ParameterNet: Parameters Are All You Need

Kai Han[1,*] Yunhe Wang[1,*] Jianyuan Guo[1,2,*] Enhua Wu[3,4]
[1]Huawei Noah's Ark Lab [2]The University of Sydney
[3]State Key Lab of Computer Science, ISCAS [4]University of Macau
{kai.han,yunhe.wang,jianyuan.guo}@huawei.com, weh@ios.ac.cn

Abstract

The large-scale visual pretraining has significantly improve the performance of large vision models. However, we observe the low FLOPs pitfall *that the existing low-FLOPs models cannot benefit from large-scale pretraining. In this paper, we introduce a novel design principle, termed ParameterNet, aimed at augmenting the number of parameters in large-scale visual pretraining models while minimizing the increase in FLOPs. We leverage dynamic convolutions to incorporate additional parameters into the networks with only a marginal rise in FLOPs. The ParameterNet approach allows low-FLOPs networks to take advantage of large-scale visual pretraining. Furthermore, we extend the ParameterNet concept to the language domain to enhance inference results while preserving inference speed. Experiments on the large-scale ImageNet-22K have shown the superiority of our ParameterNet scheme. For example, ParameterNet-600M can achieve higher accuracy on ImageNet than the widely-used Swin Transformer (81.6%* vs. *80.9%) and has much lower FLOPs (0.6G* vs. *4.5G). In the language domain, LLaMA-1B enhanced with ParameterNet achieves 2% higher accuracy over vanilla LLaMA. The code will be released at* https://parameternet.github.io/.

1. Introduction

Thanks to advancements in computational hardware and data engineering, large-scale visual pretraining has witnessed remarkable progress as a fundamental component in computer vision. Pretrained vision models act as efficient representation learners, showcasing their utility in various downstream visual tasks, including image recognition [42, 47], object detection [33, 55] and semantic segmentation [20, 63].

The mainstream pretrained vision models usually requires a large amount of resources including data, parameters and FLOPs. These three key factors heavily influence the performance and basically follow the scaling law [61]. The large pretraining data can provide diverse samples for representation learning. The sizes of these datasets range from millions [29, 42] to billions [44, 49], for example, the widely-used ImageNet-22K dataset [42] consists of 14M images and 21,841 categories. To better fitting on the large dataset, the model sizes (including both parameters and FLOPs) are getting larger and larger in recent years, *e.g*., ViT-G/14 model has 1.8B parameters and 965B FLOPs [61].

The visual applications on mobile devices usually requires fast inference, so it is difficult to deploy the existing pretrained vision models due to the high computational cost. To address this issue, we empirically study the effect of FLOPs in large-scale visual pretraining. ImageNet-22K is adopted as the large-scale pretraining data and ImageNet-1K is a relatively small dataset for comparison. The pretrained transformer and CNN models are then finetuned on ImageNet-1K to evaluate the performance. As shown in Figure 2 and 3, when model FLOPs gradually increase, the model accuracy increases consistently. For the high-FLOPs models, 22K pretrained models outperform 1K ones. However, the low-FLOPs models cannot benifit from large-scale pretraining, and we called this observation as *low FLOPs pitfall*.

In this paper, we construct low-FLOPs ParameterNet by adding more parameters while maintaining low FLOPs for large-scale visual pretraining. It is a general design principle and there are various approaches with more parameters and low FLOPs. For instance, here we mainly consider the efficient dynamic convolution which manyfold increase the number of parameters while almost does not bring in extra FLOPs. The ParameterNet scheme can enable the previous networks to benefit from the large-scale visual pretraining and overcome the *low FLOPs pitfall*. In the experiments, the ImageNet-22K pretrained ParameterNets can improve the performance by about +2% over the regular ImageNet-1K training. For example, ParameterNet-600M achieves 81.6% top-1 accuracy on ImageNet-1K val

[*] Equal contribution.

set whose #FLOPs is 7× lower than that of Swin-T. The proposed ParameterNet can be extended to the large language model (LLM) domain and experiments on LLaMA model [54] verify the effectiveness.

The main contributions of this paper can be summarized as follows.

- We observe an interesting phenomenon in large-scale visual pretraining called the *low FLOPs pitfall*, that is, the performances of high-FLOPs models increase with more training data, but the models with low-FLOPs.
- We propose that parameters are more important than FLOPs for large-scale visual pretraining and further introduce the ParameterNet scheme by adding more parameters while maintaining low FLOPs.
- The proposed ParameterNet scheme can overcome the *low FLOPs pitfall*, and experimental results on vision and language tasks show that ParameterNet achieves significantly higher performance with large-scale pretraining.

2. Related Work

In this section, we briefly revisit the related works about visual backbone networks and visual pretraining.

Visual Backbone Networks. The deep neural networks in computer vision can be divided into CNNs, vision transformers and others. CNN used to be the mainstream network architecture for visual tasks [19, 28, 30]. The first trainable CNN *i.e.*, LeNet [30] is applied on optimal character recognition (a typical visual task). From 2012, CNNs began to be deeper and larger for more complex visual tasks, such as image classification [28], object detection [40] and semantic segmentation [35]. ResNet [19] introduces the shortcut connection to train the deeper networks and is widely used in vision and other communities. MobileNet [23] is designed for mobile devices and EfficientNet [46] scales the network from small to large.

Vision transformer is introduced into visual tasks from 2020 [4, 11, 17]. ViT [11] is the first transformer backbone by dividing the image into patches and processing them using the standard transformer architcture. Then a number of variants and improvements are proposed incluidng the pyramid architectures [33, 55], the local attentions [16, 33] and hybrid networks [13, 59].

Beyond CNNs and transformers, other types of neural networks are also explored for visual tasks. MLP-like architectures [51, 52] with only fully-connected layers as main operators can potentially simplify the software and hardware design for AI. The improved versions of MLP [5, 14, 32, 48] can enhance locality and translation equivalence. GNN has also been expored in vision and achieves competitive performance to transformers [18]. The pretrained backbone neteworks help much for the downstream visual tasks, and the study on the pretraining of backbones is an important topic.

Visual Pretraining. Large-scale pretraining has achieved great success on natural language processing such as GPT series [3, 37]. In the field of computer vision, large-scale pretraining is also beneficial and helps for downstream tasks [25, 27, 33]. The large datasets are the foundations of pretraining. To distinguish from the regular ImageNet-1K training, we consider the dataset with an order of magnitude more than ImageNet-1K (*i.e.*, more than 10M samples) as the large-scale dataset. The commonly-used large visual datasets include ImageNet-22K [42], JFT-300M [44], YFFCC100M [50] and IG-1B-Targeted [58]. The supervised pretraining is popular as it can learn semantic and meaningful representations for downstream visual tasks like segmentation and detection. BiT [27] pretrained on JFT-300M achieves state-of-the-art on fine-grained recognition datasets. The ImageNet-22K pretrained Swin Transformer [33] obtains high performance on segmentation and detection tasks. The unsupervised pretraining appeals many researchers as it does not requires labels and may leverage the massive unlabeled data. The most popular approaches of visual unsupervised pretraining are contrastive learning [6, 21] and masked image modeling (MIM) [1, 22, 57]. In this paper, we utilize the vanilla supervised pretraining for simplicity and generality.

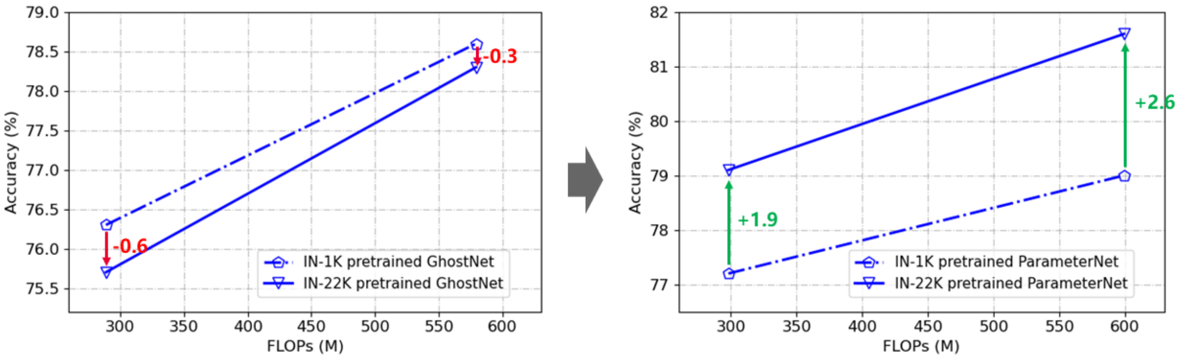
3. Low FLOPs Pitfall

The computational cost (*i.e.*, FLOPs) is an important term in the scaling of visual models. We first investigate the effect of FLOPs and observe inspiring phenomenon. Both transformer and CNN architectures are studied on ImageNet-22K and ImageNet-1K pretraining.

Transformer. Swin Transformer [33] is a representative vision transformer architecture with window attention and shifted window. We reproduce the models using the official code[1] and pretrain Swin Transformers with different scales on both ImageNet-22K and ImageNet-1K. The ImageNet-1k finetuning results are reported in Figure 2 for comparison. From the results, we can see that the accuracy increases as the FLOPs increase gradually with both ImageNet-1K and ImageNet-22K pretraining. For models with high FLOPs (>10G), pretraining on ImageNet-22K outperforms that on ImageNet-1K. However, pretraining on more data does not improve the performance for models with lower FLOPs (<4G).

CNN. For CNN, we select the widely-used EfficientNetV2 [47] which is a family of convolutional networks

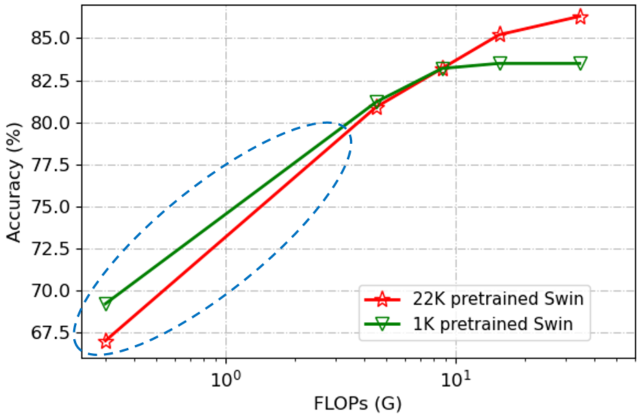
[1] https://github.com/microsoft/Swin-Transformer



(a) Original GhostNet

(b) ParameterNet

Figure 1. Results on ImageNet-1K validation set. The original GhostNet falls into the *low FLOPs pitfall*. The proposed ParameterNet overcomes the *low FLOPs pitfall*.



Model	Pretrain data	#Params	#FLOPs	Top1
Swin-300M	IN-1K	1.9M	0.3G	69.2
Swin-T	IN-1K	28M	4.5G	81.2
Swin-S	IN-1K	50M	8.7G	83.2
Swin-B	IN-1K	88M	15.4G	83.5
Swin-L	IN-1K	197M	34.5G	83.5
Swin-300M	IN-22K	1.9M	0.3G	67.0
Swin-T	IN-22K	28M	4.5G	80.9
Swin-S	IN-22K	50M	8.7G	83.2
Swin-B	IN-22K	88M	15.4G	85.2
Swin-L	IN-22K	197M	34.5G	86.3

Figure 2. Low FLOPs pitfall. Swin Transformer results on ImageNet-1K validation set. The red and blue lines denote ImageNet-22K and ImageNet-1K pretraining, respectively.

scaling from small to large. We use the official code[2] and pretrain the models on both ImageNet-22K and ImageNet-1K. From the ImageNet-1k finetuning results in Figure 3, we can observe the similar trend as that in Swin Transformer, especially, EfficientNetV2 models with less than 2G FLOPs pretraining on ImageNet-22K cannot perform better than those pretraining on ImageNet-1K.

From the observations of both transformer and CNN networks, we have a empirical conclusion that low-FLOPs models cannot benefit from large-scale pretraining, which is named as *low FLOPs pitfall*.

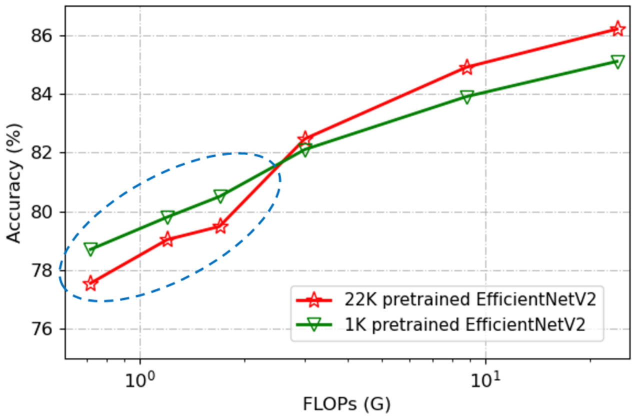
4. Approach

In this section, we investigate the low-FLOPs networks under large-scale pretraining setting.

4.1. Architecture: Transformer vs. CNN

Here we do not propose a new architecture and select the most suitable low-FLOPs network architecture for large-scale visual pretraining. ViT [11] and its variants [16, 33, 55] have shown the superiority of transformer over CNN in the field of large vision models. As shown in the appendix, Transformer-based models consistently outperform CNNs with similar computational cost when the FLOPs are higher than 5G FLOPs. As for smaller models especially mobile-level model within 600M FLOPs, CNN with inductive bias including locality and s translation equivariance remain dominant. To build efficient backbones for visual tasks, we select CNN as the base model. GhostNet [15] is the representative state-of-art mobile model which introduces cheap operation to simplify the standard convolutional layer.

[2] https://github.com/google/automl/tree/master/efficientnetv2



Model	Pretrain data	#Params	#FLOPs	Top1
EfficientNetV2-B0	IN-1K	7.1M	0.72G	78.7
EfficientNetV2-B1	IN-1K	8.1M	1.2G	79.8
EfficientNetV2-B2	IN-1K	10.1M	1.7G	80.5
EfficientNetV2-B3	IN-1K	14.4M	3.0G	82.1
EfficientNetV2-S	IN-1K	21.5M	8.4G	83.9
EfficientNetV2-M	IN-1K	54.1M	24.7G	85.2
EfficientNetV2-B0	IN-22K	7.1M	0.72G	77.6
EfficientNetV2-B1	IN-22K	8.1M	1.2G	79.0
EfficientNetV2-B2	IN-22K	10.1M	1.7G	79.5
EfficientNetV2-B3	IN-22K	14.4M	3.0G	82.5
EfficientNetV2-S	IN-22K	21.5M	8.4G	84.9
EfficientNetV2-M	IN-22K	54.1M	24.7G	86.2

Figure 3. Low FLOPs pitfall. EfficientNetV2 results on ImageNet-1K validation set. The red and blue lines denote ImageNet-22K and ImageNet-1K pretraining, respectively.

4.2. Parameters Are All You Need

The number of parameters and FLOPs are highly corelated in neural networks. The model with large number of parameters usually has high FLOPs. Considering the intuition that large data requires more parameters, we construct ParameterNet by adding parameters while maintaining low FLOPs.

We start from the conventional convolutional layer. Given the input feature $X \in \mathbb{R}^{C_{in}\times H\times W}$ and the weight tensor $W \in \mathbb{R}^{C_{out}\times C_{in}\times K\times K}$, the conventional convolutional layer operates as

$$Y = X * W, \tag{1}$$

where $Y \in \mathbb{R}^{C_{out}\times H'\times W'}$ is the output, $*$ is the convolution operation and the bias term is omitted for concision. The fully-connected layer can be viewed as the convolutional layer with 1×1 kernel size.

Our design principle is adding more parameters while maintaining low FLOPs. Thus, we introduce the parameter augmentation function which aims to introduce more parameters:

$$W' = f(W). \tag{2}$$

This function f should satisfy two basic rules: 1) it does not require much computational cost, and 2) it can largely increase the model capacity or trainable parameters. There are various approaches to construct ParameterNet, such as dynamic convolution [7] and re-parameterized convolution [10]. Although the re-parameterized convolution increase the number of parameters during training, its parameters and FLOPs are unchanged for inference, that is, the model capacity is not increased. In this paper, we mainly consider the efficient dynamic convolution (a type of MoE layer in Figure 4) which manyfold increase the number of parameters while almost does not bring in extra FLOPs.

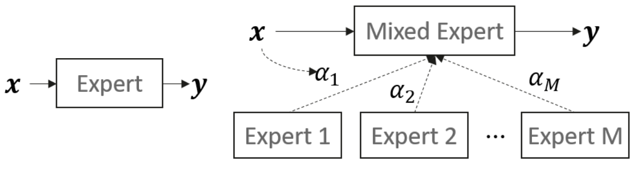
The dynamic convolution [7] with M dynamic experts can be written as

$$\begin{aligned} Y &= X * W', \\ W' &= \sum_{i=1}^{M} \alpha_i W_i. \end{aligned} \tag{3}$$

where $W_i \in \mathbb{R}^{C_{out}\times C_{in}\times H\times W}$ is the i-th convolutional weight tensor and α_i is the corresponding dynamic coefficient. The coefficient α_i is dynamically generated w.r.t. different input samples, and a typical manner is generating based on the input using MLP module. For the input X, a global average pooling is applied to fuse the information into a vector and then a two-layer MLP module with softmax activation is used to produce the coefficients dynamically:

$$\alpha = softmax(MLP(Pool(X))), \tag{4}$$

where $\alpha \in \mathbb{R}^M$. The coefficient generation in Eq. 4 only brings a nelegable FLOPs compared to the original convolutional layer. In this way, ParameterNet implemented with dynamic convolution can largely introducing much more parameters while minimizing the increase in FLOPs.



(a) Normal layer (b) MoE layer

Figure 4. The MoE layer can add more parameters while maintain low FLOPs.

Table 1. Training hyper-parameters on ImageNet datasets.

Config	ImageNet-1K	ImageNet-22K	Finetuning
Epochs	300	90	30
Optimizer	AdamW	AdamW	AdamW
Batch size	1024	4096	512
Start learning rate	1e-3	4e-3	5e-4
Layer decay	✗	✗	0.5
LR schedule	Cosine	Cosine	Cosine
Warmup epochs	20	5	0
Weight decay	0.05	0.05	1e-8
Label smoothing [45]	0.1	0.1	0.1
Stochastic path [24]	✗	✗	✗
RandAugment [9]	✓	✓	✓
Mixup [62]	✗	✗	✗
Cutmix [60]	✗	✗	✗
Random erasing [64]	0.25	0.25	✗
EMA	0.9999	✗	0.9999

Complexity Analysis. For the standard convolutional layer, the number of parameters is $C_{out} \cdot C_{in} \cdot K \cdot K$ and the number of FLOPs are $H' \cdot W' \cdot C_{out} \cdot C_{in} \cdot K \cdot K$. The dynamic convolution consists of coefficient generation module, dynamic weight fusion and convolution process. The coefficient generation module with C_{in} hidden dimensions requires $C_{in}^2 + C_{in}M$ parameters and $C_{in}^2 + C_{in}M$ FLOPs. The dynamic weight fusion is parameter-free and has $M \cdot C_{out} \cdot C_{in} \cdot K \cdot K$ FLOPs. Thus, the total numbers of parameters and FLOPs of dynamic convolution are $C_{in}^2 + C_{in}M + M \cdot C_{out} \cdot C_{in} \cdot K \cdot K$ and $C_{in}^2 + C_{in}M + M \cdot C_{out} \cdot C_{in} \cdot K \cdot K + H' \cdot W' \cdot C_{out} \cdot C_{in} \cdot K \cdot K$ respectively. The parameter ratio of dynamic convolution over standard convolution is

$$\begin{aligned} R_{param} &= \frac{C_{in}^2 + C_{in}M + MC_{out}C_{in}K^2}{C_{out}C_{in}KK} \\ &= \frac{C_{in}}{C_{out}K^2} + \frac{M}{C_{out}K^2} + M \\ &\approx \frac{1}{K^2} + M. \quad (M \ll C_{out}K^2,\ C_{in} \approx C_{out}) \end{aligned} \tag{5}$$

The FLOPs ratio is

$$\begin{aligned} R_{flops} &= \frac{C_{in}^2 + C_{in}M + MC_{out}C_{in}K^2 + H'W'C_{out}C_{in}K^2}{H'W'C_{out}C_{in}K^2} \\ &= \frac{C_{in}}{H'W'C_{out}K^2} + \frac{M}{H'W'C_{out}K^2} \\ &+ \frac{M}{H'W'} + 1 \\ &\approx 1. \quad (1 < M \ll H'W',\ C_{in} \approx C_{out}) \end{aligned} \tag{6}$$

Thus, compared to the standard convolution, the dynamic convolution has about $M\times$ parameters with negligible extra FLOPs.

4.3. Extending ParameterNet to Language Domain

Sparse-activated Mixture-of-Experts (MoE) models [43], initially introduced in the NLP domain, allow for a substantial increase in the number of parameters while maintaining the computational load per token or sample unchanged. Numerous subsequent studies [12, 41, 65] have delved into exploring efficient routing mechanisms and have demonstrated the effectiveness of MoE in various large language models (LLMs) such as T5 [38], NLLB [26], LLaMA [54] and Palm [8]. In this context, our emphasis is primarily on low-FLOPs language models to validate the proposed hypothesis that incorporating more parameters can enhance the benefits of large-scale pretraining for low-FLOPs models, *i.e.*, we proportionally reduce and construct a scaled-down version, LLaMA-1B.

Much like MoE, our approach involves taking a token representation, denoted as x, and subsequently routing it to the top-k determined experts from a set of N. The router module generates logits represented as $h(x) = softmax(router(x))$, creating a normalized distribution through a softmax function over the available N experts at that particular layer. The top-k experts, where we consistently set $k = 1$ in our experiments to maintain similar FLOPs to the original counterparts, are then selected for routing the token x. The training loss on expert capacity (the number of tokens each expert computes) follows the setting in Switch Transformer [12].

5. Experiment

In this section, we conduct experiments to verify the effectiveness of the proposed ParameterNet scheme on visual pretrianing and extend it to language domain.

5.1. Experimental Settings

Datasets. We adopt the widely-used ImageNet-22K for large-scale pretraing and ImageNet-1K as the normal training data for comparison. ImageNet-22K [42] is a large-scale image dataset with 14,197,122 images belonging to 21841 categories. ImageNet-1K [42] is a subset of ImageNet-22K with 1,000 object classes. It contains 1,281,167 training images, and 50,000 validation images.

Training on ImageNet-1K. Following the common setting [33, 34, 53], we train the models for 300 epochs with 20 warm-up epochs using AdamW [36] optimizer. We use a batch size of 1024. The base learning rate is set as 0.001 and decays with the cosine schedule. The data augmentation strategy includes RandAugment [9] and random erasing [64]. Weight decay and label smoothing are adopted for regularization. More details can be found in Table 1.

Table 2. ParameterNet results on ImageNet-1K val set by pretraining on ImageNet-1K and ImageNet-22K respectively.

Model	Pretrain data	Parameters	FLOPs	Top-1
GhostNet-300M [15]	ImageNet-1K	8.6M	289M	76.3
GhostNet-300M [15]	ImageNet-22K	8.6M	289M	75.7 (**-0.6**)
ParameterNet-300M	ImageNet-1K	15.7M	298M	77.2
ParameterNet-300M	ImageNet-22K	15.7M	298M	79.1 (**+1.9**)
GhostNet-600M [15]	ImageNet-1K	19.8M	579M	78.6
GhostNet-600M [15]	ImageNet-22K	19.8M	579M	78.3 (**-0.3**)
ParameterNet-600M	ImageNet-1K	34.5M	599M	79.0
ParameterNet-600M	ImageNet-22K	34.5M	599M	81.6 (**+2.6**)

Table 3. Comparison of ParameterNet and other SOTA models on ImageNet-1K val set. All the models are pretrained on large-scale visual datasets such as ImageNet-22K, JFT-300M and IG-1B-Targeted.

Model	Pretrain data	Parameters	FLOPs	Top-1
EfficientNet-B0 [56]	JFT-300M	5.3M	390M	78.1
Swin-300M [33]	ImageNet-22K	1.9M	312M	68.6
GhostNet-300M [15]	ImageNet-22K	8.6M	289M	75.7
ParameterNet-300M	ImageNet-22K	15.7M	298M	**79.1**
ResNet101 [44]	JFT-300M	44.5M	7.8G	79.2
ResNet50 (Billion-scale) [58]	IG-1B-Targeted	25.6M	4.1G	81.2
ResNet50 (BiT) [27]	ImageNet-22K	25.6M	12.0G	80.0
EfficientNetV2-B0 [47]	ImageNet-22K	7.1M	0.72G	77.6
EfficientNetV2-B1 [47]	ImageNet-22K	8.1M	1.2G	79.0
Swin-T [33]	ImageNet-22K	28M	4.5G	80.9
GhostNet-600M [15]	ImageNet-22K	19.8M	579M	78.3
ParameterNet-600M	ImageNet-22K	34.5M	599M	**81.6**

Pretraining on ImageNet-22K. The models are pretraining on ImageNet-22K for 90 epochs with 5 warm-up epochs. The batch size is 4096 and The base learning rate is set as 0.004. Other settings basically follow those on ImageNet-1K as shown in Table 1.

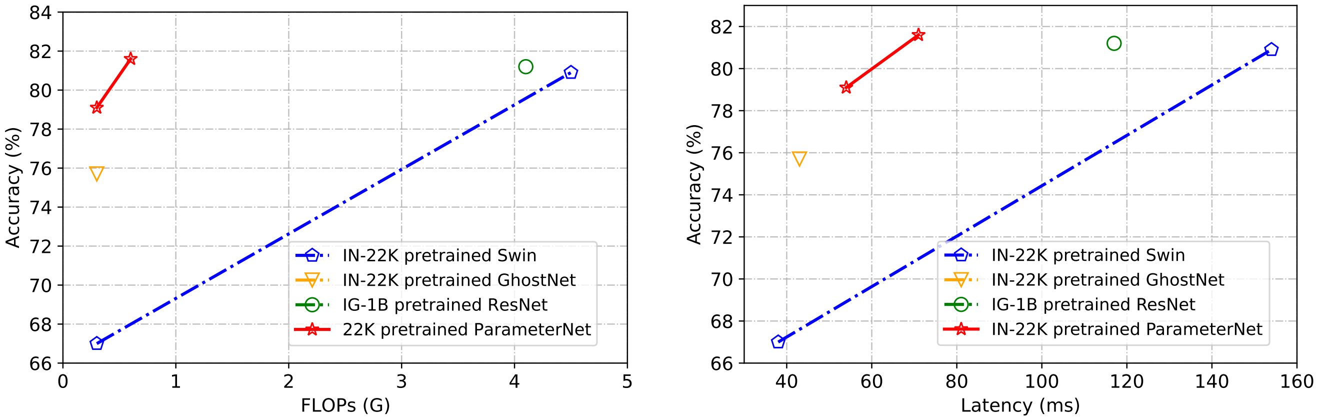
Finetuning on ImageNet-1K. We finetune the pretrained models on ImageNet-1K for 30 epochs without warm-up epochs. The batch size is 512 and The base learning rate is set as 0.0005. The weight decay is set as 1e-8 and random erasing [64] is switched off for better fitting on ImageNet-1K. Other settings basically follow those on ImageNet-1K as shown in Table 1.

5.2. Main Results on Vision Domain

We build the baseline GhostNet with different FLOPs (*i.e.*, ~300M and ~600M) by tuning the width and depth. Our ParameterNet is constructed by replacing the conventional convolutional layer with dynamic convolution. The number of experts is set as 4 by default. The details of the network architectures are available in the appendix. The results are shown in Table 2. Training only on ImageNet-1K, ParameterNet outperforms the original GhostNet by 0.4-xx accuracy. For GhostNet, pretraining on ImageNet-22K does not help to the performance. ImageNet-22K pretrained ParameterNet can achieve more than 2% improvement over ImageNet-1K. This indicates that our ParameterNet with more parameters yet similar FLOPs can benefit from the large-scale visual pretraining.

Comparison with SOTA. We compare ParameterNet with other representative models pretrained on ImageNet-22K or larger datasets such as JFT-300M [44] and IG-1B-Targeted [58]. From the results in Table 3, we can see that our ParameterNet with fewer FLOPs outperforms other models pretrained on large-scale datasets. For example, ParameterNet-600M achieves 81.6% top-1 accuracy whose #FLOPs is about 7× lower than that of ResNet50 or Swin-T.

Inference speed. We evaluate the inference speed of ParameterNet and other representative models for comparison. We run models using ONNX toolkit on Intel Xeon



(a) Acc v.s. FLOPs

(b) Acc v.s. Latency

Figure 5. Performance comparison of the representative visual backbone networks with ImageNet-22K pretraining.

Platinum 8378C CPU with single-thread mode. As shown in Figure 5, our ParameterNet outperforms the widely-used ResNet and Swin Transformer for much better accuracy-latency trade-off.

Table 4. ImageNet-1K val set results w.r.t. #Expert. The base network architecture is GhostNet-300M.

#Expert	Pretrain data	Parameters	FLOPs	Top-1
1	ImageNet-1K	8.6M	289M	76.3
1	ImageNet-22K	8.6M	289M	75.7
2	ImageNet-1K	11.0M	293M	76.9
2	ImageNet-22K	11.0M	293M	77.7
4	ImageNet-1K	15.7M	298M	77.2
4	ImageNet-22K	15.7M	298M	79.1
8	ImageNet-1K	25.2M	308M	77.7
8	ImageNet-22K	25.2M	308M	79.4

5.3. Ablation Study

The number of dynamic experts. The number of dynamic experts is an important hyperparameter of dynamic convolution, which directly controls the parameters and FLOPs. As shown in Table 4, more experts will largely increase the number of parameters and slightly influence FLOPs. The performance of more experts improves over fewer experts. We use 4 experts by default for efficiency trade-off.

Table 5. Results on ImageNet-1K val set. The base network architecture is Swin-300M.

	Pretrain	Parameters	FLOPs	Top-1
Original	ImageNet-1K	1.9M	312M	69.2
Original	ImageNet-22K	1.9M	312M	67.0 (-2.2)
Ours	ImageNet-1K	6.8M	323M	72.3
Ours	ImageNet-22K	6.8M	323M	74.5 (+2.2)

Dynamic convolution *vs.* re-parameterized convolution. As we discussed before, there are various approaches to construct ParameterNet, such as dynamic convolution [7] and re-parameterized convolution [10]. We compare these two approaches where the dynamic convolution has 4 experts and the re-parameterized convolution has 3 more paralleled branches based on the original convolution. From the results in Table 6, although the re-parameterized convolution increase the training parameters, its parameters and FLOPs are unchanged for inference, that is, the model capacity is not increased and the ImageNet-22K pretrained performance does not improve.

ParameterNet for other network architectures. In addition to CNN, we extend ParameterNet to the transformer architecture (*i.e.*, Swin Transformer). To construct a smaller version, we set the token dimension of Swin-T to 24 to obtain Swin-300M with about 300M FLOPs. From the results in Table 5, the original Swin-300M has a significant accuracy drop when pretraining on ImageNet-22K. Our strategy can achieve +2.2% performance gain from ImageNet-22K pretraining.

5.4. Extensive Experiment on Language Domain

Datasets. Our training dataset is a mixture of several sources, including C4 [39], Wikipedia [54], and ArXiv [31]. The data are all publicly available, and we directly mix them without any quality filtering. Overall, the training dataset

Table 6. Comparison of different approaches to construct ParameterNet on ImageNet-1K val set. The base network architecture is GhostNet-300M.

Method	Pretrain data	Train parameters	Inference parameters	FLOPs	Top-1
RepConv	ImageNet-1K	15.7M	8.6M	289M	76.8
RepConv	ImageNet-22K	15.7M	8.6M	289M	76.9
DynamicConv	ImageNet-1K	15.7M	15.7M	298M	77.2
DynamicConv	ImageNet-22K	15.7M	15.7M	298M	79.1

Table 7. ParameterNet (sparse-activated MoE) on LLaMA-1B. Given the SwiGLU activation function in LLaMA, there are three linear projections in the FFN module. We added parameters at each linear projection, and we present the corresponding zero-shot results, except for SST-2 (where we fine-tuned the classifier). The best results are in **bold** and the second best are underlined.

Model	#Expert	Parameters	FLOPs	Training loss	ARC (easy)	BoolQ	SST-2	HellaSwag	Avg
LLaMA-1B	-	0.94B	919G	1.86	47.99	57.31	88.53	38.92	58.19
MoE on gate	4	1.54B	919G	1.75	48.81	57.86	88.88	42.09	59.41
MoE on gate	8	2.35B	919G	1.64	49.04	57.70	89.56	43.78	60.02
MoE on up proj	4	1.54B	919G	1.72	49.09	58.04	89.67	42.35	59.79
MoE on up proj	8	2.35B	919G	1.61	**49.58**	58.39	90.05	**44.21**	**60.56**
MoE on down proj	4	1.54B	919G	1.70	49.36	57.99	89.13	41.95	59.61
MoE on down proj	8	2.35B	919G	1.62	49.22	**58.58**	**90.34**	43.60	60.44

contains roughly 90B tokens after tokenization. Each token is used only once during training. The learning rate is set to 0.0003 with a batch size of 4M (input token length is 2048). We use the AdamW optimizer with a cosine learning rate schedule, ensuring that the final learning rate is equal to 10% of the maximal learning rate.

Network architecture. We build a baseline LLaMA-1B by proportionally reduce the dimension and the number of layers based on original LLaMA [54], as shown in Table 8. Specifically, the hidden size, intermediate size, number of head, and the number of layer are 2048, 8191, 16 and 12, respectively. The tokenizer is the same with LLaMA.

Table 8. Network architecture of LLaMA-1B baseline. In the following experiments, we equipe the fully-connected layer in FFN with MoE to verify the proposed method.

Model	Dimension	Heads	Layers	Parameters	FLOPs
LLaMA-1B	2048	16	12	0.94B	919G

Results and analysis. Following previous work [2], we present the corresponding training loss and zero-shot results on several common-sense reasoning tasks, where the model ranks the proposed answers. FLOPs are calculated with the output response length set to 1. The router module is implemented with a linear layer, with the input channel being the hidden size and the output channel equal to the number of experts. As shown in Table 7, we observe that more experts bring additional parameters to the baseline model, leading to a noticeable improvement in downstream performance. For example, LLaMA-1B with 8 experts on up projection layers obtains a 2.37% accuracy gain on average. Moreover, the increased parameters help reduce the training loss, indicating enhanced understanding of the input data by incorporating ParameterNet into the language model. Additionally, experimental results suggest that the three linear projections in LLaMA's FFN have similar effects.

6. Conclusion

In this paper, we propose a design principle (*i.e.*, ParameterNet) for large-scale visual pretraining by adding more parameters while maintaining low FLOPs. ParameterNet is a general scheme and has various approaches to implement such as dynamic convolution and re-parameterized convolution. We use the dynamic convolution in practice to construct the ParameterNet models. ParameterNet can overcome the *low FLOPs pitfall* and much benefit from large-scale visual pretraining. The experiments on ImageNet-22K large-scale dataset have demonstrated the effectiveness of the proposed ParameterNet. We also verify the generalization of our method on language domain. We hope our work can motivate and inspire the future research on vision, multimodality and large language models.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 2

[2] 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. *Advances in neural information processing systems*, 33:1877–1901, 2020. 8

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 2

[5] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *ICLR*, 2022. 2

[6] 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

[7] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *CVPR*, pages 11030–11039, 2020. 4, 7

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 5

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020. 5

[10] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 4, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[12] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 2022. 5

[13] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, pages 12175–12185, 2022. 2

[14] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. In *CVPR*, 2022. 2

[15] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *CVPR*, 2020. 3, 6

[16] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 2, 3

[17] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[18] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. In *NeurIPS*, 2022. 2

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

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[24] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 5

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2

[26] Yeskendir Koishekenov, Vassilina Nikoulina, and Alexandre Berard. Memory-efficient nllb-200: Language-specific expert pruning of a massively multilingual machine translation model. *arXiv preprint arXiv:2212.09811*, 2022. 5

[27] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020. 2, 6

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 2

[29] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The

open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 1

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[31] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 2022. 7

[32] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. In *ICLR*, 2022. 2

[33] 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 *ICCV*, pages 10012–10022, 2021. 1, 2, 3, 5, 6

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 5

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

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

[37] OpenAI. Gpt-4 technical report, 2023. 2

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 5

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020. 7

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 2

[41] Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems*, 2021. 5

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 2, 5

[43] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 5

[44] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, pages 843–852, 2017. 1, 2, 6

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 5

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[47] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *ICML*, pages 10096–10106. PMLR, 2021. 1, 2, 6

[48] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. In *CVPR*, 2022. 2

[49] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 1(8), 2015. 1

[50] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 2

[51] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021. 2

[52] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 2

[53] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 5

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 5, 7, 8

[55] 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*, 2021. 1, 2, 3

[56] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020. 6

[57] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2

[58] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 2, 6

[59] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, 2022. 2

[60] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 5

[61] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 12104–12113, 2022. 1

[62] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 5

[63] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 1

[64] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020. 5, 6

[65] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 2022. 5