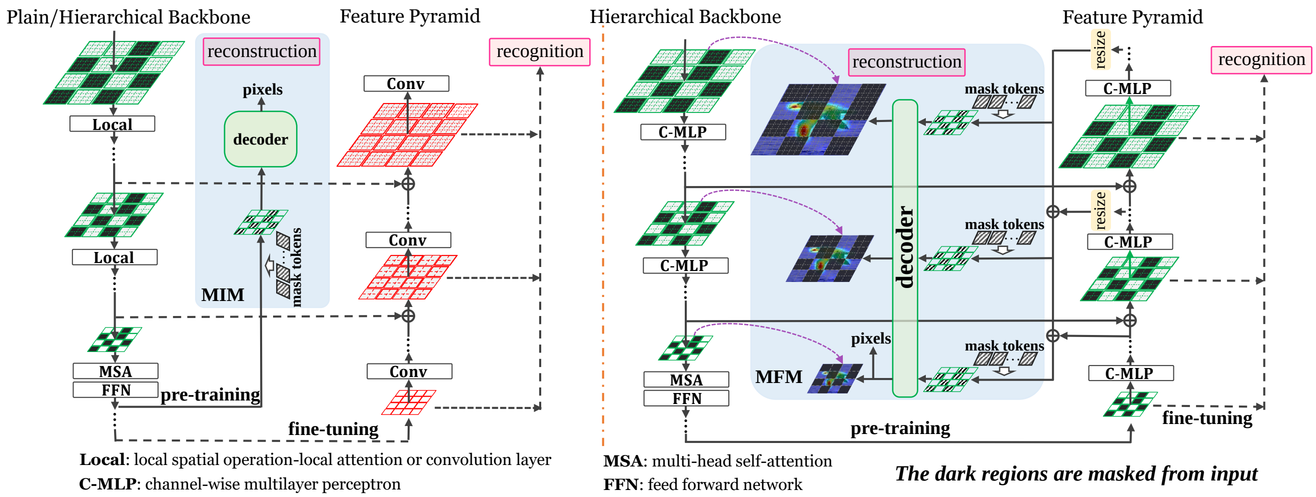


Figure 2. The comparison between a conventional pre-training (left) and the proposed integral pre-training framework (right). We use a feature pyramid as the unified **neck** module, and apply masked feature modeling for pre-training the feature pyramid. The green and red blocks indicate that the network weights are pre-trained and un-trained (*i.e.*, randomly initialized for fine-tuning), respectively.

the feature level [3, 61], or in the frequency space [44]. MIM has shown an important property named scalability, *i.e.*, augmenting the amount of pre-training data (*e.g.*, from ImageNet-1K to ImageNet-22K) and/or increasing the model size (*e.g.*, from the base level to the large or huge level) can boost the downstream performance [13,28], which aligns with the observations in language modeling [5, 18].

Most existing MIM methods worked on the plain vision transformers, yet the hierarchical vision transformers have higher potentials in visual recognition. The first work which tried to bridge the gap was SimMIM [68], but the overall pre-training overhead was largely increased because the entire image (with dummy masked patches) were fed to the encoder. This issue was later alleviated by reforming the hierarchical vision transformers [33, 74] to fit MIM better. This paper inherits the design and goes one step further by integrating the neck (*e.g.*, a feature pyramid) into the pre-training phase, constructing the integrally pre-trained transformer pyramid network.

3. The Proposed Approach

3.1. Motivation: Integral Pre-Training

We first establish a notation system. The pre-training stage is built upon a dataset $\mathcal{D}^{\mathrm{pt}} = \{\mathbf{x}_n^{\mathrm{pt}}\}_{n=1}^{N}$, where N is the number of samples. Note that these samples are not equipped with labels. The fine-tuning phase involves another dataset $\mathcal{D}^{\mathrm{ft}} = \{\mathbf{x}_m^{\mathrm{ft}}, \mathbf{y}_m^{\mathrm{ft}}\}_{m=1}^{M}$, where M is the number of samples and $\mathbf{y}_m^{\mathrm{ft}}$ is the semantic label of $\mathbf{x}_m^{\mathrm{ft}}$. Let the target deep neural network be composed of backbone, neck, and head[1], denoted as $f(\cdot;\boldsymbol{\theta})$, $g(\cdot;\boldsymbol{\phi})$, $h(\cdot;\boldsymbol{\psi})$, respectively, where $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ are learnable parameters and can be omitted for simplicity. $f(\cdot)$ directly takes $\mathbf{x}$ as input, while $g(\cdot)$ and $h(\cdot)$ works on the outputs of $f(\cdot)$ and $g(\cdot)$, *i.e.*, the entire function is $h(g(f(\mathbf{x};\boldsymbol{\theta});\boldsymbol{\phi});\boldsymbol{\psi})$.

Throughout this paper, the pre-training task is masked image modeling (MIM) and the fine-tuning tasks can be image classification, object detection, and instance/semantic segmentation. Existing approaches assumed that they share the same **backbone**, but need different **necks** and **heads**. Mathematically, the pre-training and fine-tuning objectives are written as:

$$\begin{aligned} &\min \mathbb{E}_{\mathcal{D}^{\mathrm{pt}}} \|\mathbf{x}_n^{\mathrm{pt}} - h^{\mathrm{pt}}(g^{\mathrm{pt}}(f(\mathbf{x}_n^{\mathrm{pt}};\boldsymbol{\theta}),\boldsymbol{\phi}^{\mathrm{pt}}),\boldsymbol{\psi}^{\mathrm{pt}})\|, \\ &\min \mathbb{E}_{\mathcal{D}^{\mathrm{ft}}} \|\mathbf{y}_m^{\mathrm{ft}} - h^{\mathrm{ft}}(g^{\mathrm{ft}}(f(\mathbf{x}_m^{\mathrm{ft}};\boldsymbol{\theta}),\boldsymbol{\phi}^{\mathrm{ft}}),\boldsymbol{\psi}^{\mathrm{ft}})\|, \end{aligned} \tag{1}$$

where parameters are not shared between $\boldsymbol{\phi}^{\mathrm{pt}}$, $\boldsymbol{\phi}^{\mathrm{ft}}$ and $\boldsymbol{\psi}^{\mathrm{pt}}$, $\boldsymbol{\psi}^{\mathrm{ft}}$. We argue that such a pipeline leads to a significant transfer gap between pre-training and fine-tuning, and thus brings two-fold drawbacks. **First**, the backbone parameters, $\boldsymbol{\theta}$, are not optimized towards being used for multi-level feature extraction. **Second**, the fine-tuning phase starts with optimizing a randomly initialized $\boldsymbol{\phi}^{\mathrm{ft}}$ and $\boldsymbol{\psi}^{\mathrm{ft}}$, which may slow down the training procedure and lead to unsatisfying recognition results. To alleviate the gap, we advocate for an integral framework that unifies $g^{\mathrm{pt}}(\cdot)$ and $g^{\mathrm{ft}}(\cdot)$, so that the pre-trained $\boldsymbol{\phi}^{\mathrm{pt}}$ is easily reused to be an initialization of $\boldsymbol{\phi}^{\mathrm{ft}}$, and thus only $\boldsymbol{\psi}^{\mathrm{ft}}$ is randomly initialized.

The overall framework is illustrated in Figure 2.

[1] We follow the conventional definition that the neck is used for multi-stage feature aggregation (*e.g.*, a feature pyramid [42]) while the head is used for final prediction (*e.g.*, a linear classifier).

3.2. Unifying Reconstruction and Recognition

Let a hierarchical vision transformer contain S stages and each stage has several transformer blocks. Most often, the backbone (*a.k.a* encoder) gradually down-samples the input signal and produces $S+1$ feature maps:

$$f(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}^0, \mathbf{U}^1, \ldots, \mathbf{U}^S, \tag{2}$$

where $\mathbf{U}^0$ denotes the direct embedding of input, and a smaller superscript index indicates a stage closer to the input layer. Each feature map is composed of a set of **tokens** (feature vectors), *i.e.*, $\mathbf{U}^s = \{\mathbf{u}_1^s, \mathbf{u}_2^s, \ldots, \mathbf{u}_{K^s}^s\}$, where K^s is the number of tokens in the s-th feature map.

We show that $g^{\mathrm{pt}}(\cdot)$ and $g^{\mathrm{ft}}(\cdot)$ can share the same architecture and parameters because both of them start with $\mathbf{U}^S$ and gradually aggregate it with lower-level features. Thus, we write the neck part as follows:

$$\begin{aligned} &\mathbf{V}^S = \mathbf{U}^S, \\ &\mathbf{V}^s = \mathbf{U}^s + g^s(\mathbf{V}^{s+1};\boldsymbol{\phi}^s), \quad 1 \leqslant s < S, \end{aligned} \tag{3}$$

where $g^s(\cdot)$ up-samples $\mathbf{V}^{s+1}$ to fit the resolution of $\mathbf{V}^s$. Note that the learnable parameters, $\boldsymbol{\phi}$, are composed of a layer-wise parameter set, $\{\boldsymbol{\phi}^s\}$. With these parameters being reused in fine-tuning, we largely shrink the transfer gap: the only modules that remain individual between pre-training and fine-tuning are the heads (*e.g.*, the decoder for MIM *vs.* the Mask R-CNN head for detection).

Before entering the next part that discusses the loss terms, we remind the readers that other differences exist between pre-training and fine-tuning, while they do not impact the overall design of network architectures.

- MIM samples a random mask $\mathcal{M}$ and applies it to $\mathbf{x}$, *i.e.*, $\mathbf{x}$ is replaced by $\mathbf{x} \odot \overline{\mathcal{M}}$. Consequently, all the backbone outputs, $\mathbf{U}^0, \ldots, \mathbf{U}^S$, do not contain the tokens with indices in $\mathcal{M}$, and so are $\mathbf{V}^1, \ldots, \mathbf{V}^S$. At the start of decoder, $\mathbf{V} = \sum_{s=1}^{S} \mathbf{V}^s$ is complemented by adding dummy tokens to the masked indices, and then fed into a decoder for image reconstruction.

- The downstream fine-tuning procedure makes use of specific outputs of decoder for different tasks. For image classification, $\mathbf{V}^S$ is used. For detection and segmentation, all of $\mathbf{V}^1, \ldots, \mathbf{V}^S$ are used.

3.3. Masked Feature Modeling

We first inherit the reconstruction loss from MIM that minimizes $\|\mathbf{x} - h^{\mathrm{pt},0}(\mathbf{V};\boldsymbol{\psi}^{\mathrm{pt},0})\|$, where $h^{\mathrm{pt},0}(\cdot)$ involves a few transformer blocks that reconstruct the original image from $\mathbf{V} = \sum_{s=1}^{S} \mathbf{V}^s$. To acquire the ability of capturing multi-stage features, we add a reconstruction head to each stage, termed $h^{\mathrm{pt},s}(\cdot;\boldsymbol{\psi}^{\mathrm{pt},s})$, and optimize the following multi-stage loss:

$$\mathcal{L} = \underbrace{\|\mathbf{x} - h^{\mathrm{pt},0}(\mathbf{V})\|}_{\text{image reconstruction}} + \lambda \cdot \underbrace{\sum_{s=1}^{S} \|\mathbf{x}^s - h^{\mathrm{pt},s}(\mathbf{V}^s)\|}_{\text{feature reconstruction}}, \tag{4}$$

where $\mathbf{x}^s$ is the expected output at the s-th decoder stage, and $\lambda = 0.3$ is determined in a held-out validation set. Since the goal is to recover the masked features, we name the second term as the **masked feature modeling** (MFM) loss that complements the first term, the masked image modeling (MIM) loss. We illustrate MFM in Figure 2.

The remaining issue is to determine the intermediate reconstruction target, *i.e.*, $\mathbf{x}^1, \ldots, \mathbf{x}^S$. We borrow the idea from knowledge distillation [32] that makes use of a teacher backbone $\hat{f}^{\mathrm{back}}(\cdot)$ to generate the intermediate targets, *i.e.*, $\hat{f}(\mathbf{x};\hat{\boldsymbol{\theta}}) = \mathbf{x}^1, \ldots, \mathbf{x}^S$. The teacher model is chosen to be the moving-averaged [54] encoder (in this case, no external knowledge is introduced) or another pre-trained model (*e.g.*, CLIP [52], as used by [63, 64], that was pre-trained on a large dataset of image-text pairs). In the former case, we only feed the masked patches ($\mathbf{x} \odot \mathcal{M}$, not the entire image) to the teacher model for acceleration. In the latter case, we follow BEiT [3] to feed the entire image to the pre-trained CLIP model.

3.4. Technical Details

We build the system beyond HiViT [74], a recently proposed, hierarchical vision transformer. HiViT simplified the Swin transformers [45] by (i) replacing early-stage shifted-window attentions with channel-wise multi-layer perceptrons (C-MLPs) and (ii) removing the 7×7 stage so that global attentions are computed on the 14×14 stage. With these improvements, when applied to MIM, HiViT allows the masked tokens to be directly discarded from input (by contrast, with Swin as the backbone, SimMIM [68] required the entire image to be used as input), saving 30%–50% computational costs and leading to better performance.

Table 1 summarizes the configuration of iTPN. We follow the convention to use 224×224 images during the pre-training. HiViT produces three stages ($S = 3$) with spatial resolutions of 56×56, 28×28, and 14×14, respectively. An S-stage feature pyramid is built upon the backbone. We replace all convolutions in the feature pyramid with C-MLPs to avoid leaking information from visible patches to invisible patches. As we shall see in ablation (Section 4.4), using C-MLP in the feature pyramid leads to consistent accuracy gain in various visual recognition tasks, and the improvement is complementary to that brought by MFM.

Regarding MFM, we investigate two choices of the teacher model. (i) The first option involves computing the exponential moving average (EMA) of the online target

Table 1. A comparison between ViT, Swin, HiViT, and the proposed iTPN in terms of network configuration. We use 224×224 input size to calculate the FLOPs. †: We add 4 Stage-3 blocks to HiViT-B to keep the FLOPs of iTPN-B comparable to ViT.

Model	ViT	Swin [45]	HiViT [74]	iTPN
base-level models				
# stages	1	4	3	3
# blocks	12	2+2+18+2	3+3+20	3+3+24†
Params (M)	86	88	66	79
FLOPs (G)	17.5	15.4	15.9	17.8
large-level models				
# stages	1	4	3	3
# blocks	24	2+2+18+2	2+2+40	2+2+40
Params (M)	307	197	288	288
FLOPs (G)	61.3	34.5	61.2	61.2

model with a coefficient of 0.996. We extract the supervision from the last layer of each stage, so that for any s, $\mathbf{x}^s$ has the same spatial resolution as $\mathbf{V}^s$, and thus $h^s(\cdot)$ is a linear projection working on each token individually. (ii) The second option directly inherits a CLIP pre-trained model. Note that CLIP offers standard ViTs that do not produce multi-resolution feature maps. In this scenario, we unify the S MFM terms into one by down-sampling all the feature maps to the lowest spatial resolution (14×14), adding them together, and comparing the sum to the last-layer output of the CLIP model.

4. Experiments

4.1. Settings and Implementation Details

We pre-train iTPN on the ImageNet-1K dataset [17], a subset of ImageNet that contains 1.28M training images of 1,000 classes. The class labels are not used during the pre-training stage. Each training image is pre-processed into 224×224 and partitioned into 14×14 patches sized 16×16 pixels. Following MAE [28], a random subset of 75% patches are masked from input, and the normalized pixels are preserved for reconstruction.

We use an AdamW optimizer [46] with an initial learning rate of 1.5×10^{-4}, a weight decay of 0.05, and batch size of 4,096. The learning rate follows a cosine annealing schedule and the number of warm-up epochs is set to be 40. The numbers of pre-training epochs are 400 and 1,600 in the former scenario, or 300 and 800 in the latter scenario[2]. We train all these models using 64 NVIDIA Tesla-V100 GPUs. For the base-level models, one pixel-supervised epoch and one CLIP-supervised epoch take about 2.7 and 4.7 minutes, respectively. For the large-level models, the numbers are 4.2 and 12.0 minutes, respectively. That said, a 1600-epoch pixel-supervised pre-training of iTPN-base/large takes around 75/115 hours.

[2]By using CLIP as supervision, each pre-training epoch takes longer time but the pre-training converges faster. So, we adjust the number of pre-training epochs according to the computational budget.

4.2. Image Classification

Fine-tuning We report results of ImageNet-1K classification. The number of epochs is 100 for base-level models and 50 for large-level models. We use the AdamW optimizer, with the initial learning rate being 5×10^{-4} and 1×10^{-3} for base-level and large-level models, respectively. The weight decay is 0.05 and the batch size is 1,024. The number of warm-up epochs is 5. The layer decay is set to be 0.55 and 0.50 for base-level and large-level models.

Results are summarized in Table 2. One can see that iTPN achieves higher accuracy than existing methods on all tracks, *i.e.*, using base-level or large-level backbones, with or without external supervision (*i.e.*, CLIP [52]). For example, using the base-level backbone, iTPN achieves an 85.1% accuracy with only 400 pre-training epochs, surpassing MAE [28] and HiViT [74] with 1,600 epochs. The accuracy of iTPN continues growing to 85.5% with 1,600 pre-training epochs, which is on par with BEiT-v2 [50] that distilled knowledge from CLIP-B [52] (1,600 epochs), yet iTPN reports an 86.2% accuracy with the supervision of CLIP (800 epochs). Similar situations occur when we use the large-level backbone, where the advantage of iTPN is a bit smaller due to the higher baseline. The best practice appears when an iTPN-L/14 model (*i.e.*, patch size is adjusted to 14×14) is supervised by a CLIP-L teacher – the classification accuracy, 88.0%, is the highest to date under fair comparisons.

Linear probing We then evaluate the pre-trained models using the linear probing Following the convention, we train the models for 90 epochs using the LARS optimizer [34] with a batch size of 16,384 and a learning rate of 0.1. Specifically, the iTPN-B (pixel) model with 1,600 pre-training epochs reports a 71.6% accuracy, surpassing 1,600-epoch MAE [28] by a significant margin of 3.8%, as well as 400-epoch BEiT, 800-epoch SimMIM, and 1,600-epoch CAE by 22.2%, 14.9%, and 1.2%, respectively. With CLIP supervision, iTPN with 300 epochs of pre-training reports a 77.3% accuracy, surpassing MVP [63] with the same setting by 1.9%.

Insights Note that image classification experiments do not involve transferring the pre-trained neck, *i.e.*, the feature pyramid. That said, iTPN achieves higher classification accuracy with the pre-trained backbone alone. This implies that (i) a joint optimization of the backbone and neck leads to a stronger backbone, and hence, (ii) the derived backbone can be directly transferred for various vision tasks, extending iTPN to more application scenarios.

Table 2. Top-1 classification accuracy (%) by fine-tuning the pre-trained models on ImageNet-1K. We compare models of different levels and supervisions (*e.g.*, with and without CLIP) separately.

Method at base-level	Arch.	Sup.	Eps.	Param. (M)	FT acc.
BEiT [3]	ViT-B	DALL-E	400	86	83.2
CAE [13]	ViT-B	DALL-E	800	86	83.6
MaskFeat [61]	ViT-B	HOG	800	86	84.0
SimMIM [68]	Swin-B	pixel	800	88	84.0
data2vec [2]	ViT-B	pixel	800	86	84.2
BootMAE [21]	ViT-B	pixel	800	86	84.2
ConvMAE [25]	ConViT-B	pixel	1600	88	85.0
MAE [28]	ViT-B	pixel	1600	86	83.6
HiViT [74]	HiViT-B	pixel	800	66	84.2
iTPN (ours)	HiViT-B	pixel	400	79	**85.1**
iTPN (ours)	HiViT-B	pixel	1600	79	**85.5**
MVP [63]	ViT-B	CLIP-B	300	86	84.4
BEiT-v2 [50]	ViT-B	CLIP-B	1600	86	85.5
CAE-v2 [73]	ViT-B	CLIP-B	300	86	85.3
iTPN (ours)	HiViT-B	CLIP-B	300	79	**85.9**
iTPN (ours)	HiViT-B	CLIP-B	800	79	**86.2**
Method at large-level	Arch.	Sup.	Eps.	Param. (M)	FT acc.
BEiT [3]	ViT-L	DALL-E	800	307	85.2
MaskFeat [61]	ViT-L	HOG	300	307	84.4
MaskFeat [61]	ViT-L	HOG	1600	307	85.7
SimMIM [68]	Swin-L	pixel	800	197	85.4
SimMIM [68]	Swin-H	pixel	800	658	85.7
data2vec [2]	ViT-L	pixel	800	307	86.2
BootMAE [21]	ViT-L	pixel	800	307	85.9
CAE [13]	ViT-L	DALL-E	1600	307	86.3
MAE [28]	ViT-L	pixel	1600	307	85.9
HiViT [74]	HiViT-L	pixel	1600	288	86.1
iTPN (ours)	HiViT-L	pixel	400	288	**86.3**
iTPN (ours)	HiViT-L	pixel	1600	288	**86.7**
MVP [63]	ViT-L/16	CLIP-B	300	307	86.3
BEiT-v2 [50]	ViT-L/16	CLIP-B	300	307	86.6
CAE-v2 [73]	ViT-L	CLIP-B	300	307	86.7
iTPN (ours)	HiViT-L/16	CLIP-B	300	288	**87.0**
iTPN (ours)	HiViT-L/16	CLIP-L	300	288	**87.8**
iTPN (ours)	HiViT-L/14	CLIP-L	300	288	**88.0**
ImageNet-22K + 384 input size					
iTPN (ours)	HiViT-L/16	CLIP-L	300	288	**89.2**

4.3. Detection and Segmentation

COCO: object detection & instance segmentation We follow the configuration provided by [13] to evaluate the pre-trained models on the COCO [43] dataset. We use Mask R-CNN [30] implemented by MMDetection [10]. We use the AdamW optimizer [46] with a weight decay of 0.05. The standard $1\times$ (12 epochs) and $3\times$ schedules are applied, where the initial learning rate is 3×10^{-4} and it decays by a factor of 10 after $3/4$ and $11/12$ of fine-tuning epochs. The layer-wise decay rate is set to be 0.90. We also try a $3\times$ Cascade Mask R-CNN [6] towards higher accuracy.

Table 3. Top-1 linear probing (LIN) classification accuracy (%) by training the last classifier layer on ImageNet-1K. We compare models of different supervisions (*e.g.*, with and without CLIP) separately, using the base-level models.

Method at base-level	Arch.	Sup.	Eps.	Param. (M)	LIN acc.
BEiT [3]	ViT-B	DALL-E	400	86	49.4
MAE [28]	ViT-B	pixel	1600	86	67.8
SimMIM [68]	ViT-B	pixel	800	86	56.7
CAE [13]	ViT-B	DALL-E	1600	86	70.4
ConvMAE [25]	ConViT-B	pixel	1600	88	70.9
iTPN (ours)	HiViT-B	pixel	1600	79	**71.6**
BEiT-v2 [50]	ViT-B	CLIP-B	1600	86	80.1
MVP [63]	ViT-B	CLIP-B	300	86	75.4
iTPN (ours)	HiViT-B	CLIP-B	300	79	**77.3**

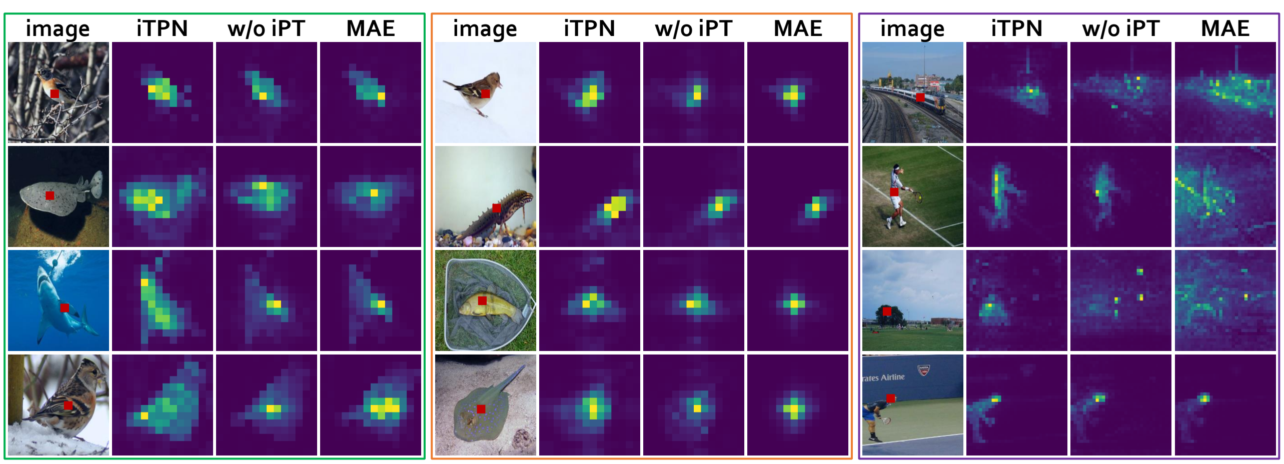
Results are summarized in Table 4. Compared to image classification, the advantages of iTPN become more significant because the pre-trained neck is reused so that the fine-tuning stage only needs to initialize a task-specific head. For example, using a pixel-supervised base-level backbone, the $1\times$ Mask R-CNN produces 53.0% box AP, surpassing all other methods significantly (*e.g.*, +4.6% over MAE [28] and +3.0% over CAE [13]). Compared to HiViT that did not pre-train the feature pyramid, iTPN claims a +1.7% gain in box AP. With stronger heads, iTPN reports stronger numbers, *e.g.*, the box/mask AP is 56.0%/48.5% using $3\times$ Cascade Mask R-CNN, setting a new record with base-level models. Later, we will show that the benefits indeed come from pre-training the feature pyramid and loading it for downstream fine-tuning.

ADE20K: semantic segmentation We follow BEiT [3] to build a UperHead [66] on top of the pre-trained backbone. We use the AdamW optimizer [46] and the learning rate is fixed as 3×10^{-5}. We fine-tune the model for a total of $160k$ iterations and the batch size is 16. The input resolution is 512×512 and we do not use a multi-scale test. Results are summarized in Table 4. Again, iTPN reports the best accuracy in terms of mIoU. In particular, the pixel-supervised base/large-level models report 53.5%/56.1% mIoUs which surpass all the competitors substantially. Introducing CLIP supervision further improves both numbers by more than 1%, setting solid new records for both base-level and large-level models.

Table 4. Visual recognition results (%) on COCO (object detection and instance segmentation, AP) and ADE20K (semantic segmentation, mIoU). Mask R-CNN (*abbr.* MR, 1×/3×) and Cascade Mask R-CNN (*abbr.* CMR, 1×) are used on COCO, and UPerHead with 512×512 input is used on ADE20K. For the base-level models, each cell of COCO results contains object detection (box) and instance segmentation (mask) APs. For the large-level models, the accuracy of 1× Mask R-CNN surpasses all existing methods. †: ConvMAE used a different setting from all other methods – fine-tuning using ViTDet [39] for 25 epochs. ‡: More techniques, such as multi-scale test and softnms [4] etc, are used in the test stage.

Method *at base-level*	Arch.	Sup.	Eps.	Param. (M)	COCO MR, 1×	COCO MR, 3×	COCO CMR, 3×	ADE20K UPerHead
MoCo-v3 [14]	ViT-B	pixel	300	86	45.5/40.5	–	–	47.3
BEiT [3]	ViT-B	DALL-E	400	86	42.1/37.8	–	–	47.1
DINO [9]	ViT-B	pixel	400	86	46.8/41.5	–	–	47.2
iBoT [76]	ViT-B	pixel	1600	86	–	–	51.2/44.2	50.0
CAE [13]	ViT-B	DALL-E	1600	86	50.0/44.0	–	–	50.2
SimMIM [68]	Swin-B	pixel	800	88	–	52.3/–	–	52.8
MAE [28]	ViT-B	pixel	1600	86	48.4/42.6	–	–	48.1
ConvMAE [25]	ConViT-B	pixel	1600	88	–	53.2/47.1†	–	51.7
HiViT [28]	HiViT-B	pixel	1600	66	51.3/44.6	53.3/47.0	–	52.8
MVP [63]	ViT-B	CLIP-B	300	86	–	–	53.5/46.3	52.4
iTPN (ours)	HiViT-B	pixel	1600	79	**53.0/46.5**	**54.0/47.4**	**56.0/48.5**	**53.5**
iTPN (ours)	HiViT-B	CLIP-B	800	79	**53.2/46.6**	**54.1/47.5**	**56.1/48.6**	**54.7**

Method *at large-level*	Arch.	Sup.	Eps.	Param. (M)	COCO object det.	COCO instance seg.	COCO *schedule*	ADE20K UPerHead
MAE [28]	ViT-L	pixel	1600	307	54.0	47.1	MR, 1×	53.6
SimMIM [68]	Swin-L	pixel	800	197	53.8	–	MR, 3×	53.5
SimMIM [68]	SwinV2-H	pixel	800	658	54.4	–	MR, 3×	54.2
CAE [28]	ViT-L	pixel	1600	304	54.5	47.6	MR, 1×	54.7
iTPN (ours)	HiViT-L	pixel	1600	288	**55.6**	**48.6**	MR, 1×	**56.1**
iTPN (ours)	HiViT-L	CLIP-L	300	288	**55.2**	**48.2**	MR, 1×	**57.7**
iTPN (ours)	HiViT-L	pixel	1600	288	**58.0‡**	**50.3**	CMR, 3×	**56.1**



(a) Encoder on ImageNet-1K **(b) Decoder on ImageNet-1K** **(c) Encoder on COCO**

Figure 3. A comparison between the attention maps generated by iTPN, the variant without integral pre-training (w/o iPT), and the MIM baseline (MAE [28]). In each case, the red block indicates the query token, and the attention map between the query and other tokens at the corresponding transformer block is shown. We use 224×224 input images in (a), (b), and 512×512 images in (c).

Table 5. Ablations on whether the model is integrally pre-trained (iPT) and whether the feature pyramid is loaded for detection and segmentation. Fine-tuning on ImageNet-1K does not involve loading the pyramid. The numbers are in % for classification accuracy, box AP, and mIoU. The models are pre-trained for 400 epochs. For COCO, $1\times$ Mask R-CNN is used and box AP is reported.

iPT	loaded	ImageNet-1K	COCO	ADE20K
✗	–	84.4	50.6	51.5
✓	✗	**85.1**	51.5	51.8
✓	✓		**52.1**	**52.2**

Table 6. Ablations on C-MLP and MFM. The settings remain the same as in Table 5. The * sign indicates that convolution is used (instead of C-MLP) for both the backbone and feature pyramid, which leads to worse recognition results.

C-MLP	MFM	ImageNet-1K	COCO	ADE20K
✗*	✗	84.3	49.8	50.0
✗*	✓	84.6	50.8	50.7
✓	✗	84.9	51.8	51.8
✓	✓	**85.1**	**52.1**	**52.2**

4.4. Analysis

Ablative studies Throughout this part, we use the 400-epoch pixel-supervised model for diagnosis. We first ablate the benefit of integral pre-training. As shown in Table 5, jointly optimizing the backbone and neck leads to higher recognition accuracy on all datasets including ImageNet-1K, COCO, and ADE20K. Beyond this point, loading the pre-trained feature pyramid (neck) further improves the recognition accuracy on COCO and ADE20K. This validates that the backbone itself is strengthened by iTPN, and thus it can be transferred to downstream tasks independently of the neck.

Next, we investigate the technical details of integral pre-training, in particular, using channel-wise multi-layer perceptron (C-MLP) in the feature pyramid and applying masked feature modeling (MFM) for multi-stage supervision. As shown in Table 6, both C-MLP and MFM contribute individually to recognition accuracy, meanwhile, integrating them yields even better recognition performance.

Visualization In Figure 3, we visualize the attention maps generated by iTPN and baseline methods. **(1)** On the encoder, iTPN shows the advantage of detecting complete objects on ImageNet and concentrating on the chosen object on COCO. Such ability arises because iTPN forces the model to preserve richer visual features (multi-scale feature maps), which facilitates better recognition results in downstream. (2) On the decoder, iTPN can still realize the semantic relationship between tokens, resulting in better reconstruction results (Figure 4). We owe such benefits to the

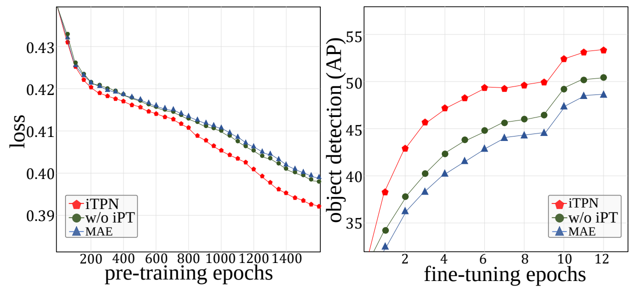


Figure 4. **Left**: the comparison of reconstruction loss values of different frameworks. **Right**: the comparison of convergence speed in terms of box AP on COCO when the pre-trained models are fine-tuned with Mask R-CNN for 12 epochs ($1\times$).

pre-trained neck that aggregates multi-stage visual features.

The benefits brought by more complete attentions can be quantified using two-fold experiments shown in Figure 4. **(1)** In the left part, we observe that iTPN achieves better reconstruction results (*i.e.*, lower reconstruction loss values). Note that simply using a hierarchical vision transformer (with multi-scale feature maps) does not improve reconstruction, implying that integral pre-training is the major contributor. **(2)** In the right part, we show that better depiction of objects helps downstream visual recognition tasks (*e.g.*, object detection) to converge faster and achieve a higher upper-bound – this aligns with the outstanding accuracy on COCO (see Section 4.3). Integrating these analysis, we conclude that iTPN successfully transfers the benefits from upstream pre-training (reconstruction) to downstream fine-tuning (recognition), completing the entire chain.

5. Conclusions and Future Remarks

In this paper, we present an integral framework for pre-training hierarchical vision transformers. The core contribution lies in a unified formulation that uses a feature pyramid for both reconstruction and recognition, so that the transfer gap between pre-training and fine-tuning is maximally reduced. Besides, a masked feature modeling (MFM) task is designed to complement masked image modeling (MIM) for a better optimization of the feature pyramid. The pre-trained iTPNs report superior recognition in a few popular visual recognition tasks. Our work clearly enlightens a future direction – designing a unified framework for upstream and downstream visual representation learning.

6. Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 62225208, 62171431 and 61836012, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 456–473. Springer, 2022. 2

[2] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022. 6

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 2, 3, 4, 6, 7

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 7

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 3

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, 43(5):1483–1498, 2019. 6

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 2

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE ICCV*, pages 9650–9660, 2021. 2

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE ICCV*, pages 9650–9660, 2021. 7

[10] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[11] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022. 2

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 2

[13] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 1, 3, 6, 7

[14] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021. 7

[15] Yabo Chen, Yuchen Liu, Dongsheng Jiang, Xiaopeng Zhang, Wenrui Dai, Hongkai Xiong, and Qi Tian. Sdae: Self-distillated masked autoencoder. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 108–124. Springer, 2022. 2

[16] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 34:3965–3977, 2021. 2

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009. 5

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE ICCV*, pages 1422–1430, 2015. 2

[20] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *IEEE CVPR*, pages 12124–12134, 2022. 2

[21] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 247–264. Springer, 2022. 6

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[24] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE ICCV*, pages 6824–6835, 2021. 2

[25] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 2, 6, 7

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[27] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE CVPR*, pages 16000–16009, 2022. 1, 2, 3, 5, 6, 7

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE CVPR*, pages 9729–9738, 2020. 2

[30] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *IEEE ICCV*, 2017. 1, 6

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 2

[32] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 4

[33] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Green hierarchical vision transformer for masked image modeling. *arXiv preprint arXiv:2205.13515*, 2022. 2, 3

[34] Zhouyuan Huo, Bin Gu, and Heng Huang. Large batch optimization for deep learning using new complete layer-wise adaptive rate scaling. In *AAAI*, 2021. 5

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012. 2

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, pages 436–444, 2015. 2

[37] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*, 2022. 2

[38] Xiaotong Li, Yixiao Ge, Kun Yi, Zixuan Hu, Ying Shan, and Ling-Yu Duan. mc-beit: Multi-choice discretization for image bert pre-training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 231–246. Springer, 2022. 2

[39] Yanghao Li, Mao Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 7

[40] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021. 2

[41] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022. 2

[42] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 2, 3

[43] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 6

[44] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. *CoRR*, abs/2204.08227, 2022. 3

[45] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE ICCV*, pages 10012–10022, 2021. 2, 4, 5

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5, 6

[47] Jiasen Lu, Roozbeh Mottaghi, Aniruddha Kembhavi, et al. Container: Context aggregation networks. *NeurIPS*, 34:19160–19171, 2021. 2

[48] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[49] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE CVPR*, pages 2536–2544, 2016. 2

[50] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 1, 5, 6

[51] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *IEEE ICCV*, pages 367–6162, 2021. 2

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 4, 5

[53] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 2

[54] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017. 4

[55] Yunjie Tian, Lingxi Xie, Jiemin Fang, Mengnan Shi, Junran Peng, Xiaopeng Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. Beyond masking: Demystifying token-based pre-training for vision transformers. *arXiv preprint arXiv:2203.14313*, 2022. 2

[56] Yunjie Tian, Lingxi Xie, Xiaopeng Zhang, Jiemin Fang, Haohang Xu, Wei Huang, Jianbin Jiao, Qi Tian, and Qixiang Ye. Semantic-aware generation for self-supervised visual representation learning. *arXiv preprint arXiv:2111.13163*, 2021. 2

[57] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *arXiv preprint arXiv:2204.01697*, 2022. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[59] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 2

[60] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2

[61] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *IEEE CVPR*, pages 14668–14678, 2022. 3, 6

[62] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *CVPR*, pages 1910–1919, 2019. 2

[63] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. *arXiv preprint arXiv:2203.05175*, 2022. 4, 5, 6, 7

[64] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022. 4

[65] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *IEEE ICCV*, pages 22–31, 2021. 2

[66] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 1, 6

[67] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *IEEE CVPR*, pages 16684–16693, 2021. 2

[68] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *IEEE CVPR*, pages 9653–9663, 2022. 1, 2, 3, 4, 6, 7

[69] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 2

[70] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, pages 558–567, 2021. 2

[71] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 2

[72] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*. Springer, 2016. 2

[73] Xinyu Zhang, Jiahui Chen, Junkun Yuan, Qiang Chen, Jian Wang, Xiaodi Wang, Shumin Han, Xiaokang Chen, Jimin Pi, Kun Yao, et al. Cae v2: Context autoencoder with clip target. *arXiv preprint arXiv:2211.09799*, 2022. 6

[74] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *International Conference on Learning Representations*, 2023. 2, 3, 4, 5, 6

[75] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 2

[76] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 7