Supplementary Materials for Revealing the Dark Secrets of Masked Image Modeling

Zhenda Xie*[13], Zigang Geng*[23], Jingcheng Hu[13], Zheng Zhang[3], Han Hu[3], Yue Cao[3†]
[1]Tsinghua University [2]University of Science and Technology of China [3]Microsoft Research Asia
{t-zhxie, t-ziganggeng, v-jingchu, zhez, hanhu, yuecao}@microsoft.com

A. Visualizations and Experimental Analysis on Vision Transformer

A.1. Visualizations with More Pre-training Methods

Figure 1, Figure 2 and Figure 3 present the visualizations of ViT-B as the backbone architecture across various pre-training methods. These pre-training methods encompass supervised pre-training (DeiT [36]), contrastive learning approaches (MoCo v3 [5], DINO [4], MSN [1]), masked image modeling (BEiT [2], MAE [15], SimMIM [40]), and hybrid methods (iBOT [44]). The visualizations reveal that different different contrastive learning methods (b) MoCo v3, (c) DINO and (d) MSN all exhibit similar representational characteristics to supervised pre-training (a) DeiT. Regarding attention distance, contrastive methods tend to focus locally in lower layers but more globally in higher layers. As for the diversity of attention heads, the contrastive methods lose diversity in deeper layers, with the last three layers showing minimal diversity. Similarly, various MIM methods (f) BEiT, (g) MAE and (h) SimMIM contribute comparable locality inductive bias and high attention head diversity to the model. These results reflect the consistency and universality of the visual analysis, showing the shared representational characteristics among pre-training methods within the same category. Notably, the hybrid pre-training approach (e) iBOT, which melds masked image modeling with contrastive learning, exhibits a blend of attention properties from both pre-training techniques. In the lower layers of the model, iBOT shows a similar behavior to contrastive learning methods, while in the higher layers, iBOT is closer to the behavior of MIM.

A.2. Experimental Results on Geometric and Motion Tasks

Based on the earlier visualizations, we further conduct experimental evaluations of various pre-trained models on geometric and motion tasks, such as pose estimation and depth estimation. As depicted in Table 1, the experimental results reveal that the performance of contrastive learning methods (MoCo v3, DINO, MSN) closely resembles that of supervised pre-training (DeiT), aligning with observations from the visualizations. While the hybrid pre-training method (iBOT) exhibits combined representational characteristics in visualizations, its performance in pose estimation and depth estimation tasks more closely aligns with contrastive learning methods. Among all pre-training techniques, masked image modeling methods (BEiT, MAE, SimMIM) achieve the highest performance, demonstrating a distinct performance advantage.

pre-train	Pose Estimation			Depth Estimation	
	COCO *val*	COCO *test*	Crowd Pose	NYUv2	KITTI
DeiT [36]	73.7	73.1	68.7	0.403	2.505
MoCo v3 [5]	73.6	72.8	67.7	0.408	2.564
DINO [4]	73.9	73.0	68.5	0.415	2.576
MSN [1]	74.0	73.3	68.6	0.420	2.563
iBOT [44]	73.8	73.2	68.4	0.383	2.506
BEiT [2]	75.4	74.8	71.6	0.365	2.324
MAE [15]	75.4	74.7	71.5	0.383	2.439
SimMIM [40]	75.3	74.9	71.8	0.349	2.287

Table 1. Comparisons of different pre-trained models on the geometric and motion tasks. We report the AP (↑) for the pose estimation tasks, and RMSE (↓) for the monocular depth estimation tasks.

B. Visualizations on Swin Transformer

It is crucial to know whether our observations in visualizations are general across different backbone architectures. Thanks to the general applicability of SimMIM [40], we further perform the visualizations on SwinV2-B [27] (in Section B) and RepLKNet [8] (in Section C). Fortunately, we find that most of the observations could be transferred across architectures, ViT-B, SwinV2-B, and RepLKNet.

*Equal Contribution. The work is done when Zhenda Xie, Zigang Geng, and Jingcheng Hu are interns at Microsoft Research Asia. † Contact person.

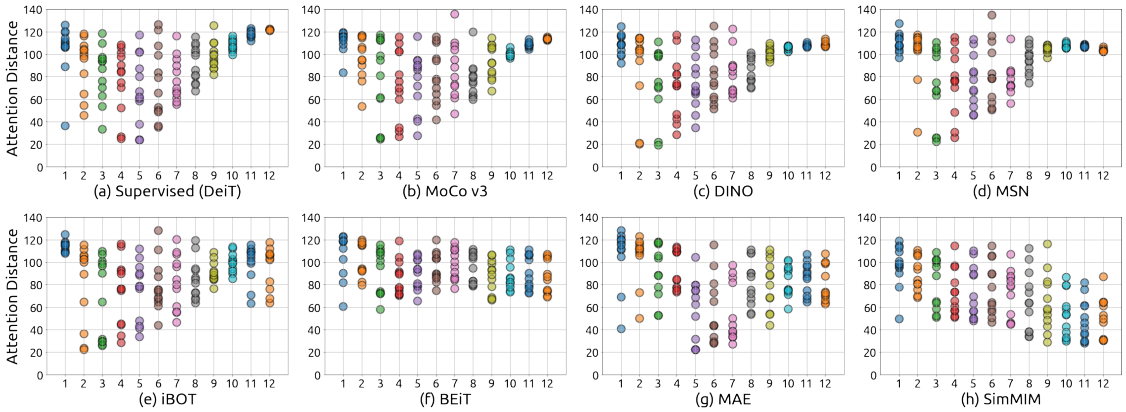

Figure 1. The averaged attention distance in different attention heads (dots) w.r.t the layer number on (a) supervised (DeiT), (b) MoCo v3, (c) DINO, (d) MSN, (e) iBOT, (f) BEiT, (g) MAE and (h) SimMIM model with ViT-B as the backbone.

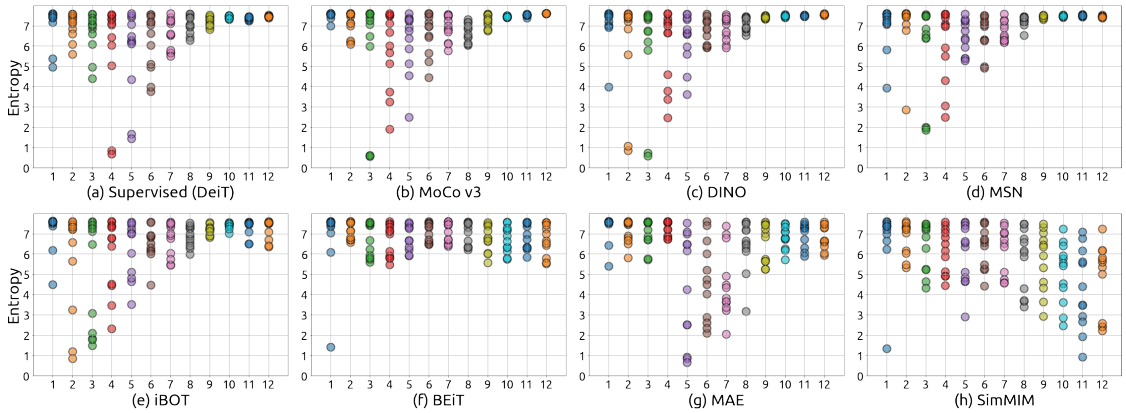

Figure 2. The entropy of each head's attention distribution w.r.t the layer number on (a) supervised (DeiT), (b) MoCo v3, (c) DINO, (d) MSN, (e) iBOT, (f) BEiT, (g) MAE and (h) SimMIM model with ViT-B as the backbone.

B.1. Visualizations on Attention Maps

Local Attention or Global Attention? Results are shown in Figure 4. First, we can have a similar observation as in ViT-B that the supervised model (a) tends to focus locally at lower layers but more globally at higher layers, and the SimMIM model (b) tends to aggregate both local and global pixels in all layers, and the average attention distance of SimMIM model is similar to the lower layers of the supervised counterpart. The supervised fine-tuned model (c) with SimMIM pre-training behaves very similarly to the supervised model trained from scratch, but still maintains some good properties in SimMIM pre-training (a larger diversity on the last several layers). Also, we find that the averaged aggregated distances in two consecutive layers are one high and one low. This is due to the shifted windowing scheme in Swin Transformer, that is, the ranges that each pixel can aggregate in two consecutive layers are different.

Focused Attention or Broad Attention? A similar observation could be found with Swin-B as the backbone as using ViT-B as the backbone in the main paper, as shown in Figure 5.

Diversity on Attention Heads As shown in Figure 6, similar to ViT-B, in SimMIM models (b), different attention heads tend to aggregate different tokens on all layers. But for supervised models (a), the diversity on attention heads becomes smaller as the layer goes deeper. Interestingly, after supervised fine-tuning the SimMIM model on ImageNet-1K, the model (c) behaves much more similarly to the supervised model (a) trained from scratch, but maintains an advantage of the SimMIM model, that is, a larger diversity on attention heads of the last two layers.

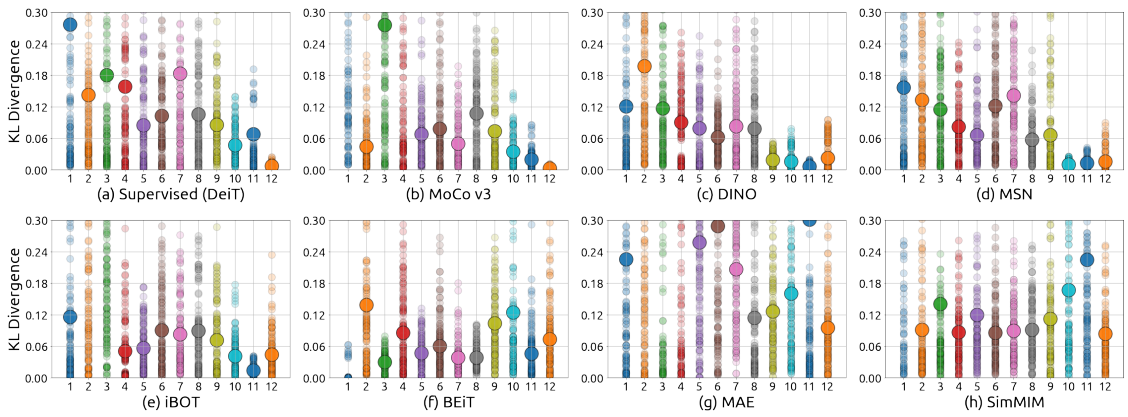

Figure 3. The KL divergence between attention distributions of different heads (small dots) and the averaged KL divergence (large dots) in each layer w.r.t the layer number on (a) supervised (DeiT), (b) MoCo v3, (c) DINO, (d) MSN, (e) iBOT, (f) BEiT, (g) MAE and (h) SimMIM model with ViT-B as the backbone.

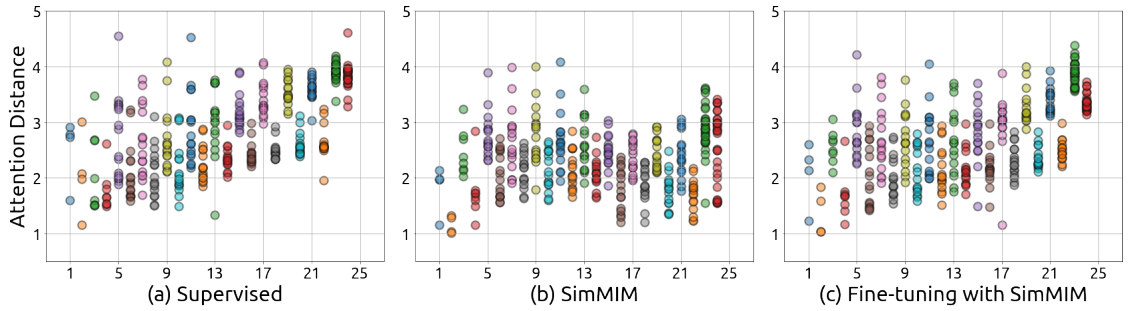

Figure 4. The averaged attention distance in different attention heads (dots) w.r.t the layer number on (a) supervised model, (b) SimMIM model, and (c) supervised fine-tuned model with SimMIM pre-training with SwinV2-B as the backbone.

B.2. Investigating the Representation Structures via CKA Similarity

It is challenging to analyze and compare the layer representations of deep networks, because their features are high-dimensional and with different dimensions. Centered kernel alignment (CKA) [22] is defined to address this challenge, and enables quantitative comparisons of feature representations within and across networks. Given two inputs of $X \in \mathbb{R}^{N \times D_1}$ and $Y \in \mathbb{R}^{N \times D_2}$, where N denotes number of examples and D_1 and D_2 denote the dimension. Then the Gram matrices are computed as $K = XX^T$ and $L = YY^T$. CKA is then defined as

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K)\text{HSIC}(L, L)}}, \qquad (1)$$

where $\text{HSIC}(\cdot,\cdot)$ denotes the Hilbert-Schmidt independence criterion [14]. Note that, CKA is invariant to the orthogonal transformation and isotropic scaling, which enables valuable and effective comparison and analysis on hidden representations of deep networks.

Results of CKA similarity between feature representations of different layers on (a) supervised model, (b) SimMIM model, and (c) supervised fine-tuned model with SimMIM pre-training with SwinV2-B as the backbone, are shown in Figure 7. We still have a similar observation as in ViT-B, that the representation structures of different layers in SimMIM models are almost the same, and supervised models trained from scratch learn different representation structures in different layers. With the help of the SimMIM pre-training, the representation structures of different layers in supervised model are not as different as that in the scratch supervised models.

C. Investigations on Large-kernel ConvNets (RepLKNet [8])

From the previous visualizations on Vision Transformers (ViT) and Swin Transformers, we find that the MIM pre-training brings the locality inductive bias and larger diversity on attention heads to the trained models comparing to the supervised counterpart, which may benefit the optimization of the trained models on downstream tasks. This reminds us that large-kernel ConvNets [8] without special

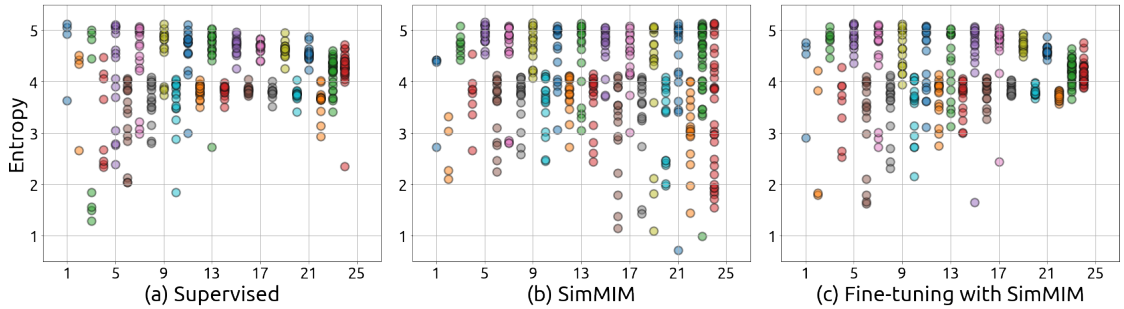

Figure 5. The entropy of each head's attention distribution in different attention heads (dots) w.r.t the layer number on (a) supervised model, (b) SimMIM model, and (c) supervised fine-tuned model with SimMIM pre-training with SwinV2-B as the backbone.

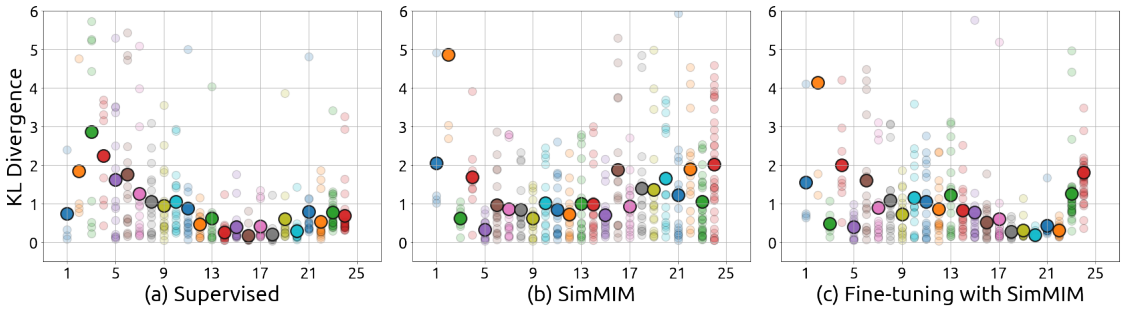

Figure 6. The KL divergence between attention distributions of different heads (small dots) and the averaged KL divergence (large dots) in each layer w.r.t the layer number on (a) supervised model, (b) SimMIM model, and (c) supervised fine-tuned model with SimMIM pre-training with SwinV2-B as the backbone.

designs still face the optimization issue, and need the re-parametrization trick with small kernels to bring the locality back and help them optimize. Thus it is valuable to know whether the masked image modeling (MIM) as pre-training could help the large-kernel ConvNets to optimize without the re-parametrization trick. Thanks to the general applicability of SimMIM [40], we could also perform experiments and visualizations on large-kernel ConvNets [8] with the MIM pre-training.

C.1. Experimental Results

Setup For MIM pretraining, we utilize the RepLKNet-31B [8] without the specially designed re-parametrization trick. Before the stem of the RepLKNet, using a normal 1×1 convolution, we map the 3-dimension space of the image into a high-dimensional space where we randomly mask out some patches. Following SimMIM [40], the image size is 192×192, we divide it into 6×6 patches and randomly mask out 60% patches. The decoder contains a linear projection layer and an upsample layer. We use ℓ_1-loss to supervise the reconstruction of the masked pixels.

We use the ImageNet-1k for MIM pre-training and augment the data using the random resize cropping (scale range $[0.67, 1]$ and aspect ratio range $[3/4, 4/3]$), and random flipping. The optimizer is the AdamW [29] optimizer with a weight decay of $5e$-2 and a base learning rate of $4e$-4. We use warm-up for 10 epochs, drop the learning rate to $4e$-5 at 260th epoch, and train for 300 epochs in total. The batch size is 2048. We use the DropPath of 0.1 for RepLKNet-31B and gradient clipping.

We report the top-1 accuracy of the supervised pre-trained model on ImageNet-1k in the original paper [8]. For fine-tuning of MIM pre-trained model on ImageNet-1k, we follow the setting of SimMIM [40] and use the AdamW optimizer with a weight decay of $5e$-2, a base learning rate of $5e$-3 with a layer decay of 0.8. The learning rate is scheduled via cosine strategy and we use 20 epochs for warm-up and train for 100 epochs in total. The batch size is 2048. We adopt the DropPath of 0.1 and gradient clipping. The data augmentations contain AutoAug [7], Mixup [42], CutMix [41], color jitter, random erasing [43], and label smoothing [35]. The settings of the pose estimation are the same as the details in Section F.

Results As shown in Table 2, the MIM pre-training can help the large-kernel convnets to address the optimization issue to some extent and achieve on par performance on ImageNet-1K compared with the supervised model with the re-parametrization trick. Note that, on pose estimation, MIM models still surpass supervised counterparts with the

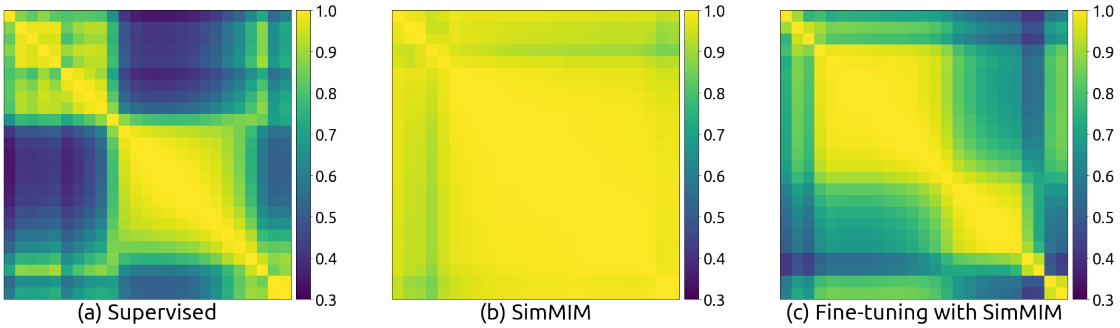

Figure 7. The CKA heatmap between the feature maps of different layers of (a) supervised model, (b) SimMIM model, and (c) supervised fine-tuned model with SimMIM pre-training with SwinV2-B as the backbone.

backbone	pre-train	ImageNet-1K	Pose Estimation		
			COCO *val*	COCO *test*	Crowd-Pose
RepLKNet-31B	1K-SUP w/ Reparam.	83.5	74.6	73.9	70.2
RepLKNet-31B	1K-MIM w/o Reparam.	83.3	76.5	75.8	72.4

Table 2. Detailed comparisons of pre-trained RepLKNet models on the classification and the pose estimation tasks. We report the top-1 accuracy (↑) for the ImageNet-1K dataset and the AP (↑) for the pose estimation tasks.

re-parametrization trick by large margins, which indicates that the benefit of MIM pre-training on geometric and motion tasks is general across different backbone architectures.

C.2. Visualizations

To further understand whether the behaviors of large-kernel ConvNets with MIM pre-training are similar to those of Vision/Swin Transformers, we visualize the convolutional kernels with similar tools used in visualizing the attention maps. As the basic component in RepLKNet is the depth-wise convolution with the kernel dimension of $C \times H \times W$, we normalize each channel of the depth-wise convolutional kernels (on the dimension of $H \times W$) to make them as a similar role of attention map, and regard different channels (C channels) of the depth-wise convolutional kernels as the attention heads. Then we could directly apply the previous tools on attention maps for visualizations.

Local Kernels or Global Kernels? As shown in Figure 8, with the re-parametrization trick, the RepLKNet-31B model (b) with supervised training focuses much more locally in all layers. Similar to previous supervised trained models, RepLKNet-31B models with supervised training still tend to focus locally at lower layers but more globally at higher layers. But for the model trained by SimMIM (c), each layer has diverse kernels that tend to aggregate both local and global pixels, and the average aggregated distance is much smaller than the supervised trained model without the re-parametrization trick (a), indicating that MIM still brings locality inductive bias to the large-kernel ConvNets with a similar role of the re-parametrization trick but less strength.

Focused Kernels or Broad Kernels? As shown in Figure 9, with the re-parametrization trick, the supervised RepLKNet-31B model (b) has very focused attention in lower layers, but broader attention in higher layers. But for the MIM model (c), the entropy values in different kernels focus diversely in all layers, that some kernels are more focused and some kernels have very broad attention. These observations well match that in the Vision/Swin Transformers.

Diversity across Different Kernels Interestingly, in Figure 8, it seems that the different kernels in both supervised model with the re-parametrization trick and SimMIM model have diverse averaged aggregated distance. But in Figure 10, we could clearly observe that the diversity on different convolution kernels of SimMIM model (c) is remarkably larger than that of supervised counterparts (b), especially for the deeper layers.

D. Detailed Results on Semantic Understanding Tasks

Detailed comparisons of Kornblith 12-dataset classification benchmark [23] and Concept Generalization (CoG) benchmark [32] with a fine-grained classification dataset iNaturalist-18 [37] using SwinV2-B as the backbone, are

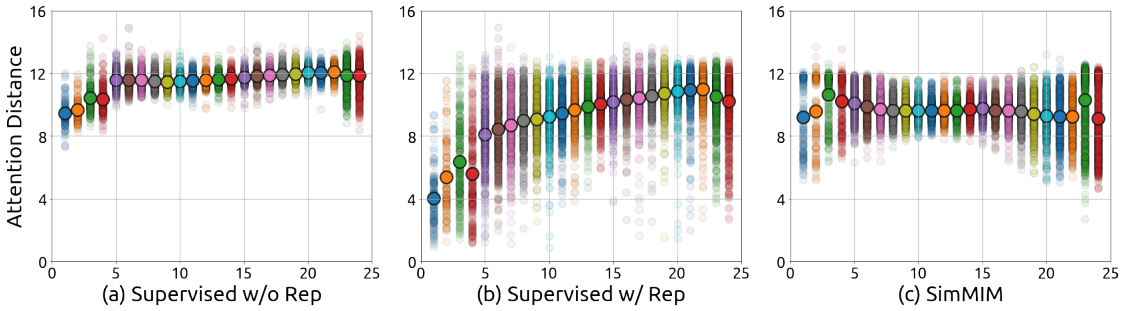

Figure 8. The aggregated distance in different channels (small dots) and the averaged aggregated distance (large dots) w.r.t the layer number on (a) supervised model without the re-parametrization trick, (b) supervised model with the re-parametrization trick, and (c) SimMIM model, with RepLKNet-31B as the backbone architecture.

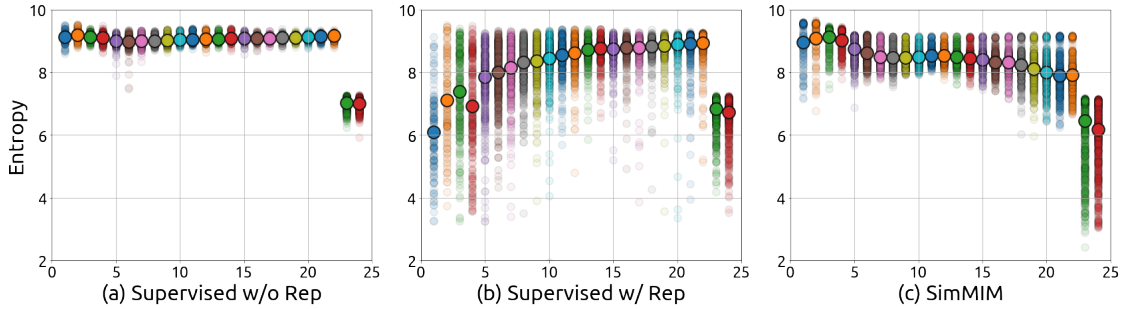

Figure 9. The entropy values in different channels (small dots) and the averaged entropy values (large dots) w.r.t the layer number on (a) supervised model without the re-parametrization trick, (b) supervised model with the re-parametrization trick, and (c) SimMIM model, with RepLKNet-31B as the backbone architecture.

shown in Table 3 and 4, respectively. These results are already discussed in Section 4.1 of the main paper.

E. Comparisons on Combined Task of Semantic Segmentation

We further select semantic segmentation on ADE-20K as another combine task which simultaneously performs both semantic understanding and geometric learning. For this task, we select two different frameworks, UperNet [39] and Mask2former [6] for evaluation. The detailed settings are shown in Section F.

Results are shown in Table 5. Different to COCO, we find that the supervised pre-trained model slightly outperforms the MIM counterpart on ADE-20K semantic segmentation. Therefore, for the combined tasks, it may be difficult to predict which pretrained model will perform better. But if the model gets larger, MIM models still have the unique advantage that MIM tasks are harder to be overfitted than supervised tasks [15, 40], which is beyond the scope of this paper. Also, we can observe that the performance gap between supervised and MIM models on Mask2former is smaller than that of UperNet (-1.6 v.s. -0.6). This may be due to that Mask2former decomposes the semantic segmentation task into object localization and recognition tasks, while MIM is better at object localization tasks, as shown in Figure 7 of the main paper.

F. Detailed Settings

Concept Generalization benchmark (CoG). The Concept Generalization benchmark (CoG) consists of five 1k-category datasets splitted from ImageNet-22K, which have increasing semantic gaps with ImageNet-1K, from L_1 to L_5. On the CoG datasets, for a fair comparison, we first fine-tune the models on the CoG L_1 training set and search for the best hyper-parameter based on the validation top-1 accuracy of CoG L_1, and then directly apply the searched setting to CoG L_2 to L_5 and report the top-1 accuracy. The detailed hyperparameters are shown in Table 6.

Kornlith et al's 12-dataset benchmark (K12) and iNaturalist-18 (iNat18). On the K12 dataset, we follow the previous standard settings [23] to use training set and validation set to search for the best hyper-parameters, and then merge the training and validation sets as the final training set with the searched best hyper-parameters, and evaluate the final trained models on the test set. And we adopt standard splits of train/val/test sets as in [23]. For Aircraft, Pets,

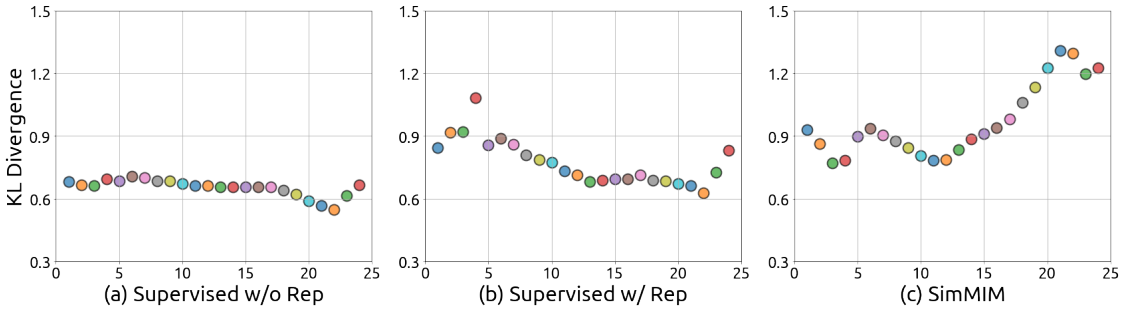

Figure 10. The averaged KL divergence in each layer w.r.t the layer number on (a) supervised model without the re-parametrization trick, (b) supervised model with the re-parametrization trick, and (c) SimMIM model, with RepLKNet-31B as the backbone architecture.

Methods	*Food101*	*Birdsnap*	*Stanford Cars*	*FGVC Aircraft*	*Oxford Pets*	*Caltech101*	*Flowers102*	*DTD*	*SUN397*	*CIFAR10*	*CIFAR100*
1K-SUP	93.2	81.7	88.6	83.0	95.9	91.9	97.7	80.3	72.3	99.1	91.0
1K-MIM	94.2	83.7	89.2	83.5	90.9	85.5	91.4	73.4	70.8	99.2	91.4

Table 3. Detailed comparisons of MIM and supervised (SUP) pre-trained models on Kornblith 12-dataset classification benchmark [23] with SwinV2-B as the backbone. We follow [23] to report top-1 accuracy (↑) and mean per-class accuracy (↑) for specific datasets. Results on the multi-label dataset Pascal Voc 2007 are not included, whose evaluation metric is not compatible with others.

pre-train	Concept Generalization (CoG)					iNat18
	L_1	L_2	L_3	L_4	L_5	
RAND	79.4	76.7	73.1	72.7	68.5	76.5
1K-SUP	79.4	76.2	72.7	72.5	68.4	77.7
1K-MIM	79.6	77.1	73.6	73.0	69.1	79.6

Table 4. Detailed comparisons of randomly initialized model (RAND), MIM and supervised (SUP) pre-trained models on Concept Generalization (CoG) benchmark [32] and a fine-grained classification dataset iNaturalist-18 [37] with SwinV2-B as the backbone. Top-1 accuracy (↑) is reported.

Caltech-101, Oxford 102 Flowers, the mean-per-class accuracy metric is adopted, for other datasets, the top-1 accuracy is adopted. For K12, we follow [23] to select the optimal learning rate, weight decay, layer decay, and drop path rate. In pilot experiments, we find that for 1K-SUP pre-trained models, the drop path rate can be fixed as 0.2, and for 1K-MIM pre-trained models, on smaller datasets like Stanford Cars, FGVC Aircraft, DTD, Caltech101, Flowers102, and Oxford Pets, drop path rate is first fixed as 0.0 and fixed as 0.2 for other datasets. And the weight decay can be fixed as 0.05. Then we do a grid search on learning rate and layer decay. For 1K-MIM pre-trained models, our grid consists of 5 approximately logarithmically spaced learning rates between $1.25e$-4 and $2.5e$-3 and 3 equally spaced layer decay between 0.75 and 0.95. For 1K-SUP pre-trained models, our grid consists of 5 approximately logarithmically spaced learning rates between $2.5e$-5 and $5e$-4 and 3 equally spaced layer decay between 0.75 and 0.95. Then we adjust the learning rate, layer decay, and drop path rate in the neighborhood of the best setting in the grid search to get the final results.

The iNat18 dataset includes 437,513 training images and 24,426 validation images, with more than 8,000 categories. The detailed hyperparameters of iNat18 are shown in Table 6.

Pose estimation. We compare the performance of MIM and supervised pre-trained models on the COCO [26] and CrowdPose [24] dataset. For the COCO dataset, We train the models on the $train2017$ set ($57K$ training images) and report the performance of the COCO $val2017$ split ($5K$ images), COCO $test\text{-}dev2017$ split ($20K$ images). For the CrowdPose dataset, following the DEKR [12], we train the models on the CrowdPose train and val sets ($12K$ training images) and evaluate on the test split ($8K$ images). The standard average precision based on OKS is adopted as the evaluation metric for all datasets.

We adopt the heatmap-based top-down pipeline. We upsample the last feature of the backbone by deconvolutions and predict the heatmaps at $4\times$ resolution like Simple Baseline [38].

In the ablation study on the number of the dropped layers

backbone	pre-train	Object Det. (COCO) Mask R-CNN AP^{box}	Mask R-CNN AP^{mask}	Semantic Seg. (ADE-20K) UperNet mIoU	Mask2former mIoU
SwinV2-B	1K-SUP	51.9	45.7	50.9	52.3
	1K-MIM	52.9	46.7	49.3 (−1.6)	51.7 (−0.6)

Table 5. Comparisons of MIM and supervised (SUP) pre-trained models on the combined tasks of object detection and semantic segmentation. We report the AP^{box} (↑) and AP^{mask} (↑) for the object detection and instance segmentation tasks, mIoU (↑) for the semantic segmentation task.

Hyperparameters	**RAND**		**1K-SUP**		**1K-MIM**	
	CoG (1-5)	**iNat18**	**CoG (1-5)**	**iNat18**	**CoG (1-5)**	**iNat18**
Input size	224					
Window size	14					
Patch size	4					
Training epochs	300	300	100	100	100	100
Warm-up epochs	20					
Layer decay	1.0	1.0	0.85	0.9	0.8	0.75
Batch size	2048					
Optimizer	AdamW					
Base learning rate	2e-3	4e-3	2e-4	1.6e-3	5e-3	1.6e-2
Weight decay	0.05	0.1	0.05	0.1	0.05	0.1
Adam ϵ	1e-8					
Adam β	(0.9, 0.999)					
Learning rate scheduler	Cosine					
Gradient clipping	5.0					
Stochastic depth	0.5	0.5	0.2	0.2	0.2	0.2
Label smoothing	0.1					
Rand crop scale	(0.08, 1)					
Rand resize ratio	(3. / 4., 4. / 3.)					
Rand horizontal flip	0.5					
Color jitter	0.4					
Rand augment	9 / 0.5					
Rand erasing prob.	0.25					
Mixup prob.	0.8					
Cutmix prob.	1.0					

Table 6. Detailed settings and hyperparameters for fine-tuning on CoG (1-5) and iNat18 with supervised and MIM pre-trained models.

in the section 3.1.3 of the main paper, we feed the feature at the different layers in the third stage of SwinV2-B into the pose head. We observe that when we use the feature at the ninth layer, the downstream performances of the supervised pre-trained model and MIM pre-trained model are almost comparable, so we use this model as the baseline of the experiments of randomly sampling pre-trained weights in the section 3.2 of the main paper. In the experiments of randomly sampling pre-trained weights, we randomly sample the weights of nine layers from the weights of the eighteen pre-trained layers in the third stage and then load them to the first nine layers.

The data augmentations include random flipping, half body transformation, random scale $(0.5, 1.5)$, random rotation $(-40°, 40°)$, grid dropout and color jitterring (h=0.2, s=0.4, c=0.4, b=0.4). The input image size is 256×256 by

default. We use the AdamW [29] optimizer with the base learning rate $5e$-4 and the weight decay $5e$-2. The learning rate is dropped to $5e$-5 at the $120th$ epoch. We totally train the models for 150 epochs. We use a layer decay of 0.9/0.85 for Swin-B/L and the DropPath [17] of 0.3/0.5 for Swin-B/L. The batch size is 512.

For the COCO dataset, we use the person detection results from the previous methods [34,38] for a fair comparison. For the CrowdPose dataset, we use a cascade mask-rcnn [3] with Swin-B backbone trained on the COCO detection dataset to generate the person detection results. We use the UDP [18] to reduce the quantization errors brought by the heatmaps and use flip testing by averaging the heatmaps predicted by the original and flipped images during the inference.

Depth estimation. We evaluate the performance of MIM and supervised pre-trained models on the NYUv2 [33] and KITTI [11] monocular depth estimation datasets. The NYUv2 dataset includes 464 indoor scenes captured by a Microsoft Kinect camera. The official training split ($24K$ images) is used for training and we report the RMSE (Root Mean Square Error) on the 654 testing images from 215 indoor scenes. The KITTI dataset contains various driving scenes. The Eigen split [9] contains $23K$ training images and 697 testing images. To compare with the previous approaches [20, 31], we set the maximum range as 10m for NYUv2 and 80m for KITTI.

The head of the depth estimation is the same as the head of the pose estimation and is comprised of three deconvolutions (with BN and ReLU) and a normal convolution. The kernel and filter of the deconvolution are 2 and 32, respectively.

Similar to the GLPDepth [20], we use the following data augmentations: random horizontal flip, random brightness (-0.2, 0.2), random gamma (-0.2, 0.2), random hue (-20, 20), random saturation (-30, 30), random value (-20, 20) and random vertical CutDepth. We randomly crop the images to 480×480 size for NYUv2 dataset and 352×352 size for KITTI dataset. The optimizer, layer decay, and DropPath is the same as the pose estimation. The learning rate is scheduled via polynomial strategy with a factor of 0.9. The minimal learning rate and the maximal learning rate are $3e$-5 and $5e$-4, respectively. The batch size is 24. The total number of epochs is 25. We use the flip testing and sliding window test for the SwinV2 backbone. We average the prediction of the two square windows for NYUv2 dataset and the sixteen square windows for KITTI dataset.

Video Object Tracking. Following the previous arts, we train the models on the train splits of four datasets GOT10k [19], TrackingNet [30], LaSOT [10], and COCO [26] and report the success score (SUC) for the TrackingNet dataset and LaSOT dataset. For the GOT10k test set, we report the average overlap as the evaluation metric. The GOT10k and the TrackingNet are two short-term large-scale benchmarks, the GOT10K test set contains 180 video sequences, and the TrackingNet test set contains 511 video sequences. The LaSOT is a long-term tracking benchmark and has 280 video sequences with an average length of about 2500 frames.

We use the SwinTrack [25] to evaluate our pre-trained models. The data augmentations and the training settings Strictly follow SwinTrack [25]. We sample 131072 pairs per epoch and train the models for 300 epochs. We use the AdamW optimizer with a learning rate of $5e$-4 for the head, a learning rate of $5e$-5 for the backbone, and a weight decay of $1e$-4. We decrease the learning rate by a ratio of 0.1 at the 210th epoch. We set the sizes of search images and templates as 224×224 and 112×112. The batch size is 160. The inference process is the same as the SwinTrack [25].

Object Detection. Following [40], we adopt a Mask-RCNN [16] framework to evaluate the pre-trained models on COCO object detection. All models are trained with a $3\times$ schedule (36 epochs). We utilize an AdamW [21] optimizer with a learning rate of 6e-5 for supervised model and a learning rate of 8e-5 for MIM model, a weight decay of 0.05 and a batch size of 32 for both models. Following [13, 28], we employ a large jittering augmentation (1024×1024 resolution, scale range [0.1, 2.0]). The window size is set to 14 for both models and drop path rate is set to 0.3 for supervised model and 0.1 for MIM model. AP^{box} and AP^{mask} are reported for comparison.

Semantic Segmentation. Following [28], an UPerNet [39] framework is used for ADE-20K semantic segmentation. We use an AdamW [21] optimizer with a learning rate of 8e-5 for supervised model and a learning rate of 1e-4 for MIM model, a weight decay of 0.05 and a batch size of 32 for both models. Both models utilize a layer-wise learning rate decay of 0.95. All models are trained for 80K iterations with an input resolution of 640×640 and a window size of 20. The drop path rate is set to 0.3 for supervised model and 0.1 for MIM model. In inference, a single-scale test using resolution of 2560×640 is employed.

Besides, we also adopt Mask2Former [6] to evaluate the pre-trained models on ADE-20K semantic segmentation. We use an AdamW [21] optimizer with a base learning rate of 1e-4 for supervised model and a base learning rate of 3e-4 for MIM model, a weight decay of 0.05 and a patch size of 16 for both supervised and MIM models. The learning rate of backbone is multiplied by a factor of 0.1. All models are trained for 160K iterations with an input resolution of 512×512, a scale ratio range from 0.5 to 2, a window size of 8, and a drop path rate of 0.3. In inference, the input resolution will be set to 2048×512. mIoU is reported for comparison for both UPerNet and Mask2Former.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin,

Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 456–473. Springer, 2022.

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

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

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

[6] 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.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. *arXiv preprint arXiv:2203.06717*, 2022.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[10] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5374–5383, 2019.

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[12] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14676–14686, 2021.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021.

[14] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

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

[18] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5699–5708, 2020.

[19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1562–1577, 2021.

[20] Doyeon Kim, Woonghyun Ga, Pyunghwan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[23] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[24] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Haoshu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10863–10872, 2019.

[25] Liting Lin, Heng Fan, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *arXiv preprint arXiv:2112.00995*, 2021.

[26] 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 *ECCV*, pages 740–755. Springer, 2014.

[27] 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. *arXiv preprint arXiv:2111.09883*, 2021.

[28] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[30] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision*, pages 310–327. Springer, 2018.

[31] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12159–12168, 2021.

[32] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9629–9639, 2021.

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[34] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[36] 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. PMLR, 2021.

[37] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[38] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision*, pages 472–487. Springer, 2018.

[39] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[40] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[43] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

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