InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions

Wenhai Wang[1*], Jifeng Dai[2,1*], Zhe Chen[3,1*], Zhenhang Huang[1*], Zhiqi Li[3,1*], Xizhou Zhu[4*], Xiaowei Hu[1], Tong Lu[3], Lewei Lu[4], Hongsheng Li[5], Xiaogang Wang[4,5], Yu Qiao[1✉]
[1] Shanghai AI Laboratory [2]Tsinghua University
[3]Nanjing University [4]SenseTime Research [5]The Chinese University of Hong Kong
https://github.com/OpenGVLab/InternImage

Abstract

Compared to the great progress of large-scale vision transformers (ViTs) in recent years, large-scale models based on convolutional neural networks (CNNs) are still in an early state. This work presents a new large-scale CNN-based foundation model, termed InternImage, which can obtain the gain from increasing parameters and training data like ViTs. Different from the recent CNNs that focus on large dense kernels, InternImage takes deformable convolution as the core operator, so that our model not only has the large effective receptive field required for downstream tasks such as detection and segmentation, but also has the adaptive spatial aggregation conditioned by input and task information. As a result, the proposed InternImage reduces the strict inductive bias of traditional CNNs and makes it possible to learn stronger and more robust patterns with large-scale parameters from massive data like ViTs. The effectiveness of our model is proven on challenging benchmarks including ImageNet, COCO, and ADE20K. It is worth mentioning that InternImage-H achieved a new record 65.4 mAP on COCO test-dev and 62.9 mIoU on ADE20K, outperforming current leading CNNs and ViTs.

1. Introduction

With the remarkable success of transformers in large-scale language models [3–8], vision transformers (ViTs) [2, 9–15] have also swept the computer vision field and are becoming the primary choice for the research and practice of large-scale vision foundation models. Some pioneers [16–20] have made attempts to extend ViTs to very large models with over a billion parameters, beating convolutional neural networks (CNNs) and significantly pushing the performance bound for a wide range of computer vision tasks, including basic classification, detection, and segmentation. While these results suggest that CNNs are inferior to ViTs in the era of massive parameters and data, we argue that *CNN-based foundation models can also achieve comparable or even better performance than ViTs when*

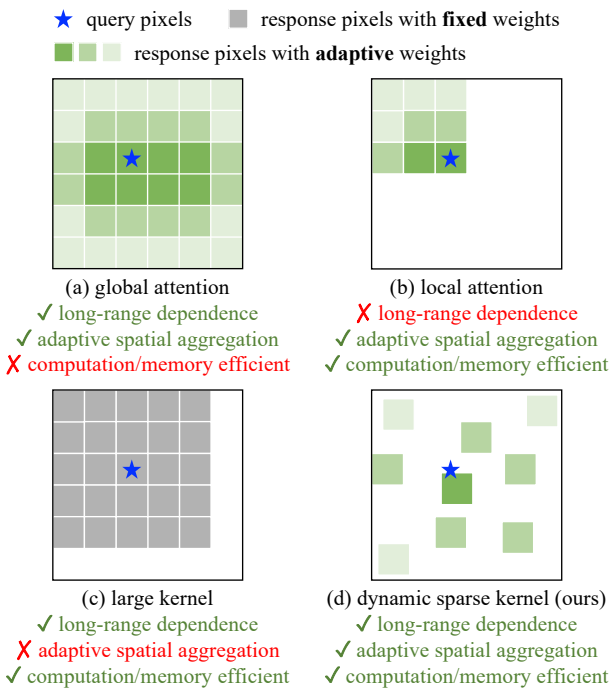

Figure 1. **Comparisons of different core operators.** (a) shows the global aggregation of multi-head self-attention (MHSA) [1], whose computational and memory costs are expensive in downstream tasks that require high-resolution inputs. (b) limits the range of MHSA into a local window [2] to reduce the cost. (c) is a depth-wise convolution with very large kernels to model long-range dependencies. (d) is a deformable convolution, which shares similar favorable properties with MHSA and is efficient enough for large-scale models. We start from it to build a large-scale CNN.

* equal contribution, ✉ corresponding author (qiaoyu@pjlab.org.cn)

equipped with similar operator-/architecture-level designs, scaling-up parameters, and massive data.

To bridge the gap between CNNs and ViTs, we first summarize their differences from two aspects: (1) From the operator level [9, 21, 22], the multi-head self-attention (MHSA) of ViTs has long-range dependencies and adaptive spatial aggregation (see Fig. 1(a)). Benefiting from the flexible MHSA, ViTs can learn more powerful and robust representations than CNNs from massive data. (2) From the architecture view [9, 22, 23], besides MHSA, ViTs contain a series of advanced components that are not included in standard CNNs, such as Layer Normalization (LN) [24], feed-forward network (FFN) [1], GELU [25], etc. Although recent works [21, 22] have made meaningful attempts to introduce long-range dependencies into CNNs by using dense convolutions with very large kernels (*e.g.*, 31×31) as shown in Fig. 1 (c), there is still a considerable gap with the state-of-the-art large-scale ViTs [16, 18–20, 26] in terms of performance and model scale.

In this work, we concentrate on designing a CNN-based foundation model that can efficiently extend to large-scale parameters and data. Specifically, we start with a flexible convolution variant—deformable convolution (DCN) [27, 28]. By combining it with a series of tailored block-level and architecture-level designs similar to transformers, we design a brand-new convolutional backbone network, termed *InternImage*. As shown in Fig. 1, different from recently improved CNNs with very large kernels such as 31×31 [22], the core operator of InternImage is a dynamic sparse convolution with a common window size of 3×3, (1) whose sampling offsets are flexible to dynamically learn appropriate receptive fields (can be long- or short-range) from given data; (2) the sampling offsets and modulation scalars are adaptively adjusted according to the input data, which can achieve adaptive spatial aggregation like ViTs, reducing the over-inductive bias of regular convolutions; and (3) the convolution window is a common 3×3, avoiding the optimization problems and expensive costs caused by large dense kernels [22, 29].

With the aforementioned designs, the proposed InternImage can efficiently scale to large parameter sizes and learn stronger representations from large-scale training data, achieving comparable or even better performance to large-scale ViTs [2, 11, 30] on a wide range of vision tasks. In summary, our main contributions are as follows:

(1) We present a new large-scale CNN-based foundation model—InternImage. To our best knowledge, it is the first CNN that effectively scales to over 1 billion parameters and 400 million training images and achieves comparable or even better performance than state-of-the-art ViTs, showing that convolutional models are also a worth-exploring direction for large-scale model research.

(2) We successfully scale CNNs to large-scale settings by introducing long-range dependencies and adaptive spatial aggregation using an improved 3×3 DCN operator, and explore the tailored basic block, stacking rules, and scaling strategies centered on the operator. These designs make effective use of the operator, enabling our models to obtain the gains from large-scale parameters and data.

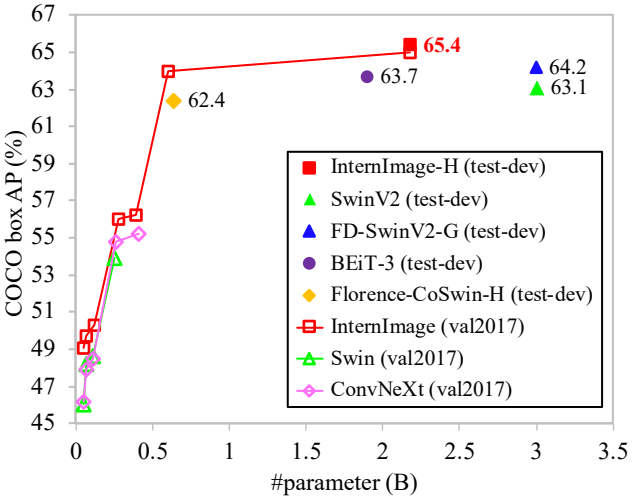

Figure 2. **Performance comparison on COCO of different backbones.** The proposed InternImage-H achieves a new record 65.4 box AP on COCO test-dev, significantly outperforming state-of-the-art CNNs and large-scale ViTs.

(3) We evaluate the proposed model on representative vision tasks including image classification, object detection, instance and semantic segmentation, and compared it with state-of-the-art CNNs and large-scale ViTs by scaling the model size ranging from 30 million to 1 billion, the data ranging from 1 million to 400 million. Specifically, our model with different parameter sizes can consistently outperform prior arts on ImageNet [31]. InternImage-B achieves 84.9% top-1 accuracy trained only on the ImageNet-1K dataset, outperforming CNN-based counterparts [21, 22] by at least 1.1 points. With large-scale parameters (*i.e.*, 1 billion) and training data (*i.e.*, 427 million), the top-1 accuracy of InternImage-H is further boosted to 89.6%, which is close to well-engineering ViTs [2, 30] and hybrid-ViTs [20]. In addition, on COCO [32], a challenging downstream benchmark, our best model InternImage-H achieves state-of-the-art 65.4% box mAP with 2.18 billion parameters, 2.3 points higher than SwinV2-G [16] (65.4 *vs.* 63.1) with 27% fewer parameters as shown in Fig. 2.

2. Related Work

Vision foundation models. Convolutional neural networks (CNNs) became the mainstream for visual recognition after the large-scale dataset and computation resources were available. Straining from AlexNet [33], lots of deeper

and more effective neural network architectures have been proposed, such as VGG [34], GoogleNet [35], ResNet [36], ResNeXt [37], EfficientNet [38, 39], etc. In addition to the architectural design, more sophisticated convolution operations such as depth-wise convolution [40] and deformable convolution [27, 28] are formulated. By considering the advanced designs of transformers, modern CNNs showed promising performance on the vision tasks by discovering better components in macro/micro designs and introducing improved convolutions with long-range dependencies [21, 41–43] or dynamic weights [44].

In recent years, a new line of vision foundation models focuses on transformer-based architecture. ViT [9] is the most representative model, which achieves great success in vision tasks thanks to global receptive fields and dynamic spatial aggregation. However, global attention in ViT suffers from expensive computational/memory complexity, especially on large feature maps, which limits its application in downstream tasks. To address this problem, PVT [10, 11] and Linformer [45] performed global attention on the downsampled key and value maps, DAT [46] employed deformable attention to sparsely sample information from value maps, while HaloNet [47] and Swin transformer [2] developed local attention mechanisms and used haloing and shift operations to transfer information among adjacent local regions.

Large-scale models. Scaling up models is an important strategy to improve feature representation quality, which has been well-studied in the natural language processing (NLP) domain [48]. Inspired by the success in the NLP field, Zhai *et al.* [19] first extended ViT to 2 billion parameters. Liu *et al.* [16] enlarged the hierarchical-structure Swin transformer to a deeper and wider model with 3 billion parameters. Some researchers developed large-scale hybrid ViTs [20, 49] by combining the advantages of ViTs and CNNs at different levels. Recently, BEiT-3 [17] further explored stronger representations based on ViT with large-scale parameters using multimodal pre-training. These methods significantly raise the upper bound of basic vision tasks. However, research on CNN-based large-scale models has lagged behind transformer-based architectures in terms of the total number of parameters and performance. Although newly-proposed CNNs [21, 41–43] introduce long-range dependencies by using convolutions with very large kernels or recursive gated kernels, there is still a considerable gap with state-of-the-art ViTs. In this work, *we aim to develop a CNN-based foundation model that can extend efficiently to a large scale comparable to ViT.*

3. Proposed Method

To design a large-scale CNN-based foundation model, we start with a flexible convolution variant, namely deformable convolution v2 (DCNv2) [28] and make some tune-ups based on it to better suit the requirements of large-scale foundation models. Then, we build the basic block by combining the tuned convolution operator with advanced block designs used in modern backbones [16, 19]. Finally, we explore the stacking and scaling principles of DCN-based blocks to build a large-scale convolutional model that can learn strong representations from massive data.

3.1. Deformable Convolution v3

Convolution *vs*. MHSA. Previous works [21, 22, 50] have extensively discussed the differences between CNNs and ViTs. Before deciding on the core operator of InternImage, we first summarize the main differences between regular convolution and MHSA.

(1) *Long-range dependencies*. Although it has long been recognized that models with large effective receptive fields (long-range dependencies) usually perform better on downstream vision tasks [51–53], the de-facto effective receptive field of CNNs [34, 36] stacked by 3×3 regular convolutions is relatively small. Even with very deep models, the CNN-based model still cannot acquire long-range dependencies like ViTs, which limits its performance.

(2) *Adaptive spatial aggregation*. Compared to MHSA whose weights are dynamically conditioned by the input, regular convolution [54] is an operator with static weights and strong inductive biases such as 2D locality, neighborhood structure, translation equivalence, etc. With the highly-inductive properties, models composed by regular convolutions might converge faster and require less training data than ViTs, but it also restricts CNNs from learning more general and robust patterns from web-scale data.

Revisiting DCNv2. A straightforward way to bridge the gap between convolution and MHSA is to introduce long-range dependencies and adaptive spatial aggregation into regular convolutions. Let us start with DCNv2 [28], which is a general variant of regular convolution. Given an input $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ and current pixel p_0, DCNv2 can be formulated as:

$$\mathbf{y}(p_0) = \sum_{k=1}^{K} \mathbf{w}_k \mathbf{m}_k \mathbf{x}(p_0 + p_k + \Delta p_k), \tag{1}$$

where K represents the total number of sampling points, and k enumerates the sampling point. $\mathbf{w}_k \in \mathbb{R}^{C \times C}$ denotes the projection weights of the k-th sampling point, and $\mathbf{m}_k \in \mathbb{R}$ represents the modulation scalar of the k-th sampling point, which is normalized by sigmoid function. p_k denotes the k-th location of the pre-defined grid sampling $\{(-1,-1), (-1,0), ..., (0,+1), ..., (+1,+1)\}$ as in regular convolutions, and Δp_k is the offset corresponding to the k-th grid sampling location. We see from the equation that (1) for long-range dependencies, the sampling offset Δp_k is flexible and able to interact with short- or

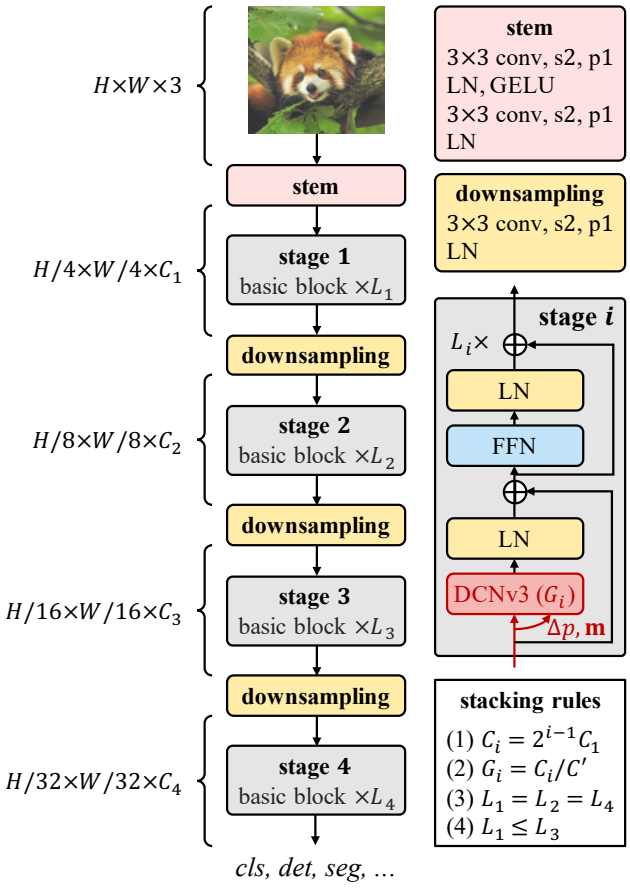

Figure 3. **Overall Architecture of InternImage**, where the core operator is DCNv3, and the basic block composes of layer normalization (LN) [24] and feed-forward network (FFN) [1] as transformers, the stem and downsampling layers follows conventional CNN's designs, where "s2" and "p1" mean stride 2 and padding 1, respectively. Constrained by the stacking rules, only 4 hyper-parameters (C_1, C', L_1, L_3) can decide a model variant.

long-range features; and (2) for adaptive spatial aggregation, both the sampling offset Δp_k and modulation scalar $\mathbf{m}_k$ are learnable and conditioned by input $\mathbf{x}$. So it can be found that *DCNv2 shares similar favorable properties with MHSA*, which motivated us to develop large-scale CNN-based foundation models on the basis of this operator.

Extending DCNv2 for Vision Foundation Models. In common practice, DCNv2 is usually used as an extension to regular convolutions, loading pre-trained weights of regular convolutions and fine-tuning for better performance, which is not exactly suitable for large-scale vision foundation models that need to be trained from scratch. In this work, to address this problem, we extend DCNv2 from aspects as follows:

(1) *Sharing weights among convolutional neurons.* Similar to regular convolution, different convolutional neurons[1] in original DCNv2 have independent linear projection weights, and thus its parameter and memory complexity are linear with the total number of sampling points, which significantly limits the efficiency of the model, especially in large-scale models. To remedy this problem, we borrow the idea from the separable convolution [55] and detach the original convolution weights $\mathbf{w}_k$ into depth-wise and point-wise parts, where the depth-wise part is responsible by the original location-aware modulation scalar $\mathbf{m}_k$, and the point-wise part is the shared projection weights $\mathbf{w}$ among sampling points.

(2) *Introducing multi-group mechanism.* The multi-group (head) design first appeared in group convolution [33], and it is widely used in MHSA [1] of transformers and works with adaptive spatial aggregation to effectively learn richer information from different representation subspaces at different locations. Inspired by this, we split the spatial aggregation process into G groups, each of which has individual sampling offsets Δp_{gk} and modulation scale $\mathbf{m}_{gk}$, and thus different groups on a single convolution layer can have different spatial aggregation patterns, resulting in stronger features for downstream tasks.

(3) *Normalizing modulation scalars along sampling points.* The modulation scalars in the original DCNv2 are element-wise normalized by the sigmoid function. Therefore, each modulation scalar is in the range [0, 1], and the sum of the modulation scalars of all sample points is not stable and varies from 0 to K. This leads to unstable gradients in DCNv2 layers when training with large-scale parameters and data. To alleviate the instability issues, we change element-wise sigmoid normalization to softmax normalization along sample points. In this way, the sum of the modulation scalars is constrained to 1, which makes the training process of models at different scales more stable.

Combining the aforementioned modifications, the extended DCNv2, marked as DCNv3, can be formulated as Eqn. (2).

$$\mathbf{y}(p_0) = \sum_{g=1}^{G} \sum_{k=1}^{K} \mathbf{w}_g \mathbf{m}_{gk} \mathbf{x}_g(p_0 + p_k + \Delta p_{gk}), \quad (2)$$

where G denotes the total number of aggregation groups. For the g-th group, $\mathbf{w}_g \in \mathbb{R}^{C \times C'}$ denotes the location-irrelevant projection weights of the group, where $C' = C/G$ represents the group dimension. $\mathbf{m}_{gk} \in \mathbb{R}$ denotes the modulation scalar of the k-th sampling point in the g-th group, normalized by the softmax function along the dimension K. $\mathbf{x}_g \in \mathbb{R}^{C' \times H \times W}$ represents the sliced input feature map. Δp_{gk} is the offset corresponding to the grid sampling location p_k in the g-th group.

In general, DCNv3, as an extension of the DCN series, enjoys three merits as follows: (1) This operator made up

[1] A 3×3 regular convolution has 9 linear projection neurons.

for the deficiencies of regular convolution in terms of long-range dependencies and adaptive spatial aggregation; (2) Compared with attention-based operators such as common MHSA and closely-related deformable attention [46, 56], this operator inherits the inductive bias of convolution, making our model more efficient with fewer training data and shorter training time; (3) This operator is based on sparse sampling, which is more computational and memory efficient than previous methods such as MHSA [1] and re-parameterizing large kernel [22]. In addition, due to the sparse sampling, DCNv3 only needs a 3×3 kernel to learn long-range dependencies, which is easier to be optimized and avoids extra auxiliary techniques such as re-parameterizing [22] used in large kernels.

3.2. InternImage Model

Using DCNv3 as the core operator brings a new problem: *how to build a model that can make effective use of the core operator?* In this section, we first present the details of the basic block and other integral layers of our model, and then we construct a new CNN-based foundation model termed InternImage, by exploring a tailored stacking strategy for these basic blocks. Finally, we study scaling-up rules for the proposed model to obtain the gain from increasing parameters.

Basic block. Unlike the widely used bottlenecks in traditional CNNs [36], the design of our basic block is closer to ViTs, which is equipped with more advanced components including LN [24], feed-forward networks (FFN) [1], and GELU [25]. This design is proved to be efficient [2, 10, 11, 21, 22] in various vision tasks. The details of our basic block are illustrated in Fig. 3, where the core operator is DCNv3, and the sampling offsets and modulation scales are predicted by passing input feature $\mathbf{x}$ through a separable convolution (a 3×3 depth-wise convolution followed by a linear projection). For other components, we use the post-normalization setting [57] by default and follow the same design as that of the plain transformer [1, 9].

Stem & downsampling layers. To obtain hierarchical feature maps, we use convolutional stem and downsampling layers to resize the feature maps to different scales. As shown in Fig. 3, the stem layer is placed before the first stage to reduce the input resolution by 4 times. It consists of two convolutions, two LN layers, and one GELU layer, where the kernel size of the two convolutions is 3, the stride is 2, the padding is 1, and the output channel of the first convolution is half of the second one. Similarly, the downsampling layer is made up of a 3×3 convolution with a stride of 2 and a padding of 1, followed by one LN layer. It sits between the two stages and is used to downsample the input feature map by 2 times.

Stacking rules. To clarify the block-stacking process, we first list the integral hyperparameters of the InternImage as follows:

C_i: the channel number of the i-th stage;
G_i: the group number of the DCNv3 in the i-th stage;
L_i: the number of basic blocks in the i-th stage.

model name	C_1	C'	$L_{1,2,3,4}$	#params
InternImage-T (origin)	64	16	4, 4, 18, 4	30M
InternImage-S	80	16	4, 4, 21, 4	50M
InternImage-B	112	16	4, 4, 21, 4	97M
InternImage-L	160	16	5, 5, 22, 5	223M
InternImage-XL	192	16	5, 5, 24, 5	335M
InternImage-H	320	32	6, 6, 32, 6	1.08B

Table 1. **Hyper-parameters for models of different scales**. InternImage-T is the origin model, and -S/B/L/XL/H are scaled up from -T. "#params" denotes the number of parameters.

Since our model has 4 stages, a variant is decided by 12 hyper-parameters, whose search space is too large to exhaustively enumerate and find the best variant. To reduce the search space, we summarize the design experiences of prior arts [2, 21, 36] into 4 rules as shown in Fig. 3, where the first rule makes the channel numbers of the last three stages determined by the channel number C_1 of the first stage, and the second rule lets the group number correspond to the channel number of stages. For the number of stacked blocks in different stages, we simplify the stacking pattern to "AABA", which means the block number of stage 1, 2, and 4 are the same, and are not greater than that of the stage 3 as illustrated in the last two rules. With these rules, a InternImage variant can be defined by using only 4 hyper-parameters (C_1, C', L_1, L_3).

Let us choose a model with 30 million parameters as the origin and discretize C_1 to $\{48, 64, 80\}$, L_1 to $\{1, 2, 3, 4, 5\}$, and C' to $\{16, 32\}$. In this way, the original huge search space is reduced to 30, and we can find the best model from the 30 variants by training and evaluating them in ImageNet [31]. In practice, we use the best hyper-parameter setting $(64, 16, 4, 18)$ to define the origin model and scale it to different scales.

Scaling rules. Based on the optimal origin model under the aforementioned constraints, we further explore the parameter scaling rules inspired by [38]. Specifically, we consider two scaling dimensions: depth D (*i.e.*, $3L_1 + L_3$) and width C_1, and scale the two dimensions using α, β and a composite factor ϕ. The scaling rules can be written as: $D' = \alpha^\phi D$ and $C_1' = \beta^\phi C_1$, where $\alpha \geq 1$, $\beta \geq 1$, and $\alpha\beta^{1.99} \approx 2$. Here, 1.99 is specific for InternImage and calculated by doubling the model width and keeping the depth constant. We experimentally find out that the best scaling setting is $\alpha = 1.09$ and $\beta = 1.36$, and then we base on it to construct InternImage variants with different parameter scales, namely InternImage-T/S/B/L/XL, whose complexity is similar to those of ConvNeXt [21]. To further test the capability, we built a larger InternImage-H with 1 billion

parameters, and to accommodate very large model widths, we also change the group dimension C' to 32. The configurations are summarized in Table 1.

4. Experiment

We analyze and compare InternImage with the leading CNNs and ViTs on representative vision tasks including image classification, object detection, instance and semantic segmentation. Besides the experiments in the main paper, due to space constraints, more experimental setups and ablation studies are presented in the supplementary material.

4.1. Image Classification

Settings. We evaluate the classification performance of InternImage on ImageNet [31]. For fair comparisons, following common practices [2, 10, 21, 58], InternImage-T/S/B are trained on ImageNet-1K ($\sim$1.3 million) for 300 epochs, and InternImage-L/XL are first trained on ImageNet-22K ($\sim$14.2 million) for 90 epochs and then fine-tuned on ImageNet-1K for 20 epochs. To further explore the capability of our model and match the large-scale private data used in previous methods [16, 20, 59], we adopt M3I Pre-training [60], a unified pre-training approach available for both unlabeled and weakly-labeled data, to pre-train InternImage-H on a 427 million joint dataset of public Laion-400M [61], YFCC-15M [62], and CC12M [63] for 30 epochs, and then we fine-tune the model on ImageNet-1K for 20 epochs.

Results. Table 2 shows the classification results of models with different scales. With similar parameters and computational costs, our models are comparable or even superior to the state-of-the-art transformer-based and CNN-based models. For example, InternImage-T achieves 83.5% top-1 accuracy, outperforming ConvNext-T [21] with a clear margin of 1.4 points. InternImage-S/B keeps the leading position and InternImage-B surpasses the hybrid-ViT CoAtNet-2 [20] by 0.8 points. When pre-trained on ImageNet-22K and the large-scale joint dataset, the top-1 accuracy of InternImage-XL and -H are boosted to 88.0% and 89.6%, respectively, which is better than previous CNNs [22, 67] also trained with large-scale data, and closes the gap with the state-of-the-art large-scale ViTs to about 1 point. This gap may be caused by the discrepancy between large-scale inaccessible private data and the aforementioned joint public data. These results show that our InternImage not only has good performance on the common parameter scale and the public training data, but also can effectively extend to large-scale parameters and data.

4.2. Object Detection

Settings. We verify the detection performance of our InternImage on the COCO benchmark [32], on top of

method	type	scale	#params	#FLOPs	acc (%)
DeiT-S [58]	T	224^2	22M	5G	79.9
PVT-S [10]	T	224^2	25M	4G	79.8
Swin-T [2]	T	224^2	29M	5G	81.3
CoAtNet-0 [20]	T	224^2	25M	4G	81.6
CSwin-T [12]	T	224^2	23M	4G	82.7
PVTv2-B2 [11]	T	224^2	25M	4G	82.0
DeiT III-S [64]	T	224^2	22M	5G	81.4
SwinV2-T/8 [16]	T	256^2	28M	6G	81.8
Focal-T [65]	T	224^2	29M	5G	82.2
ConvNeXt-T [21]	C	224^2	29M	5G	82.1
ConvNeXt-T-dcls [66]	C	224^2	29M	5G	82.5
SLaK-T [29]	C	224^2	30M	5G	82.5
HorNet-T [43]	C	224^2	23M	4G	83.0
InternImage-T (ours)	C	224^2	30M	5G	83.5
PVT-L [10]	T	224^2	61M	10G	81.7
Swin-S [2]	T	224^2	50M	9G	83.0
CoAtNet-1 [20]	T	224^2	42M	8G	83.3
PVTv2-B4 [11]	T	224^2	63M	10G	83.6
SwinV2-S/8 [16]	T	256^2	50M	12G	83.7
ConvNeXt-S [21]	C	224^2	50M	9G	83.1
ConvNeXt-S-dcls [66]	C	224^2	50M	10G	83.7
SLaK-S [29]	C	224^2	55M	10G	83.8
HorNet-S [43]	C	224^2	50M	9G	84.0
InternImage-S (ours)	C	224^2	50M	8G	84.2
DeiT-B [58]	T	224^2	87M	18G	83.1
Swin-B [2]	T	224^2	88M	15G	83.5
CoAtNet-2 [20]	T	224^2	75M	16G	84.1
PVTv2-B5 [11]	T	224^2	82M	12G	83.8
DeiT III-B [64]	T	224^2	87M	18G	83.8
SwinV2-B/8 [16]	T	256^2	88M	20G	84.2
RepLKNet-31B [22]	C	224^2	79M	15G	83.5
ConvNeXt-B [21]	C	224^2	88M	15G	83.8
ConvNeXt-B-dcls [66]	C	224^2	89M	17G	84.1
SLaK-B [29]	C	224^2	95M	17G	84.0
HorNet-B [43]	C	224^2	88M	16G	84.3
InternImage-B (ours)	C	224^2	97M	16G	84.9
Swin-L‡ [2]	T	384^2	197M	104G	87.3
CoAtNet-3‡ [20]	T	384^2	168M	107G	87.6
CoAtNet-4‡ [20]	T	384^2	275M	190G	87.9
DeiT III-L‡ [64]	T	384^2	304M	191G	87.7
SwinV2-L/24‡ [16]	T	384^2	197M	115G	87.6
RepLKNet-31L‡ [22]	C	384^2	172M	96G	86.6
HorNet-L‡ [43]	C	384^2	202M	102G	87.7
ConvNeXt-L‡ [21]	C	384^2	198M	101G	87.5
ConvNeXt-XL‡ [21]	C	384^2	350M	179G	87.8
InternImage-L‡ (ours)	C	384^2	223M	108G	87.7
InternImage-XL‡ (ours)	C	384^2	335M	163G	88.0
ViT-G/14$^\#$ [30]	T	518^2	1.84B	5160G	90.5
CoAtNet-6$^\#$ [20]	T	512^2	1.47B	1521G	90.5
CoAtNet-7$^\#$ [20]	T	512^2	2.44B	2586G	90.9
Florence-CoSwin-H$^\#$ [59]	T	–	893M	–	90.0
SwinV2-G$^\#$ [16]	T	640^2	3.00B	–	90.2
RepLKNet-XL$^\#$ [22]	C	384^2	335M	129G	87.8
BiT-L-ResNet152x4$^\#$ [67]	C	480^2	928M	–	87.5
InternImage-H$^\#$ (ours)	C	224^2	1.08B	188G	88.9
InternImage-H$^\#$ (ours)	C	640^2	1.08B	1478G	89.6

Table 2. **Image classification performance on the ImageNet validation set**. "type" refers to model type, where "T" and "C" denote transformer and CNN, respectively. "scale" is the input scale. "acc" is the top-1 accuracy. "‡" indicates the model is pre-trained on ImageNet-22K [31]. "$^\#$" indicates pretraining on extra large-scale private dataset such as JFT-300M [68], FLD-900M [59], or the joint public dataset in this work.

method	#params	#FLOPs	Mask R-CNN 1× schedule						Mask R-CNN 3×+MS schedule					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Swin-T [2]	48M	267G	42.7	65.2	46.8	39.3	62.2	42.2	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T [21]	48M	262G	44.2	66.6	48.3	40.1	63.3	42.8	46.2	67.9	50.8	41.7	65.0	44.9
PVTv2-B2 [11]	45M	309G	45.3	67.1	49.6	41.2	64.2	44.4	47.8	69.7	52.6	43.1	66.8	46.7
ViT-Adapter-S [69]	48M	403G	44.7	65.8	48.3	39.9	62.5	42.8	48.2	69.7	52.5	42.8	66.4	45.9
InternImage-T (ours)	49M	270G	47.2	69.0	52.1	42.5	66.1	45.8	49.1	70.4	54.1	43.7	67.3	47.3
Swin-S [2]	69M	354G	44.8	66.6	48.9	40.9	63.4	44.2	48.2	69.8	52.8	43.2	67.0	46.1
ConvNeXt-S [21]	70M	348G	45.4	67.9	50.0	41.8	65.2	45.1	47.9	70.0	52.7	42.9	66.9	46.2
PVTv2-B3 [11]	65M	397G	47.0	68.1	51.7	42.5	65.7	45.7	48.4	69.8	53.3	43.2	66.9	46.7
InternImage-S (ours)	69M	340G	47.8	69.8	52.8	43.3	67.1	46.7	49.7	71.1	54.5	44.5	68.5	47.8
Swin-B [2]	107M	496G	46.9	–	–	42.3	–	–	48.6	70.0	53.4	43.3	67.1	46.7
ConvNeXt-B [21]	108M	486G	47.0	69.4	51.7	42.7	66.3	46.0	48.5	70.1	53.3	43.5	67.1	46.7
PVTv2-B5 [11]	102M	557G	47.4	68.6	51.9	42.5	65.7	46.0	48.4	69.2	52.9	42.9	66.6	46.2
ViT-Adapter-B [69]	120M	832G	47.0	68.2	51.4	41.8	65.1	44.9	49.6	70.6	54.0	43.6	67.7	46.9
InternImage-B (ours)	115M	501G	48.8	70.9	54.0	44.0	67.8	47.4	50.3	71.4	55.3	44.8	68.7	48.0

method	#param	#FLOPs	Cascade Mask R-CNN 1× schedule						Cascade Mask R-CNN 3×+MS schedule					
Swin-L‡ [2]	253M	1382G	51.8	71.0	56.2	44.9	68.4	48.9	53.9	72.4	58.8	46.7	70.1	50.8
ConvNeXt-L‡ [21]	255M	1354G	53.5	72.8	58.3	46.4	70.2	50.2	54.8	73.8	59.8	47.6	71.3	51.7
RepLKNet-31L‡ [22]	229M	1321G	–	–	–	–	–	–	53.9	72.5	58.6	46.5	70.0	50.6
HorNet-L‡ [43]	259M	1358G	–	–	–	–	–	–	56.0	–	–	48.6	–	–
InternImage-L‡ (ours)	277M	1399G	54.9	74.0	59.8	47.7	71.4	52.1	56.1	74.8	60.7	48.5	72.4	53.0
ConvNeXt-XL‡ [21]	407M	1898G	53.6	72.9	58.5	46.5	70.3	50.5	55.2	74.2	59.9	47.7	71.6	52.2
InternImage-XL‡ (ours)	387M	1782G	55.3	74.4	60.1	48.1	71.9	52.4	56.2	75.0	61.2	48.8	72.5	53.4

Table 3. **Object detection and instance segmentation performance on COCO `val2017`.** The FLOPs are measured with 1280×800 inputs. AP^b and AP^m represent box AP and mask AP, respectively. "MS" means multi-scale training.

two representative object detection frameworks: Mask R-CNN [70], and Cascade Mask R-CNN [71]. We follow common practices [2, 11] to initialize the backbone with pre-trained classification weights, and train models use a 1× (12 epochs) or 3× (36 epochs) schedule by default.

Results. As shown in Table 3, when using Mask R-CNN for object detection, we find that under a comparable number of parameters, our models significantly surpass their counterparts. For example, with the 1× training schedule, the box AP (AP^b) of InternImage-T is 4.5 points better than Swin-T [2] (47.2 *vs*. 42.7), and 3.0 points higher than ConvNeXt-T [21] (47.2 *vs*. 44.2). With the 3× multi-scale training schedule, more parameters, and more advanced Cascade Mask R-CNN [71], InternImage-XL achieves AP^b of 56.2, surpassing ConvNeXt-XL by 1.0 points (56.2 *vs*. 55.2). Similar results are also seen in instance segmentation experiments. With the 1× training schedule, InternImage-T yields 42.5 mask AP (*i.e.*, AP^m), which outperforms Swin-T and ConvNeXt-T by 3.2 points (42.5 *vs*. 39.3) and 2.4 points (42.5 *vs*. 40.1), respectively. The best AP^m 48.8 is obtained by InternImage-XL with Cascade Mask R-CNN, which is at least 1.1 points higher than its counterparts.

To further push the performance bound of object detection, we follow the advanced setting used in leading methods [16, 17, 26, 74, 78] to initialize the backbone with the weights pre-trained on ImageNet-22K or the large-scale joint dataset, and double its parameters via the composite techniques [78] (see the model with 2 billion parameters in Fig. 2). Then, we fine-tune it along with the DINO [74] detector on the Objects365 [79] and COCO datasets one after another for 26 epochs and 12 epochs, respectively. As shown in Table 4, our method achieves the best results of 65.0 AP^b and 65.4 AP^b on COCO val2017 and test-dev. Compared to previous state-of-the-art models, we surpass FD-SwinV2-G [26] by 1.2 points (65.4 *vs*. 64.2), with 27% fewer parameters and without complicated distillation processes, which shows the effectiveness of our models on the detection task.

method	detector	#params	AP^b	
			val2017	test-dev
Swin-L [2]	DyHead [72]	213M	56.2	58.4
Swin-L‡ [2]	HTC++ [2]	284M	58.0	58.7
Swin-L‡ [2]	Soft-Teacher [73]	284M	60.7	61.3
Florence-CoSwin-H# [59]	DyHead [72]	637M	62.0	62.4
ViT-L‡ [9]	ViT-Adapter [69]	401M	62.6	62.6
Swin-L‡ [2]	DINO [74]	218M	63.2	63.3
FocalNet-H‡ [75]	DINO [74]	746M	64.2	64.3
ViT-Huge [76]	Group-DETRv2 [76]	629M	–	64.5
SwinV2-G# [16]	HTC++ [2]	3.00B	62.5	63.1
BEiT-3# [17]	ViTDet [77]	1.90B	–	63.7
FD-SwinV2-G# [26]	HTC++ [2]	3.00B	–	64.2
InternImage-XL‡ (ours)	DINO [74]	602M	64.2	64.3
InternImage-H# (ours)	DINO [74]	2.18B	65.0	65.4

Table 4. **Comparison of the state-of-the-art detectors on COCO `val2017` and test-dev.**

4.3. Semantic Segmentation

Settings. To evaluate the semantic segmentation performance of InternImage, we initialize the backbone with

method	crop size	#params	#FLOPs	mIoU (SS)	mIoU (MS)
Swin-T [2]	512^2	60M	945G	44.5	45.8
ConvNeXt-T [21]	512^2	60M	939G	46.0	46.7
SLaK-T [29]	512^2	65M	936G	47.6	–
InternImage-T (ours)	512^2	59M	944G	47.9	48.1
Swin-S [2]	512^2	81M	1038G	47.6	49.5
ConvNeXt-S [21]	512^2	82M	1027G	48.7	49.6
SLaK-S [29]	512^2	91M	1028G	49.4	–
InternImage-S (ours)	512^2	80M	1017G	50.1	50.9
Swin-B [2]	512^2	121M	1188G	48.1	49.7
ConvNeXt-B [21]	512^2	122M	1170G	49.1	49.9
RepLKNet-31B [22]	512^2	112M	1170G	49.9	50.6
SLaK-B [29]	512^2	135M	1172G	50.2	–
InternImage-B (ours)	512^2	128M	1185G	50.8	51.3
Swin-L‡ [2]	640^2	234M	2468G	52.1	53.5
RepLKNet-31L‡ [22]	640^2	207M	2404G	52.4	52.7
ConvNeXt-L‡ [21]	640^2	235M	2458G	53.2	53.7
ConvNeXt-XL‡ [21]	640^2	391M	3335G	53.6	54.0
InternImage-L‡ (ours)	640^2	256M	2526G	53.9	54.1
InternImage-XL‡ (ours)	640^2	368M	3142G	55.0	55.3
SwinV2-G$^{\#}$ [16]	896^2	3.00B	–	–	59.9
InternImage-H$^{\#}$ (ours)	896^2	1.12B	3566G	59.9	60.3
BEiT-3$^{\#}$ [17]	896^2	1.90B	–	–	62.8
FD-SwinV2-G$^{\#}$ [26]	896^2	3.00B	–	–	61.4
InternImage-H$^{\#}$ (ours) + Mask2Former [80]	896^2	1.31B	4635G	62.5	62.9

Table 5. **Semantic segmentation performance on the ADE20K validation set.** The FLOPs are measured with 512×2048, 640×2560, or 896×896 inputs according to the crop size. "SS" and "MS" means single-scale and multi-scale testing, respectively.

pre-trained classification weights and train our models with UperNet [81] on ADE20K [82] for 160k iterations and compare fairly with previous CNN-based and transformer-based backbones. To further reach top performance, we arm InternImage-H with more advanced Mask2Former [80], and adopt the same training settings in [17, 69].

Results. As shown in Table 5, when using UperNet [81] for semantic segmentation, our InternImage consistently outperforms prior arts [2, 21, 22, 29]. For example, with almost the same parameter numbers and FLOPs, our InternImage-B reports 50.8 mIoU on the ADE20K val, which is outstanding from the strong counterparts such as ConvNeXt-B (50.8 *vs.* 49.1) and RepLKNet-31B (50.8 *vs.* 49.9). Furthermore, our InternImage-H yields 60.3 MS mIoU, which is better than SwinV2-G [16], while the parameter number is much smaller (1.12B *vs.* 3.00B).

It is worth noting that, when using Mask2Former [80] and multi-scale testing, our InternImage-H achieves the best mIoU of 62.9, higher than the current best BEiT-3 [17] on the ADE20K benchmark. These results demonstrate that the CNN-based foundation model can also enjoy the dividends of massive data and challenge the leading position of transformer-based models.

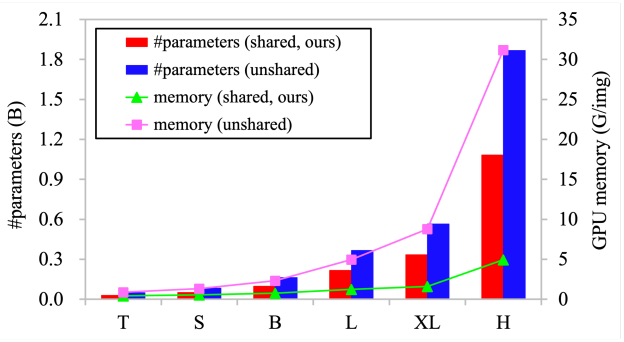

Figure 4. **Model parameters and GPU memory usage of shared weights *v.s* unshared weights among convolution neurons.** The left vertical axis indicates the model parameters and the right one indicates the GPU memory usage per image when the batch size is 32 and the input image resolution is 224 × 224.

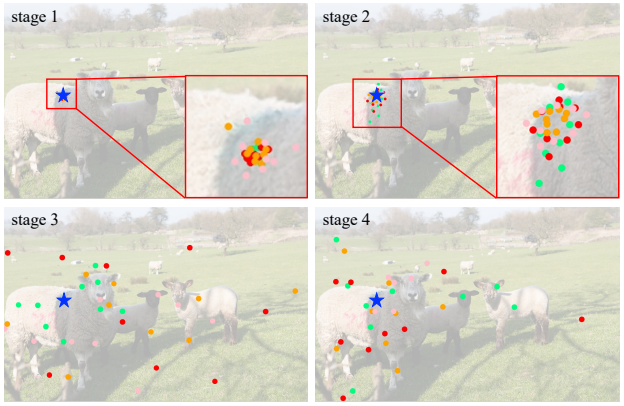

Figure 5. **Visualization of sampling locations for different groups at different stages**. The blue star indicates the query point (on the left sheep), and the dots with different colors indicate the sampling locations of different groups.

4.4. Ablation Study

Sharing weights among convolution neurons matters. Large-scale models are sensitive to parameters and memory cost of the core operator, due to hardware limitations. To address this problem, we share weights among convolution neurons of DCNv3. As shown in Fig. 4, we compare the parameters and memory cost of the models based on DCNv3 with shared or unshared weights. We see that the parameters and memory cost of models with unshared weights are much higher than the shared one, especially for the -H scale, the ratio of saved parameters and GPU memory is 42.0% and 84.2%, respectively. As shown in Table 6, we also examine that the two models at -T scale have similar top-1 accuracy on ImageNet (83.5 *vs.* 83.6) and AP^b on COCO (47.2 *vs.* 47.4), even the model without shared weights has 66.1% more parameters.

Multi-group spatial aggregation brings stronger fea-

shared $\mathbf{w}$	multi-group	softmax	top-1 acc	AP^b	AP^m
✗	✓	✓	83.6	47.4	42.6
✓	✗	✓	82.3	43.8	40.0
✓	✓	✗	65.7	38.7	35.6
✓	✓	✓	83.5	47.2	42.5

Table 6. **Ablation comparison of the three modifications in DCNv3.** These experiments are based on InternImage-T for classification and Mask R-CNN 1× schedule for detection.

tures. We introduce aggregation groups to allow our model to learn information from different representation subspaces like transformers [9]. As shown in Fig. 5, for the same query pixel, the offsets from different groups are concentrated in different regions, resulting in hierarchical semantic features. We also compare the performance of the model with and without multiple groups. As reported in Table 6, the model significantly drops 1.2 points on ImageNet and 3.4 points on COCO val2017. In addition, we also see that in the first two stages, the learned effective receptive field (ERF) is relatively small, and as the model goes deeper (*i.e.*, stages 3 and 4), the ERF increases to be global. This phenomenon is different from ViTs [9, 10, 83] whose ERF is usually global.

5. Conclusion & Limitations

We introduce InternImage, a new large-scale CNN-based foundation model that can provide strong representations for versatile vision tasks, such as image classification, object detection, and semantic segmentation. We tune the flexible DCNv2 operator to satisfy the requirement of foundation models, and develop a series of blocks, stacking and scaling rules centered on the core operator. Extensive experiments on object detection and semantic segmentation benchmarks verify that our InternImage can obtain comparable or better performance than well-designed large-scale vision transformers trained with massive data, showing that CNN is also a considerable choice for large-scale vision foundation model research. Nonetheless, latency remains an issue for DCN-based operators adapting to downstream tasks with high-speed requirements. Also, large-scale CNNs are still in their early stages of development, and we hope InternImage can serve as a good starting point.

Appendix

A. Detailed Training Settings

In this section, we present the detailed training recipes for image classification, object detection, and semantic segmentation.

A.1. Settings for Backbone-Level Comparison

ImageNet image classification. The training details of image classification on ImageNet [31] are shown in Table 7, which are similar to common practices [2, 21, 58, 64] and with some tweaks. To further explore the capability of our model and match the large-scale private data used in previous methods [16, 20, 59], we adopt M3I Pre-training [60], a unified pre-training approach available for both unlabeled and weakly-labeled data, to pre-train InternImage-H on a 427 million joint dataset of public Laion-400M [61], YFCC-15M [62], and CC12M [63] for 30 epochs, and then we fine-tune the model on ImageNet-1K for 20 epochs. For the more detailed pre-training settings of InternImage-H, please refer to M3I Pre-training [60].

COCO object detection. We verify the detection performance of our InternImage on the COCO benchmark [32], on top of Mask R-CNN [70] and Cascade Mask R-CNN [71]. For fair comparisons, we follow common practices [2, 11] to initialize the backbone with pre-trained classification weights, and train these models using a $1\times$ (12 epochs) or $3\times$ (36 epochs) schedule by default. For $1\times$ schedule, the image is resized to have a shorter side of 800 pixels, while the longer side does not exceed 1,333 pixels. During testing, the shorter side of the input image is fixed to 800 pixels. For $3\times$ schedule, the shorter side is resized to $480-800$ pixels, while the longer side does not exceed 1,333 pixels. All these detection models are trained with a batch size of 16 and optimized by AdamW [84] with an initial learning rate of 1×10^{-4}.

ADE20K semantic segmentation. We evaluate our InternImage models on the ADE20K dataset [82], and initialize them with the pre-trained classification weights. For the InternImage-T/S/B models, we optimize them using AdamW [84] with an initial learning rate of 6×10^{-5}, and 2×10^{-5} for InternImage-X/XL. The learning rate is decayed following the polynomial decay schedule with a power of 1.0. Following previous methods [2, 11, 21], the crop size is set to 512 for InternImage-T/S/B, and 640 for InternImage-L/XL. All segmentation models are trained using UperNet [81] with a batch size of 16 for 160k iterations, and compared fairly with previous CNN-based and transformer-based backbones.

A.2. Settings for System-Level Comparison

COCO object detection. For system-level comparison with state-of-the-art large-scale detection models [16, 17, 26, 74, 78], we first initialize the InternImage-XL/H backbone with the weights pre-trained on ImageNet-22K or the 427M large-scale joint dataset, and double its parameters using the composite techniques [78]. Then, we pre-train the model along with the DINO [74] detector on the Objects365 [79] for 26 epochs, with an initial learning rate of 2×10^{-4} and a batch size of 256. The shorter size of input images is resized to $600-1200$ pixels during pre-training, and the learning rate drops by 10 times at epoch 22. Finally, we fine-tune these detectors on the COCO dataset for 12 epochs, where the batch size is 64, and the initial learning rate is 5×10^{-5}, which drops by 10 times at the final epoch.

ADE20K semantic segmentation. To further reach leading segmentation performance, we first initialize our InternImage-H backbone with the pre-trained weights on the 427M large-scale joint dataset, and arm it with the state-of-the-art segmentation method Mask2Former [80]. We follow the same training settings in [17, 69], *i.e.* pre-training and fine-tuning the model on COCO-Stuff [85] and ADE20K [82] datasets both for 80k iterations, with a crop size of 896 and an initial learning rate of 1×10^{-5}.

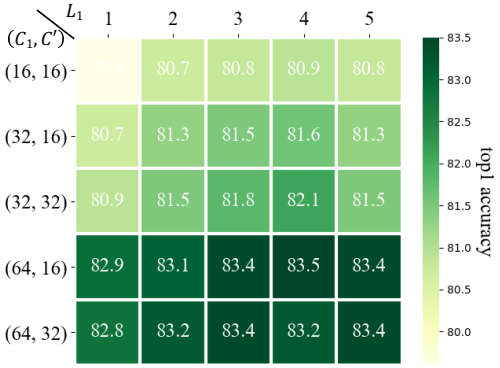

Figure 6. **Comparison of different stacking hyper-parameters**. Each square indicates the accuracy of the model determined by hyperparameter, with the darker the color, the higher the accuracy.

B. Exploration of Hyper-parameters

B.1. Model Stacking

As discussed in Section 3.2, our model is constructed in four stacking rules, and we further restrict the model parameters to 30M for the origin model. We discretize the stacking hyperparameters C_1 to $\{16, 32, 64\}$, L_1 to $\{1, 2, 3, 4, 5\}$, and C' to $\{16, 32\}$. And L_2 is determined by selecting the model size to approximately 30M. In this way, we obtained 30 models by combining the three hyper-parameters.

We adopt the training recipe listed in Table 7 to train our -T models unless otherwise stated. Fig. 6 shows the ImageNet-1K top-1 accuracy of these models under the

settings	InternImage-T	InternImage-S	InternImage-B	InternImage-L		InternImage-XL		InternImage-H
	IN-1K pt	IN-1K pt	IN-1K pt	IN-22K pt	IN-1K ft	IN-22K pt	IN-1K ft	IN-1K ft
input scale	224	224	224	192	384	192	384	224/640
batch size	4096	4096	4096	4096	512	4096	512	512
optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
LR	4×10^{-3}	4×10^{-3}	4×10^{-3}	1×10^{-3}	2×10^{-5}	1×10^{-3}	2×10^{-5}	2×10^{-5}
LR schedule	cosine	cosine	cosine	cosine	cosine	cosine	cosine	cosine
weight decay	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
warmup epochs	5	5	5	5	0	5	0	0
epochs	300	300	300	90	20	90	20	20
horizontal flip	✓	✓	✓	✓	✓	✓	✓	✓
random resized crop	✓	✓	✓	✓	✓	✓	✓	✓
auto augment	✓	✓	✓	✓	✓	✓	✓	✓
layer scale	✗	✓	✓	✓	✓	✓	✓	✓
mixup alpha	0.8	0.8	0.8	0.8	✗	0.8	✗	✗
cutmix alpha	1.0	1.0	1.0	1.0	✗	1.0	✗	✗
erasing prob.	0.25	0.25	0.25	0.25	✗	0.25	✗	✗
color jitter	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
label smoothing ε	0.1	0.1	0.1	0.1	0.3	0.1	0.3	0.3
dropout	✗	✗	✗	✗	✗	✗	✗	✗
drop path rate	0.1	0.4	0.5	0.1	0.1	0.2	0.2	0.2
repeated aug	✗	✗	✗	✗	✗	✗	✗	✗
gradient clip	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
loss	CE	CE	CE	CE	CE	CE	CE	CE

Table 7. **Detailed training recipe for InternImage of different parameter scales on ImageNet [31].** "CE" denotes the cross entropy loss, "LR" denotes the learning rate. The training recipe follows common practices [2, 21, 58, 64] and has some tune-ups. "IN-1K pt", "IN-22K pt", and "IN-1K ft" represent ImageNet-1K pre-training, ImageNet-22K pre-training, and ImageNet-1K fine-tuning, respectively.

same training settings, with darker green indicating higher accuracy, *i.e.*, models with stronger representational capability. When C' equals 16, models are generally higher than that with C' of 32, and L_1 works best at 4, thanks to a reasonable stacking ratio. A large number of channels allows for more gain. Finally, through the above exploration experiments, we determine our basic stacking hyper-parameter (C_1, C', L_1, L_3) to $(64, 16, 4, 18)$.

B.2. Model Scaling

In Section 3.2, we have shown the constraints on the depth scaling factor α and the width scaling factor β. Based on this condition and the -T model (30M), we display reasonable scaling possibilities for extending the -T model to -B models (100M). As illustrated in Table 8, the first two columns show the formulas for α and β. The penultimate column indicates model parameters, and the last column indicates the ImageNet-1K top-1 accuracy of these models after 300 training epochs.

It is worth noting that the model width C_1 needs to be divisible by C'. Therefore some adjustment is required in determining the specific scaling parameters. This results in a small fluctuation in the number of parameters, but this is acceptable. Our exploratory experiments prove that when (α, β) is set at $(1.09, 1.36)$ for the best performance. In addition, the other size models -S/L/XL/H also confirmed the effectiveness of our scaling rules.

scaling factors α	scaling factors β	#parameters	top-1 accuracy (%)
1.03	1.40	118M	84.5
1.06	1.38	95M	83.8
1.09	1.36	97M	84.9
1.12	1.34	105M	83.1
1.15	1.32	95M	81.8

Table 8. **Comparison of different scaling factors**. The default setting is marked with a gray background.

kernel size	#parameters	FLOPs	top-1 accuracy (%)
3×3	30M	5G	83.5
5×5	37M	6G	83.6
7×7	48M	8G	82.8

Table 9. **Comparison of different kernel sizes in our operator**. The default setting is marked with a gray background.

B.3. Kernel Size

As mentioned in Section 3.1, we argue 3×3 dynamic sparse convolution is enough for the large receptive field. Here, we explore the role played by the number of convolutional neurons in the DCNv3 operator. Specifically, we replaced the 3×3 kernel in the DCNv3 operator with the 5×5 or 7×7 kernel. They are all trained by the -T training recipes (see Table 7) and validated on the ImageNet-1K validation set. The results are shown in Table 9.

The results show that when enlarging the convolution kernel, the parameters and FLOPs are followed by the

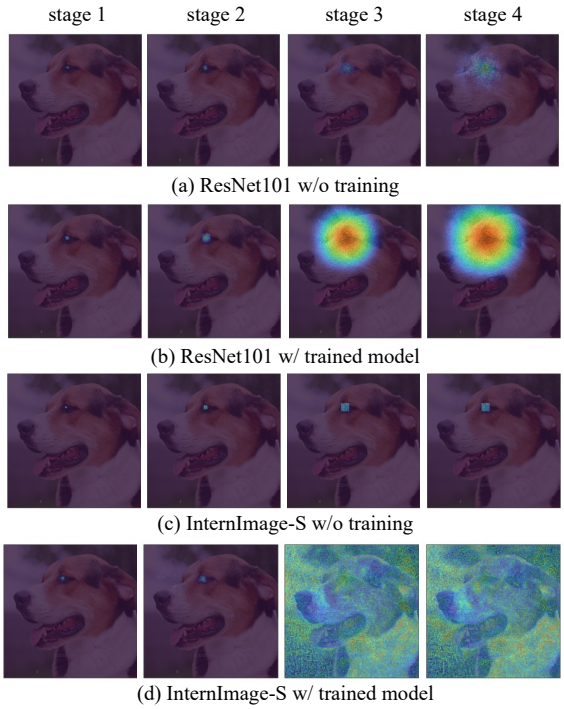

Figure 7. **Visualization of the effective receptive field (ERF) of different backbones.** The activated pixel is at dog's eye. (a) and (b) shows the ERF of ResNet-101 [36] with (w/) and without (w/o) training on ImageNet-1K [31], respectively. (c) and (d) are the ERF of InternImage-B with (w/) and without (w/o) training on ImageNet-1K.

surge, while the accuracy is not significantly improved (83.5 *v.s* 83.6) or even decreased (83.5 *v.s* 82.8). These results show that when the number of convolutional neurons in a single layer increases, the model becomes more difficult to optimize. This phenomenon is also confirmed in RepLKNet [22], and it addresses this problem by re-parameterizing [22] techniques, which might bring extra time and memory costs in the training phase. In this work, we avoid this problem by adopting the simple yet effective 3×3 DCNv3 as InternImage's core operator.

Fig. 7 shows the effective receptive fields (ERF) of ResNet-101 [36] and InternImage-S. A wider distribution of bright areas indicates a larger ERF. We uniformly activate the input image at the dog's eye, count the gradient map of each block, aggregate by channel, and map back to the input image. We see that the ERF of ResNet-101 [36] without training is limited to a local area, while the fully trained ResNet-101 still has an ERF around the eye, and the gradient amplitude is lower, and the distribution is more sparse. Therefore, the area that ResNet-101 can effectively perceive is very limited. For the InternImage-S without training, its ERF is concentrated around the activation point. Since the offset is not learned, its ERF is also very small in the last two blocks. But after sufficient training, InternImage-L can effectively perceive the information of the entire image in the 3-rd and 4-th stages.

C. Additional Downstream Tasks

C.1. Classification

iNaturalist 2018 [98] is a read-word long-tailed dataset containing 8142 fine-graned species. The dataset comprises 437.5K training images and an imbalance factor of 500. For this experiment, we initialize our InternImage-H model with the pre-trained weights on the 427M large-scale joint dataset, and fine-tune it on the training set of iNaturalist 2018 for 100 epochs. We follow MetaFormer [86] to adopt a resolution of 384×384 for fine-tuning, with the utilization of meta information. Other training settings are the same as the recipe for fine-tuning InternImage-H on ImageNet-1K, as reported in Table 7. As a result, our method achieves the state-of-the-art accuracy of 92.6 (see Table 10) on the validation set of iNaturalist 2018, 3.9 points better than the previous best model MetaFormer [86].

Places205 [99] is a dataset containing 2.5 million images of 205 scene categories, which are dedicated to the scene recognition task. The images in this dataset cover a wide range of indoor and outdoor scenes, such as offices, kitchens, forests, and beaches. We initialize our model with pre-trained weights on a large-scale joint dataset, consisting of 427 million images, and fine-tune it on the Places205 training set. Other training settings are the same as the recipe for fine-tuning InternImage-H on ImageNet-1K, as reported in Table 7. Our method achieves state-of-the-art accuracy of 71.7 (see Table 10) on the validation set of Places205, outperforming the previous best model MixMIM-L [87] by 2.4 points.

Places365 [100] is a dataset containing 1.8 million images of 365 scene categories, which are dedicated to the scene recognition task. The images in this dataset cover a wide range of indoor and outdoor scenes, such as airports, bedrooms, deserts, and waterfalls. The specific pre-training and fine-tuning strategies are the same as for Places205. Our method achieves state-of-the-art accuracy of 61.2 (see Table 10) on the validation set of Places365, outperforming the previous best model SWAG [88] by 0.5 points. The Places365 dataset provides a more fine-grained classification task compared to Places205, allowing our model to learn more subtle differences between similar scenes.

C.2. Object Detection

LVIS v1.0 [101] is a large-scale vocabulary dataset for object detection and instance segmentation tasks, which contains 1203 categories in 164k images. For this dataset, we initialize our InternImage-H with the Objects365 [79]

method	classification			semantic segmentation				
	iNaturalist2018	Places205	Places365	COCO-Stuff-10K	Pascal Context	Cityscapes (val)	Cityscapes (test)	NYU Depth V2
previous best	88.7[a]	69.3[b]	60.7[c]	54.2[d]	68.2[d]	86.9[e]	85.2[d]	56.9[f]
InternImage-H	92.6 (+3.9)	71.7 (+2.4)	61.2 (+0.5)	59.6 (+5.4)	70.3 (+2.1)	87.0 (+0.1)	86.1 (+0.9)	68.1 (+11.2)

method	object detection						
	LVIS (minival)	LVIS (val)	VOC2007	VOC2012	OpenImages	CrowdHuman	BDD100K
previous best	59.8[g]	62.2[h]	89.3[i]	92.9[j]	72.2[k]	94.1[l]	35.6[m]
InternImage-H	65.8 (+6.0)	63.2 (+1.0)	94.0 (+4.7)	97.2 (+4.3)	74.1 (+1.9)	97.2 (+3.1)	38.8 (+3.2)

Table 10. **Summary of InternImage-H performance on various mainstream vision benchmarks**. a: MetaFormer [86]. b: MixMIM-L [87]. c: SWAG [88]. d: ViT-Adapter [69]. e: PSA [89]. f: CMX-B5 [90]. g: GLIPv2 [91]. h: EVA [92]. i: Cascade Eff-B7 NAS-FPN [93]. j: ATLDETv2 [94]. k: OpenImages 2019 competition 1st [95]. l: Iter-Deformable-DETR [96]. m: PP-YOLOE [97].

pre-trained weights, then fine-tune it on the training set of LVIS v1.0. Here, we report the box AP (*i.e.*, AP^b) with multi-scale testing on the minival set and the val set, respectively. As shown in Table 10, our InternImage-H creates a new record of 65.8 AP^b on the LVIS minival, and 63.2 AP^b on the LVIS val, outperforming previous state-of-the-art methods by clear margins.

Pascal VOC [102] contains 20 object classes, which has been widely used as a benchmark for object detection tasks. We adopt this dataset to further evaluate the detection performance of our model. Specifically, we employ the Objects365 [79] pre-trained weights to initialize our InternImage-H, and fine-tune it on the trainval set of Pascal VOC 2007 and Pascal VOC 2012 following previous method [93]. As shown in Table 10, on the Pascal VOC 2007 test set, our InternImage-H yields 94.0 AP^{50} with single-scale testing, which is 4.7 points better than previous best Cascade Eff-B7 NAS-FPN [93]. On the Pascal VOC 2012 test set, our method achieves 97.2 mAP, 4.3 points higher than the best record on the official leaderboard [94].

OpenImages v6 [103] is a dataset of about 9 million images with 16M bounding boxes for 600 object classes on 1.9 million images dedicated to the object detection task, which are very diverse and often embrace complex scenes with multiple objects (8.3 per image on average). For this dataset, we use the same settings as the previous two datasets. In addition, we follow [95] to use the class-aware sampling during fine-tuning. As reported in Table 10, our InternImage-H yields 74.1 mAP, achieving 1.9 mAP improvement compared to the previous best results [95].

CrownHuman [104] is a benchmark dataset to better evaluate detectors in crowd scenarios. The CrowdHuman dataset is large, rich-annotated and contains high diversity. CrowdHuman contains 15000, 4370 and 5000 images for training, validation, and testing, respectively. There are a total of 470K human instances from train and validation subsets and 23 persons per image, with various kinds of occlusions in the dataset. We used the same training setup as for the previous dataset. Our pre-trained model reached optimal performance in 3750 iterations, exceeding the previous best model Iter-Deformable-DETR [96] by 3.1 AP.

BDD100K [105] is a dataset of around 100K high-resolution images with diverse weather and lighting conditions, containing 10 object categories, including pedestrians, cars, buses, and bicycles, dedicated to the object detection task. The images in this dataset are captured from a moving vehicle, simulating real-world scenarios. For this experiment, we initialize our InternImage-H model with the pre-trained weights on the 427M joint dataset and fine-tune it on the BDD100K training set for 12 epochs. As reported in Table 10, our InternImage-H achieves 38.8 mAP on the validation set, which is the state-of-the-art performance, surpassing the previous best model by 3.2 mAP. Our method demonstrates superior performance in detecting objects in real-world driving scenarios, which can benefit autonomous driving and intelligent transportation systems.

C.3. Semantic Segmentation

COCO-Stuff [85] includes the images from the COCO [32] dataset for semantic segmentation, spanning over 171 categories. Specifically, COCO-Stuff-164K is the full set that contains all 164k images, while COCO-Stuff-10K is a subset of the -164K that splits into 9,000 and 1,000 images for training and testing. Here, we equip our InternImage-H with the advanced Mask2Former [80], and pre-train the model on the COCO-Stuff-164K for 80k iterations. Then we fine-tune it on the COCO-Stuff-10K for 40k iterations and report the multi-scale mIoU. The crop size is set to 512×512 in this experiment. As shown in Table 10, our model achieves 59.6 MS mIoU on the test set, outperforming the previous best ViT-Adapter [69] by 5.4 mIoU.

Pascal Context [106] contains 59 semantic classes. It is divided into 4,996 images for training and 5,104 images for testing. For this dataset, we also employ Mask2Former with our InternImage-H, and follow the training settings in [69]. Specifically, we first load the classification pre-trained weights to initialize the model, then fine-tune it on the training set of Pascal Context for 40k iterations. The crop size is set to 480×480 in this experiment. As shown in Table 10, our method reports 70.3 MS mIoU on the test set, which is 2.1 points better than ViT-Adapter [69].

Cityscapes [107] is a high-resolution dataset recorded

method	#params	scale	FLOPs	acc (%)	throughput (img/s)
InternImage-B (ours)	97M	224^2	16G	84.9	775
		800^2	206G	–	54
InternImage-B-DCNv2 [28]	146M	224^2	24G	–	311
		800^2	313G	–	16
ConvNeXt-B [21]	88M	224^2	15G	83.8	881
		800^2	196G	–	58
RepLKNet-B [22]	79M	224^2	15G	83.5	884
		800^2	198G	–	21
DAT-B [21]	88M	224^2	16G	84.0	661
		800^2	194G	–	24

Table 11. **Throughput comparison of different models under different input resolutions.** "#params" denotes the number of parameters. "acc" represents the top-1 accuracy on the ImageNet-1K validation set. The throughputs of 224×224 and 800×800 input resolutions are tested with the batch size of 256 and 2 respectively, using a single A100 GPU.

in street scenes including 19 classes. In this experiment, we use Mask2Former [80] as the segmentation framework. Following common practices [69, 83, 108], we first pre-train on Mapillary Vistas [109] and then fine-tune on Cityscapes for 80k iterations, respectively. The crop size is set to 1024×1024 in this experiment. As shown in Table 10, our InternImage-H achieves 87.0 MS mIoU on the validation set, and 86.1 MS mIoU on the test set.

NYU Depth V2 [110] comprises of 1449 RGB-D images, each with a size of 640×480. These images are divided into 795 training and 654 testing images, each with annotations on 40 semantic categories. We adopt the same training settings as we used when fine-tuning on Pascal Context. As shown in Table 10, our method achieves a big jump to 68.1 MS mIoU on the validation set, which is 11.2 points better than CMX-B5 [90].

D. Throughput Analysis

In this section, we benchmark the throughput of our InternImage with counterparts, including a variant equipped with DCNv2 [28], ConvNext [21], RepLKNet [22], and a vision transformer with deformable attention (DAT) [46]. As shown in Table 11, compared to the variant with DCNv2 [28], our model enjoys better parameter-efficient and significantly faster inference speed under both 224×224 and 800×800 input resolutions. Compared to RepLKNet-B [22] and DAT-B [46], our model has a throughput advantage at a high input resolution (*i.e.*, 800×800). This resolution is widely used in dense prediction tasks such as object detection. Compared with ConvNeXt [21], despite the throughput gap due to DCN-based operators, our model still has an accuracy advantage (84.9 *vs*. 83.8), and we are also looking for an efficient DCN to make our model more suitable for downstream tasks that require high efficiency.

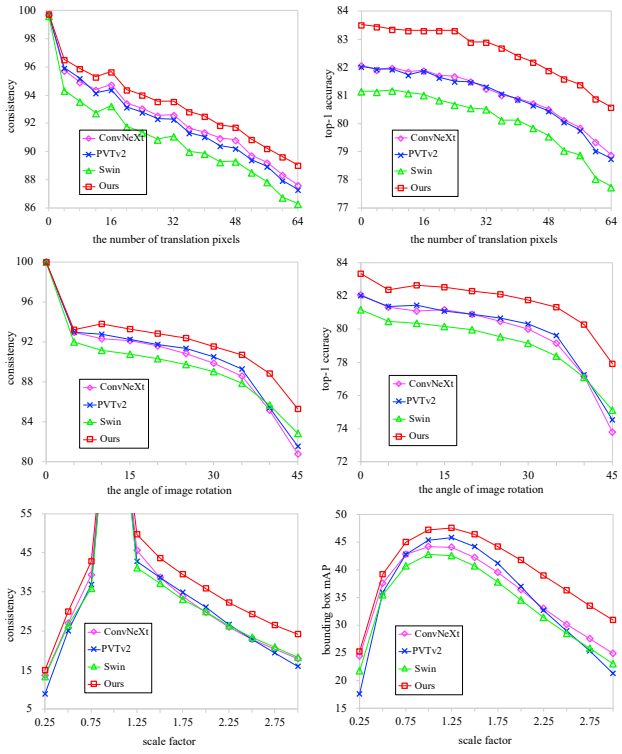

Figure 8. **Comparison of robust evaluation of different methods.** These results show that our model has better robustness in terms of translation, rotation, and input resolution.

E. Robustness Evaluation on ImageNet

In this section, we evaluate the robustness of different models under different transformations (see Fig. 8). We consider translation, rotation, and scaling to evaluate. The models we choose for comparison include a convolutional model (ConvNeXt-T [21]), a local attention-based model (Swin-T [2]), a global attention-based model (PVTv2-B2 [11]), and our InternImage-T.

E.1. Translation Invariance

Translation invariance describes the capability of the model to retain the original output when the input image is translated. We evaluate the translation invariance in the classification task by dithering the image from 0 to 64 pixels. The invariance is measured by the probability that the model predicts the same label when the same input image is translated. The first row of Fig. 8 indicates our InternImagehas the translation invariance of the different methods. It is evident that the robustness of the four models to translation is shown as our method is the best, followed by convolution-based ConvNeXt, followed by global attention-based PVTv2, and the worst local attention-based Swin transformer.

E.2. Rotation Invariance

To evaluate the rotation invariance of the classification task, we rotate the image from 0° to 45° in steps of 5°. In a similar way to translation invariance, the predicted consistency under different rotation angles is used to evaluate the rotational invariance. From the second row of Fig. 8, we found that the consistency performance of all models is comparable in the small angle phase. However, at large-angle rotation (*i.e.*, $> 10^\circ$), our model is clearly superior to the other models.

E.3. Scaling Invariance

We evaluate the scaling invariance on object detection. The scaling factor of the input image varies from 0.25 to 3.0 in steps of 0.25. Detection consistency is defined as the invariance metric for the detection task. The predicted boxes on the scaled images are first converted back to the original resolution, and then the predicted boxes at the original resolution are used as the ground truth boxes to calculate the box mAP. As seen in the last row of Fig. 8, we can observe that all methods of our experiments are sensitive to down-scaling. And they show invariance comparable to the input at small resolutions. Our method performs better when scaling up the images. Both box consistency and bounding box mAP are better than the others.

E.4. How Hungry the Model is for Data Scale?

In order to verify the robustness of the model to the data scale. We uniformly sampled the ImageNet-1K data to obtain 1%, 10%, and 100% data, respectively. And we chose ResNet-50 [36], ConvNeXt-T [21], Swin-T [2], InternImage-T-dattn and our InternImage-T to conduct 300 rounds of training experiments on these data. The experimental settings are consistent with Table 7. The experimental results can be viewed in Table 12. We see that ResNet [36] performs best on the 1% and 10% data (12.2% & 57.5%), benefiting from its inductive biases. But its upper limitation is low (80.4%) when the data is sufficient. Swin-T fails completely in 1% datasets and shows good performance only on the 100% dataset. The proposed InternImage-T has strong robustness not only on 1% and 10% data (5.9% and 56.0%) but also on full data (83.5%), which is consistently better than the InternImage-T variant with deformable attention (dattn) and ConvNeXt [21]. These results indicate the robustness of our model with respect to the data scale.

method	1%	10%	100%
ResNet-50 [36]	12.2	57.5	80.4
ConvNeXt-T [21]	8.4	52.6	82.1
Swin-T [2]	failed	12.1	81.3
InternImage-T-dattn [56]	4.1	49.9	81.9
InternImage-T (ours)	5.9	56.0	83.5

Table 12. **Accuracy of different models at different data scales**. "InternImage-dattn" refers to the model variant equipped with deformable attention [56].

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 1, 2, 4, 5

[2] 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 *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 1, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15

[3] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 1

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

[5] 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. *Journal of Machine Learning Research*, 21:1–67, 2020. 1

[6] 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. *Adv. Neural Inform. Process. Syst.*, 33:1877–1901, 2020. 1

[7] 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. 1

[8] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. 1

[9] 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 *Int. Conf. Learn. Represent.*, 2020. 1, 2, 3, 5, 7, 9

[10] 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 *Int. Conf. Comput. Vis.*, pages 568–578, 2021. 1, 3, 5, 6, 9

[11] 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. 1, 2, 3, 5, 6, 7, 10, 14

[12] 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. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12124–12134, 2022. 1, 6

[13] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Int. Conf. Comput. Vis.*, pages 22–31, 2021. 1

[14] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Adv. Neural Inform. Process. Syst.*, 34, 2021. 1

[15] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Adv. Neural Inform. Process. Syst.*, 34, 2021. 1

[16] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *Adv. Neural Inform. Process. Syst.*, pages 12009–12019, 2022. 1, 2, 3, 6, 7, 8, 10

[17] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 1, 3, 7, 8, 10

[18] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Adv. Neural Inform. Process. Syst.*, 34:8583–8595, 2021. 1, 2

[19] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12104–12113, 2022. 1, 2, 3

[20] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Adv. Neural Inform. Process. Syst.*, 34:3965–3977, 2021. 1, 2, 3, 6, 10

[21] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 2, 3, 5, 6, 7, 8, 10, 11, 14, 15

[22] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11963–11975, 2022. 2, 3, 5, 6, 7, 8, 12, 14

[23] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10819–10829, 2022. 2

[24] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2, 4, 5

[25] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arxiv. *arXiv preprint arXiv:1606.08415*, 2016. 2, 5

[26] 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. 2, 7, 8, 10

[27] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Int. Conf. Comput. Vis.*, pages 764–773, 2017. 2, 3

[28] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9308–9316, 2019. 2, 3, 14

[29] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 2, 6, 8

[30] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12104–12113, 2022. 2, 6

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009. 2, 5, 6, 10, 11, 12

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 2, 6, 10, 13

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2, 4

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–9, 2015. 3

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 3, 5, 12, 15

[37] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1492–1500, 2017. 3

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning.*, pages 6105–6114. PMLR, 2019. 3, 5

[39] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning.*, pages 10096–10106. PMLR, 2021. 3

[40] 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. 3

[41] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11963–11975, 2022. 3

[42] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 3

[43] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *arXiv preprint arXiv:2207.14284*, 2022. 3, 6, 7

[44] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *Int. Conf. Learn. Represent.*, 2021. 3

[45] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[46] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4794–4803, 2022. 3, 5, 14

[47] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12894–12904, 2021. 3

[48] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 3

[49] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. *arXiv preprint arXiv:2204.03645*, 2022. 3

[50] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Int. Conf. Comput. Vis.*, pages 6688–6697, 2019. 3

[51] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017. 3

[52] L-CCGP Florian and Schroff Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 6, 2017. 3

[53] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, pages 801–818, 2018. 3

[54] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 3

[55] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1251–1258, 2017. 4

[56] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 5, 15

[57] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning.*, pages 10524–10533. PMLR, 2020. 5

[58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning.*, pages 10347–10357, 2021. 6, 10, 11

[59] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 6, 7, 10

[60] Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. Towards all-in-one pre-training via maximizing multi-modal mutual information. *arXiv preprint arXiv:2211.09807*, 2022. 6, 10

[61] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6, 10

[62] 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. 6, 10

[63] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3558–3568, 2021. 6, 10

[64] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 6, 10, 11

[65] 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. 6

[66] Ismail Khalfaoui Hassani, Thomas Pellegrini, and Timothée Masquelier. Dilated convolution with learnable spacings. *arXiv preprint arXiv:2112.03740*, 2021. 6

[67] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Eur. Conf. Comput. Vis.*, pages 491–507. Springer, 2020. 6

[68] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10687–10698, 2020. 6

[69] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 7, 8, 10, 13, 14

[70] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. 7, 10

[71] 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. 7, 10

[72] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7373–7382, 2021. 7

[73] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Int. Conf. Comput. Vis.*, pages 3060–3069, 2021. 7

[74] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 7, 10

[75] Jianwei Yang, Chunyuan Li, and Jianfeng Gao. Focal modulation networks. *arXiv preprint arXiv:2203.11926*, 2022. 7

[76] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 7

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

[78] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. Cbnet: A composite backbone network architecture for object detection. *IEEE Trans. Image Process.*, 2022. 7, 10

[79] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Int. Conf. Comput. Vis.*, pages 8430–8439, 2019. 7, 10, 12, 13

[80] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021. 8, 10, 13, 14

[81] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Eur. Conf. Comput. Vis.*, pages 418–434, 2018. 8, 10

[82] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 633–641, 2017. 8, 10

[83] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inform. Process. Syst.*, 34, 2021. 9, 14

[84] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 10

[85] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1209–1218, 2018. 10, 13

[86] Qishuai Diao, Yi Jiang, Bin Wen, Jia Sun, and Zehuan Yuan. Metaformer: A unified meta framework for fine-grained recognition. *arXiv preprint arXiv:2203.02751*, 2022. 12, 13

[87] Jihao Liu, Xin Huang, Yu Liu, and Hongsheng Li. Mixmim: Mixed and masked image modeling for efficient visual representation learning. *arXiv preprint arXiv:2205.13137*, 2022. 12, 13

[88] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens Van Der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 804–814, 2022. 12, 13

[89] Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv preprint arXiv:2107.00782*, 2021. 13

[90] Huayao Liu, Jiaming Zhang, Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *arXiv preprint arXiv:2203.04838*, 2022. 13, 14

[91] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022. 13

[92] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. 13

[93] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2918–2928, 2021. 13

[94] Xuan Jin, Wei Su, Rong Zhang, Yuan He, and Hui Xue. Atldetv2. `http://host.robots.ox.ac.uk/leaderboard/displaylb_main.php?challengeid=11&compid=4`, 2019. 13

[95] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place solutions for openimage2019–object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2020. 13

[96] Anlin Zheng, Yuang Zhang, Xiangyu Zhang, Xiaojuan Qi, and Jian Sun. Progressive end-to-end object detection in crowded scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 857–866, 2022. 13

[97] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. Pp-yoloe: An evolved version of yolo. *arXiv preprint arXiv:2203.16250*, 2022. 13

[98] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8769–8778, 2018. 12

[99] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Adv. Neural Inform. Process. Syst.*, 27, 2014. 12

[100] Alejandro López-Cifuentes, Marcos Escudero-Vinolo, Jesús Bescós, and Álvaro García-Martín. Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, 2020. 12

[101] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5356–5364, 2019. 12

[102] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 13

[103] 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. *Int. J. Comput. Vis.*, 128(7):1956–1981, 2020. 13

[104] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 13

[105] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2636–2645, 2020. 13

[106] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 891–898, 2014. 13

[107] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 13

[108] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 14

[109] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Int. Conf. Comput. Vis.*, pages 4990–4999, 2017. 14

[110] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *Eur. Conf. Comput. Vis.*, 7576:746–760, 2012. 14