Supplementary Materials

A. Comprehensive Comparison of Image Classification Performance

Due to space limitations in the main body of the paper, we present a more comprehensive comparison of image classification performance in Table A.

B. Detailed Training Settings

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

B.1. Settings for Backbone-Level Comparison

**ImageNet image classification.** The training details of image classification on ImageNet [18] are shown in Table B, which are similar to common practices [1, 3, 7, 10] and with some tweaks. To further explore the capability of our model and match the large-scale private data used in previous methods [4, 8, 16], we adopt M3I Pre-training [20], 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 [21], YFCC-15M [22], and CC12M [23] 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 [20].

**COCO object detection.** We verify the detection performance of our InternImage on the COCO benchmark [24], on top of Mask R-CNN [25] and Cascade Mask R-CNN [26]. For fair comparisons, we follow common practices [3, 6] to initialize the backbone with pre-trained classification weights, and train these models using a 1× (12 epochs) or 3× (36 epochs) schedule by default. For 1× 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× 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 [27] with an initial learning rate of $1 \times 10^{-4}$.

**ADE20k semantic segmentation.** We evaluate our InternImage models on the ADE20k dataset [28], and initialize them with the pre-trained classification weights. For the InternImage-T/S/B models, we optimize them using AdamW [27] with an initial learning rate of $6 \times 10^{-5}$, and $2 \times 10^{-5}$ for InternImage-X/XL. The learning rate is decayed following the polynomial decay schedule with a power of 1.0. Following previous methods [3, 6, 10], the crop size is set to 512 for InternImage-T/S/B, and 640 for

<table>
<thead>
<tr>
<th>method</th>
<th>type</th>
<th>scale</th>
<th>#params</th>
<th>#FLOPs</th>
<th>acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S [1]</td>
<td>T</td>
<td>224</td>
<td>22M</td>
<td>5G</td>
<td>79.9</td>
</tr>
<tr>
<td>PVTv2-S [2]</td>
<td>T</td>
<td>224</td>
<td>25M</td>
<td>4G</td>
<td>79.8</td>
</tr>
<tr>
<td>CoAtNet-0 [4]</td>
<td>T</td>
<td>224</td>
<td>25M</td>
<td>4G</td>
<td>81.6</td>
</tr>
<tr>
<td>DeiT III-S [7]</td>
<td>T</td>
<td>224</td>
<td>22M</td>
<td>5G</td>
<td>81.4</td>
</tr>
<tr>
<td>SwinV2-T8 [8]</td>
<td>T</td>
<td>256</td>
<td>28M</td>
<td>6G</td>
<td>81.8</td>
</tr>
<tr>
<td>Focal-T [9]</td>
<td>T</td>
<td>224</td>
<td>29M</td>
<td>5G</td>
<td>82.2</td>
</tr>
<tr>
<td>ConvNeXt-T [10]</td>
<td>C</td>
<td>224</td>
<td>29M</td>
<td>5G</td>
<td>82.1</td>
</tr>
<tr>
<td>ConvNeXt-T-dcls [11]</td>
<td>C</td>
<td>224</td>
<td>29M</td>
<td>5G</td>
<td>82.5</td>
</tr>
<tr>
<td>SLaK-T [12]</td>
<td>C</td>
<td>224</td>
<td>30M</td>
<td>5G</td>
<td>82.5</td>
</tr>
<tr>
<td>HorNet-T [13]</td>
<td>C</td>
<td>224</td>
<td>23M</td>
<td>4G</td>
<td>83.0</td>
</tr>
<tr>
<td>InternImage-T (ours)</td>
<td>C</td>
<td>224</td>
<td>30M</td>
<td>5G</td>
<td>83.5</td>
</tr>
<tr>
<td>Swin-S [3]</td>
<td>T</td>
<td>224</td>
<td>50M</td>
<td>9G</td>
<td>83.0</td>
</tr>
<tr>
<td>CoAtNet-2 [4]</td>
<td>T</td>
<td>224</td>
<td>42M</td>
<td>8G</td>
<td>83.3</td>
</tr>
<tr>
<td>PVTv2-B4 [6]</td>
<td>T</td>
<td>224</td>
<td>63M</td>
<td>10G</td>
<td>83.6</td>
</tr>
<tr>
<td>SwinV2-S8 [5]</td>
<td>T</td>
<td>256</td>
<td>50M</td>
<td>12G</td>
<td>83.7</td>
</tr>
<tr>
<td>ConvNeXt-S [10]</td>
<td>C</td>
<td>224</td>
<td>50M</td>
<td>9G</td>
<td>83.1</td>
</tr>
<tr>
<td>SLaK-S [12]</td>
<td>C</td>
<td>224</td>
<td>55M</td>
<td>10G</td>
<td>83.8</td>
</tr>
<tr>
<td>HorNet-S [13]</td>
<td>C</td>
<td>224</td>
<td>50M</td>
<td>9G</td>
<td>84.0</td>
</tr>
<tr>
<td>InternImage-S (ours)</td>
<td>C</td>
<td>224</td>
<td>50M</td>
<td>8G</td>
<td>84.2</td>
</tr>
<tr>
<td>DeiT-B [1]</td>
<td>T</td>
<td>224</td>
<td>87M</td>
<td>18G</td>
<td>83.1</td>
</tr>
<tr>
<td>CoAtNet-2 [4]</td>
<td>T</td>
<td>224</td>
<td>75M</td>
<td>16G</td>
<td>84.1</td>
</tr>
<tr>
<td>PVTv2-B5 [6]</td>
<td>T</td>
<td>224</td>
<td>82M</td>
<td>12G</td>
<td>83.8</td>
</tr>
<tr>
<td>DeiT III-B [7]</td>
<td>T</td>
<td>224</td>
<td>87M</td>
<td>18G</td>
<td>83.8</td>
</tr>
<tr>
<td>SwinV2-B8 [5]</td>
<td>T</td>
<td>256</td>
<td>88M</td>
<td>20G</td>
<td>84.2</td>
</tr>
<tr>
<td>ConvNeXt-B [10]</td>
<td>C</td>
<td>224</td>
<td>85M</td>
<td>15G</td>
<td>83.8</td>
</tr>
<tr>
<td>SLaK-B [12]</td>
<td>C</td>
<td>224</td>
<td>95M</td>
<td>17G</td>
<td>84.0</td>
</tr>
<tr>
<td>HorNet-B [13]</td>
<td>C</td>
<td>224</td>
<td>88M</td>
<td>16G</td>
<td>84.3</td>
</tr>
<tr>
<td>InternImage-B (ours)</td>
<td>C</td>
<td>224</td>
<td>97M</td>
<td>16G</td>
<td>84.9</td>
</tr>
<tr>
<td>CoAtNet-4 [4]</td>
<td>T</td>
<td>384</td>
<td>275M</td>
<td>190G</td>
<td>87.9</td>
</tr>
<tr>
<td>RePLKNet-31L7 [14]</td>
<td>C</td>
<td>384</td>
<td>172M</td>
<td>96G</td>
<td>86.6</td>
</tr>
<tr>
<td>HorNet-L7 [13]</td>
<td>C</td>
<td>384</td>
<td>202M</td>
<td>102G</td>
<td>87.7</td>
</tr>
<tr>
<td>ConvNeXt-L7 [10]</td>
<td>C</td>
<td>384</td>
<td>198M</td>
<td>101G</td>
<td>87.5</td>
</tr>
<tr>
<td>ConvNeXt-XL [10]</td>
<td>C</td>
<td>384</td>
<td>350M</td>
<td>179G</td>
<td>87.8</td>
</tr>
<tr>
<td>InternImage-L7 (ours)</td>
<td>C</td>
<td>384</td>
<td>223M</td>
<td>108G</td>
<td>87.7</td>
</tr>
<tr>
<td>InternImage-XL (ours)</td>
<td>C</td>
<td>384</td>
<td>325M</td>
<td>163G</td>
<td>88.0</td>
</tr>
<tr>
<td>ViT-G14*/[15]</td>
<td>T</td>
<td>512</td>
<td>1.36B</td>
<td>5160G</td>
<td>90.5</td>
</tr>
<tr>
<td>CoAtNet-6* [4]</td>
<td>T</td>
<td>512</td>
<td>1.47B</td>
<td>1521G</td>
<td>90.5</td>
</tr>
<tr>
<td>CoAtNet-7* [4]</td>
<td>T</td>
<td>512</td>
<td>2.44B</td>
<td>2586G</td>
<td>90.9</td>
</tr>
<tr>
<td>Florence-CoSwin-H* [16]</td>
<td>T</td>
<td>893M</td>
<td>-</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>SwinV2-G* [8]</td>
<td>T</td>
<td>640</td>
<td>3.00B</td>
<td>-</td>
<td>90.2</td>
</tr>
<tr>
<td>RePLKNet-XL* [14]</td>
<td>C</td>
<td>384</td>
<td>335M</td>
<td>129G</td>
<td>87.8</td>
</tr>
<tr>
<td>BiTL-ResNet152x4*/[17]</td>
<td>C</td>
<td>480</td>
<td>928M</td>
<td>-</td>
<td>87.5</td>
</tr>
<tr>
<td>InternImage-H* (ours)</td>
<td>C</td>
<td>224</td>
<td>1.08B</td>
<td>188G</td>
<td>88.9</td>
</tr>
<tr>
<td>InternImage-H* (ours)</td>
<td>C</td>
<td>640</td>
<td>1.08B</td>
<td>1478G</td>
<td>89.6</td>
</tr>
</tbody>
</table>

Table A. 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 [18]. "#" indicates pretraining on extra large-scale private dataset such as JFT-300M [19], FLD-900M [16], or the joint public dataset in this work.
parameters to 30M for the origin model. We discretize in four stacking rules, and we further restrict the model size of 896 and an initial learning rate of 1

ADE20K [28] datasets both for 80k iterations, with a crop training and fine-tuning the model on COCO-Stuff [37] and i.e. We follow the same training settings in [30, 36], state-of-the-art segmentation method Mask2Former [35].

the 427M large-scale joint dataset, and arm it with the leading segmentation performance, we first initialize our -T models unless otherwise stated. Fig. A shows the ImageNet-1K top-1 accuracy of these models under the 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 L1 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 (C1, C', L1, L3) to (64, 16, 4, 18).

C.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 C, 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 C1 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.

C.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 B) and validated on the ImageNet-1K validation set. The results are shown in Table D.

The results show that when enlarging the convolution kernel, the parameters and FLOPs are followed by the 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 [14], and it addresses this problem by re-
Table B. Detailed training recipe for InternImage of different parameter scales on ImageNet [18]. “CE” denotes the cross entropy loss, “LR” denotes the learning rate. The training recipe follows common practices [1, 3, 7, 10] 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.

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

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

D. Additional Downstream Tasks

D.1. Classification

iNaturalist 2018 [51] 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 [39] 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 B. As a result, our method achieves the state-of-the-art accuracy of 92.6 (see Table E) on the validation set of iNaturalist 2018, 3.9 points better than the previous best model MetaFormer [39].

Places205 [52] 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
Table E. Summary of InternImage-H performance on various mainstream vision benchmarks.

<table>
<thead>
<tr>
<th>method</th>
<th>classification</th>
<th>COCO-Stuff-10K</th>
<th>semantic segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>iNaturalist2018</td>
<td>Places205</td>
<td>Places365</td>
<td>Pascal Context</td>
</tr>
<tr>
<td>previous best</td>
<td>88.7</td>
<td>69.3</td>
<td>60.7</td>
</tr>
<tr>
<td>InternImage-H</td>
<td>92.6 (+3.9)</td>
<td>71.7 (+2.4)</td>
<td>61.2 (+0.5)</td>
</tr>
</tbody>
</table>

| previous best | LVIS (minival) | LVIS (val) | VOC2007 | VOC2012 | OpenImages | CrowdHuman | BDD100K |
| InternImage-H | 59.8 | 62.2 | 89.3 | 92.9 | 72.2 | 94.1 | 35.6 |
| | 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) |


**D.2. Object Detection**

LVIS v1.0 [54] 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 [34] pre-trained weights, then fine-tune it on the training set of LVIS v1.0. Here, we report the box AP (i.e., AP<sub>b</sub>) with multi-scale testing on the minival set and the val set, respectively. As shown in Table E, our InternImage-H creates a new record of 65.8 AP<sub>b</sub> on the LVIS minival, and 63.2 AP<sub>b</sub> on the LVIS val, outperforming previous state-of-the-art methods by clear margins.

Pascal VOC [55] 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 [34] 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 [46]. As shown in Table E, on the Pascal VOC 2007 test set, our InternImage-H yields 94.0 AP<sub>50</sub> with single-scale testing, which is 4.7 points better than previous best Cascade Eff-B7 NAS-FPN [46]. 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 [47].

OpenImages v6 [56] 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.

**Figure B. 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 [38] with (w/) and without (w/o) training on ImageNet-1K [18], respectively. (c) and (d) are the ERF of InternImage-B with (w/) and without (w/o) training on ImageNet-1K.

**Stage 1**

(a) ResNet101 w/o training
(b) ResNet101 w/ trained model

**Stage 2**

(c) InternImage-S w/o training
(d) InternImage-S w/ trained model

widely 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 B. Our method achieves state-of-the-art accuracy of 71.7 (see Table E) on the validation set of Places205, outperforming the previous best model MixMIM-L [40] by 2.4 points.

Places365 [53] 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 E) on the validation set of Places365, outperforming the previous best model SWAG [41] 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.
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 [48] to use the class-aware sampling during fine-tuning. As reported in Table E, our InternImage-H yields 74.1 mAP, achieving 1.9 mAP improvement compared to the previous best results [48].

**CrownHuman** [57] 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 [49] by 3.1 AP.

**BDD100K** [58] 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 E, 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.

### D.3. Semantic Segmentation

**COCO-Stuff** [37] includes the images from the COCO [24] 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 [35], 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 E, our model achieves 59.6 MS mIoU on the test set, outperforming the previous best ViT-Adapter [36] by 5.4 mIoU.

**Pascal Context** [59] 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 [36]. 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 E, our method achieves 70.3 MS mIoU on the test set, which is 2.1 points better than ViT-Adapter [36].

### E. Throughput Analysis

In this section, we benchmark the throughput of our InternImage with counterparts, including a variant equipped with DCNv2 [65], ConvNext [10], RepLKNet [14], and a vision transformer with deformable attention (DAT) [66].
As shown in Table F, compared to the variant with DCNv2 [65], our model enjoys better parameter-efficient and significantly faster inference speed under both 224×224 and 800×800 input resolutions. Compared to RepLKNet-B [14] and DAT-B [66], 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 [10], 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. In Table G, we provide a full comparison of InternImage with Swin Transformer [3] in terms of throughput and memory, which shows that InternImage obtains better accuracy than Swin-L on various tasks with comparable inference efficiency.

F. Robustness Evaluation on ImageNet

In this section, we evaluate the robustness of different models under different transformations (see Fig. C). We consider translation, rotation, and scaling to evaluate. The models we choose for comparison include a convolutional model (ConvNeXt-T [10]), a local attention-based model (Swin-T [3]), a global attention-based model (PVTv2-B2 [6]), and our InternImage-T.

F.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. C indicates our InternImage has 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 [3].

F.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. C, we found that the consistency performance of all models is comparable in the small angle phase. However, at large-angle rotation (i.e., > 10°), our model is clearly superior to the other models.

F.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. C, we can observe that all methods of our experiments are sensitive to downscaling. 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.

F.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 [38], ConvNeXt-T [10], Swin-T [3], InternImage-T-dattn and our InternImage-T to conduct 300 rounds of training experiments on these data. The experimental settings are consistent with Table B. The experimental results can be viewed in Table H. We see that ResNet [38] performs best on the 1% and 10% data (12.2%
28

Table H. Accuracy of different models at different data scales. “InternImage-dattn” refers to the model variant equipped with deformable attention [67].

<table>
<thead>
<tr>
<th>method</th>
<th>1%</th>
<th>10%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [38]</td>
<td>12.2</td>
<td>57.5</td>
<td>80.4</td>
</tr>
<tr>
<td>ConvNeXt-T [10]</td>
<td>8.4</td>
<td>52.6</td>
<td>82.1</td>
</tr>
<tr>
<td>Swin-T [3]</td>
<td>failed</td>
<td>12.1</td>
<td>81.3</td>
</tr>
<tr>
<td>InternImage-T [67]</td>
<td>4.1</td>
<td>49.9</td>
<td>81.9</td>
</tr>
<tr>
<td>InternImage-T-dattn [67]</td>
<td>5.9</td>
<td>56.0</td>
<td>83.5</td>
</tr>
</tbody>
</table>

& 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 [10]. These results indicate the robustness of our model with respect to the data scale.

References


