Attention to the Burstiness in Visual Prompt Tuning!

Supplementary Material

Yuzhu Wang[1] Manni Duan[1] Shu Kong[2,3,✉]

[1]Zhejiang Lab [2]University of Macau [3]Institute of Collaborative Innovation

project webpage

Outline

As elaborated in the main paper, we uncover a "burstiness" phenomenon and non-Gaussian distributions in the values resulting from the interaction of the key projector, query projector, and image patch embeddings within the Transformer's self-attention module. We address these issues with several proposed methods called Bilinear Prompt Tuning (BPT). Experiments demonstrate that all our BPT methods significantly accelerate learning, reduce parameter count and computation, and importantly achieves the state-of-the-art over various benchmark datasets across a range of model scales, dataset sizes, and pre-training objectives. We provide more implementation details and additional results in the supplemental document.

A. Implementation Details

Evaluation datasets. We follow the practice of VPT [7] to perform the split of train/val/test for 5 FGVC datasets. Table 6 summarizes the details of the evaluated datasets used in the paper. Moreover, we sample a subset (*e.g.* randomly sampling 10% data for each category) of training data from ImageNet dataset [2] to study the affects of different training data size.

ViT architectures. We use the standard ViT [3] architectures that have a stack of Transformer blocks [17]. Each block consists of a multi-head self-attention layer and an MLP layer with LayerNorm [9]. Refer to Table 7 for details about the models.

Whitening matrix $\mathbf{W}$ and bilinear factor $\mathbf{B}$ are implemented using a 1×1 convolution layer. We *do not* use normalizations in-between or after their multiplication of the learned prompt $\mathbf{P}$. We tested applying normalizations but this decreases accuracy.

MAE pre-training does not use [CLS] token [5]. We follow the original designs and treat global average pooling on the sequence of $[\mathbf{P};\mathbf{X}] \in \mathbb{R}^{(n+m)\times d}$ as input for the classification head. We observe that using $[\mathbf{P};\mathbf{X}]$, $\mathbf{P}$ or $\mathbf{X}$ yields similar accuracy.

Object detection and instance segmentation. For object detection and segmentation tasks, we follow the influential ViTDet [10], which uses pre-trained ViT as backbone. We use the ViT's final feature map (16-stride, prompt tokens are discarded) to build a simple feature pyramid [10]. We remove the window attention modules [12, 17] as the backbone is frozen during prompt tuning, which allows the object detector to be directly adapted high-resolution input images without concerning about reaching memory limits or slowing down training speed. We use two hidden convolution layers for Region Proposal Networks [15] and 4 hidden convolution layers for the RoI heads as per [10, 20]. These hidden con-

Table 6. Specifications of the downstream-task datasets. We follow the practice of VPT [7] to split train/val/test for the five FGVC datasets. In addition, we also study general image classification, object detection, and instance segmentation tasks on the popular ImageNet and COCO datasets.

Datasets	Description	# Classes	Train	Val	Test
Fine-grained visual recognition tasks (FGVC)					
CUB-200 [18]	bird classification	200	5,394	600	5,794
NABirds [16]	bird classification	555	21,536	2,393	24,633
Flowers [13]	flower classification	102	1,020	1,020	6,149
Dogs [8]	dog classification	120	10,800	1,200	8,580
Cars [4]	car classification	196	7,329	815	8,041
ImageNet [2]	general classification	1,000	1,281,167	50,000	-
COCO [11]	object det. and seg.	80	118,287	5,000	-

Table 7. Model architectures.

arch.	Layers	Patch size	Embed dim	MLP size	Heads	Params
ViT-Base	12	16	768	3,072	12	86M
ViT-Large	24	16	1,024	4,096	16	307M
ViT-Huge	32	14	1,280	5,120	16	632M
ViT-2B	24	14	2,560	10,240	32	1.89B

Table 8. Hyper-parameters used on ImageNet and COCO. Multiple values in a cell are for different model sizes. Here, *lr*, *wd* and *dp* stand for learning rate, weight decay, and drop path rate, respectively. Full fine-tuning also use a layer-wise learning rate decay.

Methods	batch	*lr*	*wd*	*dp*	epochs
ImageNet					
Full fine tuning	1024	4e-3/1e-2(2B)	0.05	0.1/0.1/0.2/0.3	100/50/50/35
BPT	1024	0.1/0.2/0.2/0.3	0.01	0	100
COCO					
Full fine tuning	16	1e-4	0.1	0.1	37
BPT	16	5e-4	0.1	0.0	37

Table 9. Results of Fig.**??**. Experiments of scale backbones and epochs use 10% of the ImageNet-1K's training images.

	backbone				training data					training epochs			
Methods	ViT-B	ViT-L	ViT-H	ViT-2B	1%	3%	10%	30%	100%	100	200	300	400
Full fine-tuning	56.80	69.46	74.43	76.82	27.68	43.46	56.80	68.91	76.39	56.80	-	-	-
SPT-Shallow [19]	63.64	73.77	76.23	77.61	44.35	55.52	63.64	67.17	69.98	63.64	64.37	64.54	64.61
BPT-Shallow	64.63	75.23	77.97	79.80	45.43	56.79	64.63	68.92	72.15	64.63	64.79	64.93	65.05

Table 10. A study of which Transformer blocks used to insert prompt for the deep version of BPT. Here, "3 + 1" means insert prompts into the first 3 blocks and the last block.

4 + 0	3 + 1	2 + 2	1 + 3	0 + 4	interval
76.83	78.45	81.66	81.38	**82.00**	80.87

volution layers are followed by LayerNorm [9]. The training last for 3× schedule.

Hyper-parameters. We search for the learning rate (*lr*), weight decay (*wd*), drop path rate (*dp*), and epochs for each model size (B, L, H, 2B) in each downstream task. The hyper-parameters used for ImageNet and COCO with MAE pre-training are in Table 8.

BPT-Deep is derived by straightforwardly extending BPT-shallow to more Transformer blocks, similar to VPT-Deep [7] and SPT-Deep [19], and yields remarkable performance improvements over shallow variant. However, BPT-Deep introduces more learned parameters, 7.54× more than BPT-Shallow. To reduce parameters, we study a *partial prompt-tuning* protocol: only insert prompts in the *last* Transformer blocks, *e.g.*, 6 or 4. This protocol was also used in other visual tuning works [5, 14, 21].

We observe that the layers at which prompts are inserted have a significant impact. Table 10 is a comparison and evaluated on CUB-200. As our default settings for deep variant, learning prompts in the last 4 blocks can achieve accuracy close to that of learning all blocks. This phenomenon is similar to that of partially fine-tuning deep neural networks that fine-tuning the last few layers can achieve accuracy close to Full fine-tuning [1, 5, 6].

We also study an *interval* sampling: we split the pre-trained backbone into 4 subsets of blocks (*e.g.*, 3 in each subset for the 12-block ViT-B). We insert prompts in the first block of each subset. This strategy is reasonably good: it has 80.87% accuracy, 3.0 higher than the shallow variant, but lags behind our default settings.

B. Additional Results

Table 9 is the scale-up counterpart of Fig. 5. All ViT backbones are self-supervised pre-trained by MAE [5] and report the top-1 accuracy on ImageNet val-set.

Table 11 presents pre-task results on 5 FGVC datasets, with ImageNet-21K supervised pre-trained ViT-B backbone.

Table 11. Per-task results on FGVC benchmarks of Table **??**, with supervised pre-trained ViT-B/16 backbone.

Methods	CUB-200	NABirds	Flowers	Dogs	Cars
Full fine-tuning	87.3	82.7	98.8	89.4	84.5
Linear probing	85.3	75.9	97.9	86.2	51.3
VPT-S [7]	86.7	78.8	98.4	90.7	68.7
SPT-S [19]	90.2	85.1	99.5	89.3	86.4
BPT-S (ours)	90.1	86.2	99.6	89.4	87.4
VPT-D [7]	88.5	84.2	99.0	90.2	83.6
SPT-D [19]	90.6	87.6	99.8	89.8	89.2
BPT-D (ours)	90.5	88.1	99.9	90.1	89.9

Fig. 6 compares detection and segmentation results of our BPT-shallow and SPT-Shallow [19] on COCO. SPT exhibits systematic artifacts on overlapping instances. Our BPT shows no such artifacts.

Fig. 7 and Fig. 8 display distributions of entries of $\mathbf{W}_q\mathbf{W}_k^T$ and $\mathbf{W}_q\mathbf{W}_k^T\mathbf{X}^T$, respectively. We see non-Gaussian distributions and burstiness (especially in the first Transformer block) regardless how to pretrain the backbone (*e.g.*, MAE, MoCO-V3, or ImageNet-21K supervised learning).

C. Code and Demo

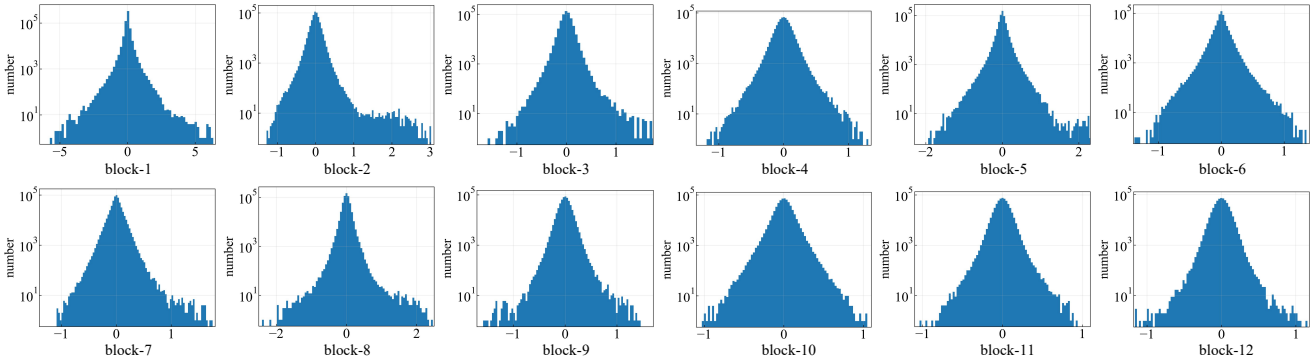
Code. We include our self-contained codebase (refer to the zip file `Code-BPT`) as a part of the supplementary material. Please refer to `README.md` for instructions how to use the code. We do not include model weights in the supplementary material as they are too large (>200MB) that exceed the space limit. We will open source our code and release our trained models to foster research.

License. We release open-source code under the MIT License to foster future research in this field.

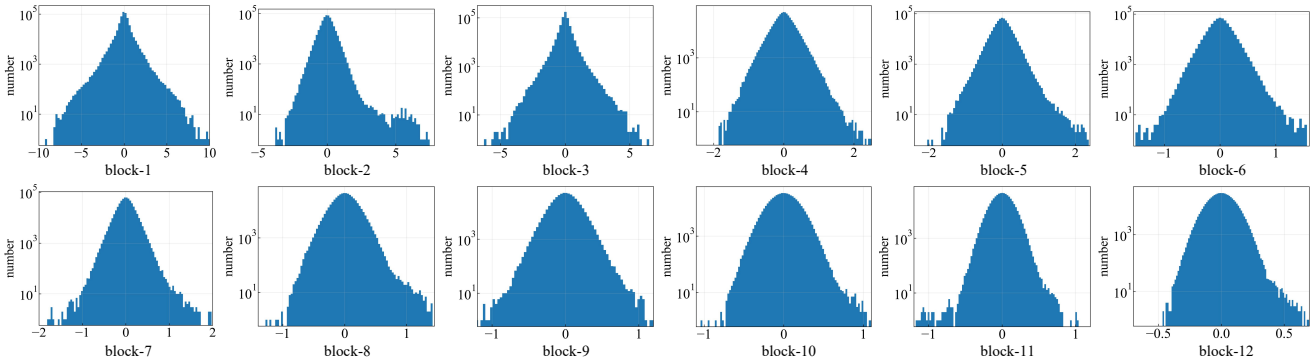
Requirement. Running our `Python` code requires some common packages, such as PyTorch, TorchVision, and timm. Please refer to `Code-BPT/README.md` for more details.

Demo. We use Jupyter Notebook to create three demos, including plot histogram, evaluate the image classification accuracy of our BPT models and visualize the results of detection and segmentation. See `demo-BPT-dis.ipynb`, `demo-BPT-eval.ipynb`, and `demo-BPT-det.ipynb` for more details.

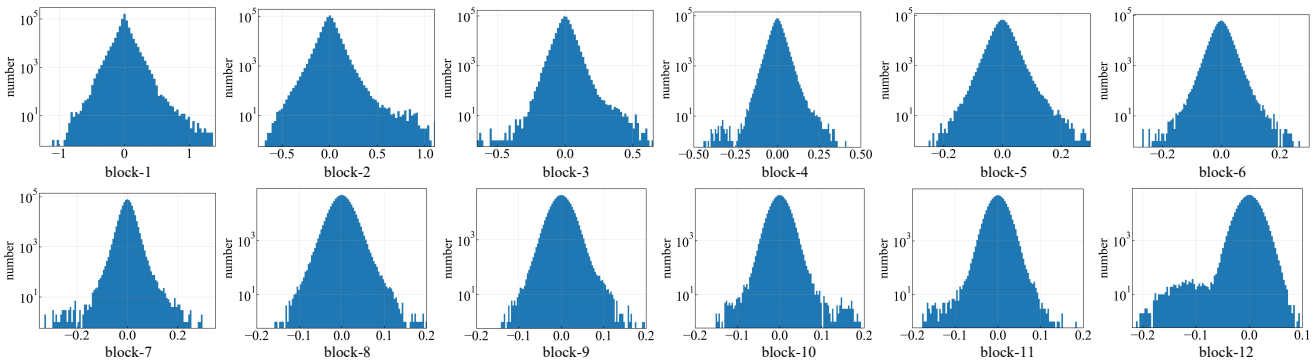
Figure 6. SPT [19] *vs.* BPT on COCO validation images. Here, MAE pre-trained ViT-B with Cascade Mask R-CNN as detector. SPT exhibits systematic artifacts on overlapping instances (marked by red arrow).



(a) MAE pre-training

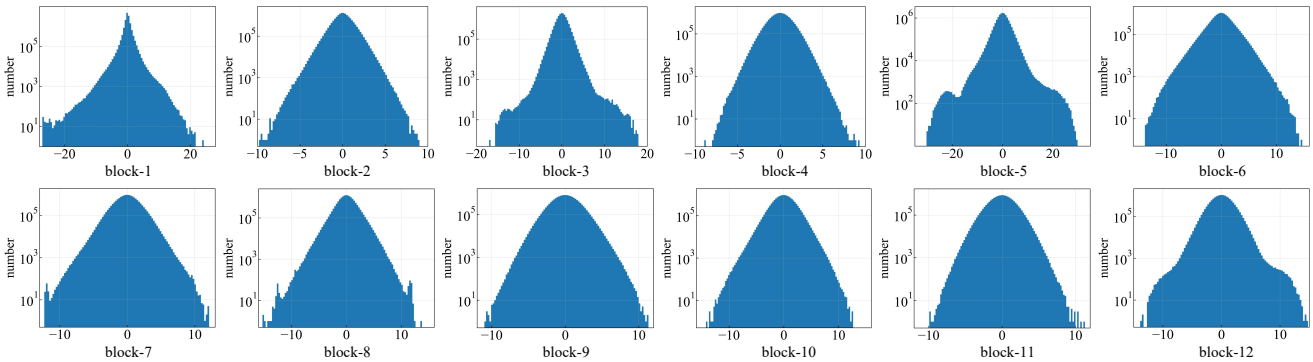


(b) MoCo-V3 pre-training

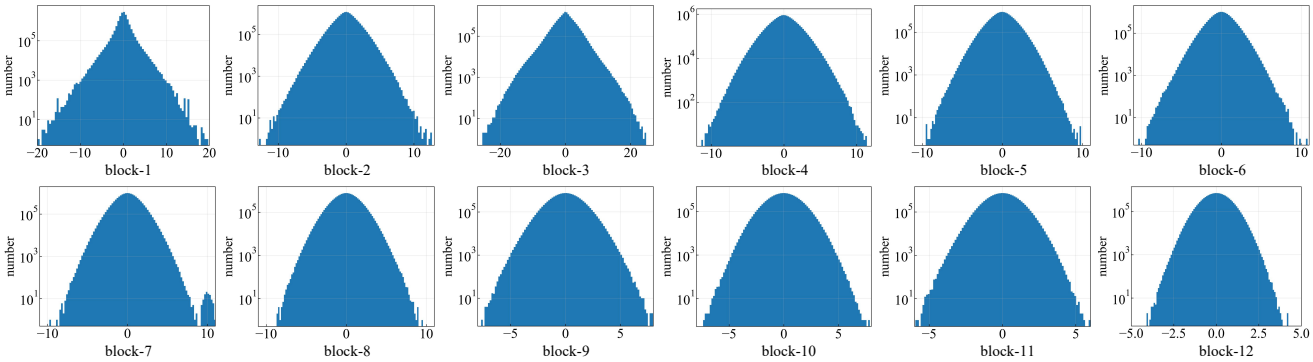


(c) ImageNet-21K supervised pre-training

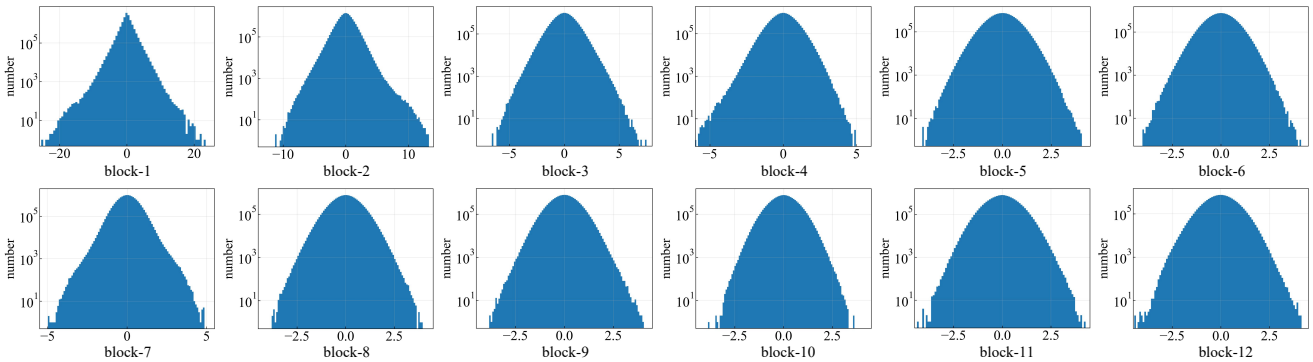
Figure 7. The distribution of $\mathbf{W}_q\mathbf{W}_k^T$ w.r.t Transformer block depth. The backbone is ViT-B (12 blocks) and is pre-trained with three different pretraining methods (a-c). The three pre-training methods consistently show non-Gaussian distributions and burstiness, especially in the first blocks.



(a) MAE pre-training



(b) MoCo-V3 pre-training



(c) ImageNet-21K supervised pre-training

Figure 8. The distribution of $\mathbf{W}_q\mathbf{W}_k^T\mathbf{X}^T$ w.r.t Transformer block depth. The backbone is ViT-B (12 blocks) and is pre-trained with three different pretraining methods (a-c). The image tokens $\mathbf{X}$ undergo normalizations as default implemented in typical Transformers. We observe non-Gaussian distributions of these values, and the burstiness (especially in the first block regardless of training methods) which means relatively few entries have much larger values.

References

[1] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 1

[3] 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 *International Conference on Learning Representations (ICLR)*, 2021. 1

[4] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 1

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

16000–16009, 2022. 1, 2

[6] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[7] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, pages 709–727. Springer, 2022. 1, 2

[8] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*. Citeseer, 2011. 1

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 1, 2

[10] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision (ECCV)*, pages 280–296. Springer, 2022. 1

[11] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[12] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[13] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 1

[14] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, pages 69–84. Springer, 2016. 2

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 1

[16] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015. 1

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[18] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011. 1

[19] Yuzhu Wang, Lechao Cheng, Chaowei Fang, Dingwen Zhang, Manni Duan, and Meng Wang. Revisiting the power of prompt for visual tuning. In *International Conference on Machine Learning (ICML)*, 2024. 2, 3

[20] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 1

[21] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014. 2