A Appendix

A.1 Additional Ablation Results

Table A.1 is the ViT-B counterpart of Table 2 on backbone adaptation. The observations are similar to that of ViT-L: comparing with the baseline using no propagation ("none"), various propagation strategies show good gains.

Table A.2 presents Table 5 with additional details about FLOPs, parameters, and inference time, plotted in Figure 3.

Table A.3 is the ablation on pre-training strategies for LVIS. Similar to Table 4, MAE pre-training has large gains over supervised pre-training.

Figure A.1 is the LVIS counterpart of Figure 3. The trends are similar to those in COCO, while the gain of IN-21K supervised pre-training is larger because it significantly improves rare category detection in LVIS.

Figure A.2 is the RetinaNet [14] counterpart of Figure 3, showing the trade-off between accuracy and model size. Here, we evaluate ViTDet with a one-stage RetinaNet [14] detector head and compare it to using Swin and MViTv2 as hierarchical backbones, all without hyper-parameter tuning. Compared to using Mask R-CNN and Cascade R-CNN (Table 5 and Figure 3), we observe similar trends with RetinaNet. In particular, our plain-backbone detector presents *better scaling behavior* (*e.g.* ViT-H gains **+3.4** $\mathrm{AP}^{\mathrm{box}}$ over MViTv2-H). These results suggest that the proposed training recipe transfers well to different detectors and that our proposed plain backbone adaptations are general and can likely work with even more detection architectures.

A.2 Implementation Details

Architectures. We build a simple feature pyramid of scales $\{\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}\}$ (see Sec. 3). The $\frac{1}{32}$ scale is built by stride-2 2×2 max pooling (average pooling or convolution works similarly). The $\frac{1}{16}$ scale simply uses the ViT's final feature map. Scale $\frac{1}{8}$ (or $\frac{1}{4}$) is built by one (or two) 2×2 deconvolution layer(s) with stride=2. In the $\frac{1}{4}$ scale case, the first deconvolution is followed by LayerNorm (LN) [1] and GeLU [12]. Then for each pyramid level, we apply a 1×1 convolution with LN to reduce dimension to 256 and then a 3×3 convolution also with LN, similar to the per-level processing of FPN [13].

We study three detection frameworks: Mask R-CNN [11], Cascade Mask R-CNN [3] and RetinaNet [14]. For Mask R-CNN and Cascade Mask R-CNN, we incorporate some common best practices developed since they [11,3] were presented years ago. We use 2 hidden convolution layers for the RPN and 4 hidden convolution layers for the RoI heads as per [16]. These hidden convolution layers are followed by LN. For all three detection frameworks, We use the same detection implementation for both plain and hierarchical backbones.

We use a patch size of 16 for all ViT backbones. As ViT-H in [6] by default has a patch size of 14, after pre-training we interpolate the patch embedding filters from 14×14×3 to 16×16×3.

prop. strategy	AP^{box}	AP^{mask}
none	48.9	43.9
4 global blocks	**51.2** (+2.3)	**45.5** (+1.6)
4 conv blocks	51.0 (+2.1)	45.3 (+1.4)
shifted win.	50.1 (+1.2)	44.8 (+0.9)

(a) Window attention with various cross-window propagation strategies.

prop. conv	AP^{box}	AP^{mask}
none	48.9	43.9
naïve	50.6 (+1.7)	45.2 (+1.3)
basic	50.7 (+1.8)	45.2 (+1.3)
bottleneck	**51.0** (+2.1)	**45.3** (+1.4)

(b) Convolutional propagation with different residual block types (4 blocks).

prop. locations	AP^{box}	AP^{mask}
none	48.9	43.9
first 4 blocks	49.1 (+0.2)	44.1 (+0.2)
last 4 blocks	50.9 (+2.0)	45.4 (+1.5)
evenly 4 blocks	**51.2** (+2.3)	**45.5** (+1.6)

(c) Locations of cross-window global propagation blocks.

prop. blks	AP^{box}	AP^{mask}
none	48.9	43.9
2	50.7 (+1.8)	45.2 (+1.3)
4	**51.2** (+2.3)	**45.5** (+1.6)
12	50.4 (+1.5)	45.1 (+1.2)

(d) Number of global propagation blocks.

Table A.1: The ViT-B counterpart of Table 2 (backbone adaptation).

Hyper-parameters for COCO. Our default training recipe is as follows (unless noted in context for ablation). The input size is 1024×1024, augmented during training by large-scale jitter [7] with a scale range of $[0.1, 2.0]$. We use AdamW [15] (β_1, β_2=0.9, 0.999) with step-wise learning rate decay. We use linear learning rate warm-up [8] for 250 iterations. The batch size is 64, distributed across 64 GPUs (1 image per GPU).

We search for the learning rate (lr), weight decay (wd), drop path rate (dp), and epochs, for each model size (B, L, H) and for each model type (ViT, Swin, MViTv2). The hyper-parameters used are in Table A.4. We also use a layer-wise lr decay [4][2] of 0.7/0.8/0.9 for ViT-B/L/H with MAE pre-training, which has a small gain of up to 0.3 AP; we have not seen this gain for hierarchical backbones or ViT with supervised pre-training.

Hyper-parameters for LVIS. Our LVIS experiments in Table 7 follow the COCO settings in Table 5. For LVIS, we set lr = 2e^{-4}/1e^{-4} (ViT-L/H), wd = 0.1, and dp = 0.4. We fine-tune for 100 epochs. We use a test score threshold of 0.02 (smaller values did not help) and repeat factor sampling (t = 0.001) [9]. We output $\leq$ 300 detections per image following [9] (*vs*. COCO's default 100).

MAE for hierarchical backbones. We implement a naïve extension of MAE pre-training [10] for the hierarchical backbone ablation (Sec. 4.2). MAE enjoys the efficiency benefit from plain ViT by skipping the encoder mask token [10]. Extending this strategy to hierarchical backbones is beyond the scope of this paper. Instead, we adopt a straightforward solution in which we do not skip the encoder mask token (similar to [5]), at the cost of slower training. We use normalized pixels as the MAE reconstruction target [10] and set the decoder depth as 2.

backbone	pre-train	Mask R-CNN AP^{box}	AP^{mask}	FLOPs	params	time	Cascade Mask R-CNN AP^{box}	AP^{mask}	FLOPs	params	time
hierarchical-backbone detectors:											
Swin-B	1K, sup	50.1	44.5	0.7T	109M	60ms	52.7	45.5	0.9T	139M	76ms
Swin-B	21K, sup	51.4	45.4	0.7T	109M	60ms	54.0	46.5	0.9T	139M	76ms
Swin-L	21K, sup	52.4	46.2	1.1T	218M	81ms	54.8	47.3	1.4T	248M	96ms
MViTv2-B	1K, sup	52.4	46.7	0.6T	73M	82ms	54.7	47.5	0.8T	103M	97ms
MViTv2-L	1K, sup	53.2	47.1	1.3T	239M	173ms	55.2	47.7	1.6T	270M	189ms
MViTv2-B	21K, sup	53.1	47.4	0.6T	73M	82ms	55.6	48.1	0.8T	103M	97ms
MViTv2-L	21K, sup	53.6	47.5	1.3T	239M	173ms	55.7	48.3	1.6T	270M	189ms
MViTv2-H	21K, sup	54.1	47.7	2.9T	688M	338ms	55.8	48.3	3.2T	718M	353ms
our plain-backbone detectors:											
ViT-B	1K, MAE	51.6	45.9	0.8T	111M	77ms	54.0	46.7	1.1T	141M	92ms
ViT-L	1K, MAE	55.6	49.2	1.9T	331M	132ms	57.6	49.8	2.1T	361M	149ms
ViT-H	1K, MAE	**56.7**	**50.1**	3.4T	662M	189ms	**58.7**	**50.9**	3.6T	692M	203ms

Table A.2: Detailed measurements of Table 5 and Figure 3.

pre-train	ViT-B AP^{box}	AP^{mask}	AP^{mask}_{rare}	ViT-L AP^{box}	AP^{mask}	AP^{mask}_{rare}
IN-1K, supervised	37.2	34.9	26.4	38.3	36.0	26.7
IN-21K, supervised	38.7	36.3	28.8	42.1	39.5	34.3
IN-1K, MAE	**40.1**	**38.1**	**29.1**	**46.1**	**43.5**	**35.3**

Table A.3: The LVIS counterpart of Table 4 (COCO pre-training ablation). The observations are similar to Table 4: MAE pre-training has large gains over supervised pre-training. Here we also report rare category results. We observe that both IN-21K supervised and IN-1K MAE pre-training significantly improve AP^{mask}_{rare}, especially for ViT-L. (Mask R-CNN, 1024 resolution, no soft-nms.)

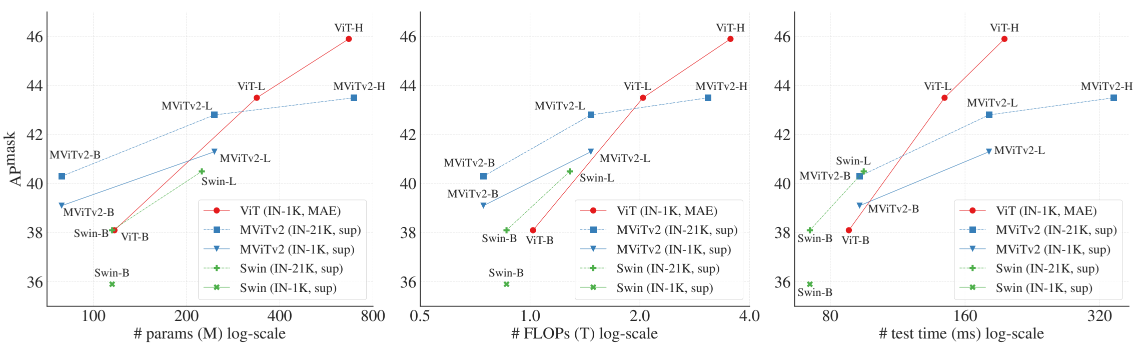

Figure A.1: The LVIS counterpart of Figure 3. All entries are implemented and run by us to align low-level details. Here the detector head is Mask R-CNN (input resolution 1024; no soft-nms). The trends are similar to those in Figure 3, while IN-21K supervised pre-training has larger gains.

backbone	pre-train	*lr*	*wd*	*dp*	epochs
ViT-B/L	none	$1.6e^{-4}$	0.2	0.1/0.4	300/200
ViT-B/L	supervised	$8e^{-5}$	0.1	0.1/0.4	50
ViT-B/L/H	MAE	$1e^{-4}$	0.1	0.1/0.4/0.5	100/100/75
Swin-B/L	supervised	$1e^{-4}/8e^{-5}$	0.05	0.3	50
MViTv2-B/L/H	supervised	$8e^{-5}$	0.1	0.4/0.5/0.6	100/50/36

Table A.4: Hyper-parameters for COCO. Multiple values in a cell are for different model sizes. The epochs are chosen such that training longer starts to overfit.

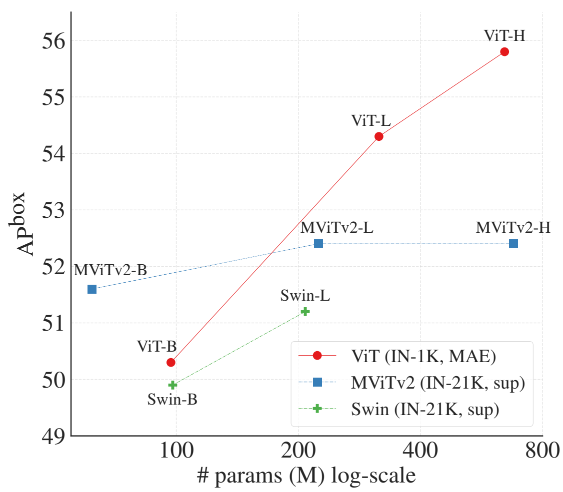

Figure A.2: The RetinaNet [14] counterpart of Figure 3, showing the trade-off between accuracy and model size. We use the same Mask R-CNN training recipe (input resolution 1024; no soft-nms) and hyper-parameters for RetinaNet. The trends are similar to those in Figure 3.

Acknowledgement. We would like to acknowledge Xinlei Chen, Saining Xie, Piotr Dollár, and Christoph Feichtenhofer for discussions and support.

References

1. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.
2. Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image Transformers. *arXiv:2106.08254*, 2021.
3. Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *TPAMI*, 2019.
4. Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *NAACL*, 2019.
6. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
7. 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 *CVPR*, 2021.
8. Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv:1706.02677*, 2017.
9. Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.
10. Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.
11. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.
12. Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GeLUs). *arXiv:1606.08415*, 2016.
13. Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.
14. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
15. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
16. Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.