Scaling Vision Transformers

Xiaohua Zhai*, Alexander Kolesnikov*, Neil Houlsby, Lucas Beyer*
Google Research, Brain Team, Zürich
{xzhai, akolesnikov, neilhoulsby, lbeyer}@google.com

Abstract

Attention-based neural networks such as the Vision Transformer (ViT) have recently attained state-of-the-art results on many computer vision benchmarks. Scale is a primary ingredient in attaining excellent results, therefore, understanding a model's scaling properties is a key to designing future generations effectively. While the laws for scaling Transformer language models have been studied, it is unknown how Vision Transformers scale. To address this, we scale ViT models and data, both up and down, and characterize the relationships between error rate, data, and compute. Along the way, we refine the architecture and training of ViT, reducing memory consumption and increasing accuracy of the resulting models. As a result, we successfully train a ViT model with two billion parameters, which attains a new state-of-the-art on ImageNet of 90.45% *top-1 accuracy. The model also performs well for few-shot transfer, for example, reaching* 84.86% *top-1 accuracy on ImageNet with only 10 examples per class.*

1. Introduction

Attention-based Transformer architectures [44] have taken computer vision domain by storm [7, 15] and are becoming an increasingly popular choice in research and practice. Previously, Transformers have been widely adopted in the natural language processing (NLP) domain [6, 14]. Optimal scaling of Transformers in NLP was carefully studied in [21], with the main conclusion that large models not only perform better, but do use large computational budgets more efficiently. However, it remains unclear to what extent these findings transfer to the vision domain, which has several important differences. For example, the most successful pre-training schemes in vision are supervised, as opposed to unsupervised pre-training in the NLP domain.

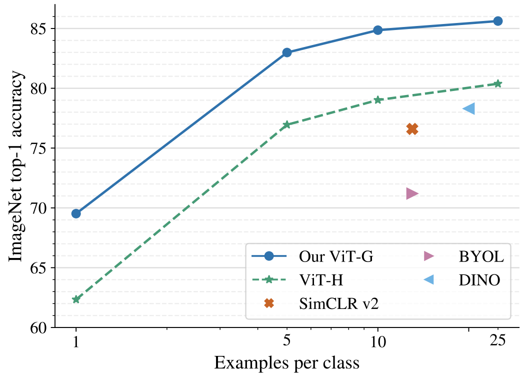

Figure 1. Few-shot transfer results. Our ViT-G model reaches 84.86% top-1 accuracy on ImageNet with 10-shot linear evaluation.

In this paper we concentrate on scaling laws for transfer performance of ViT models pre-trained on image classification tasks. In particular, we experiment with models ranging from five million to two billion parameters, datasets ranging from one million to three billion training images and compute budgets ranging from below one TPUv3 core-day to beyond 10 000 core-days. Our main contribution is a characterization of the performance-compute frontier for ViT models, on two datasets.

Along the way, we create an improved large-scale training recipe. We investigate training hyper-parameters and discover subtle choices that make drastic improvements in few-shot transfer performance. The few-shot transfer evaluation protocol has also been adopted by previous large-scale pre-training efforts in NLP domain [5]. Specifically, we discover that very strong L2 regularization, applied to the final linear prediction layer only, results in a learned visual representation that has very strong few-shot transfer capabilities. For example, with just a single example per class on the ImageNet dataset (which has 1 000 classes), our best model achieves 69.52% accuracy; and with *10 examples per class it attains 84.86%*. In addition, we substantially reduce the memory footprint of the original ViT model proposed in [15]. We achieve this by hardware-specific architecture changes and a different optimizer. As a result, we train a model with two billion parameters and attain a new *state-of-the-art 90.45% accuracy on ImageNet*.

*equal contribution

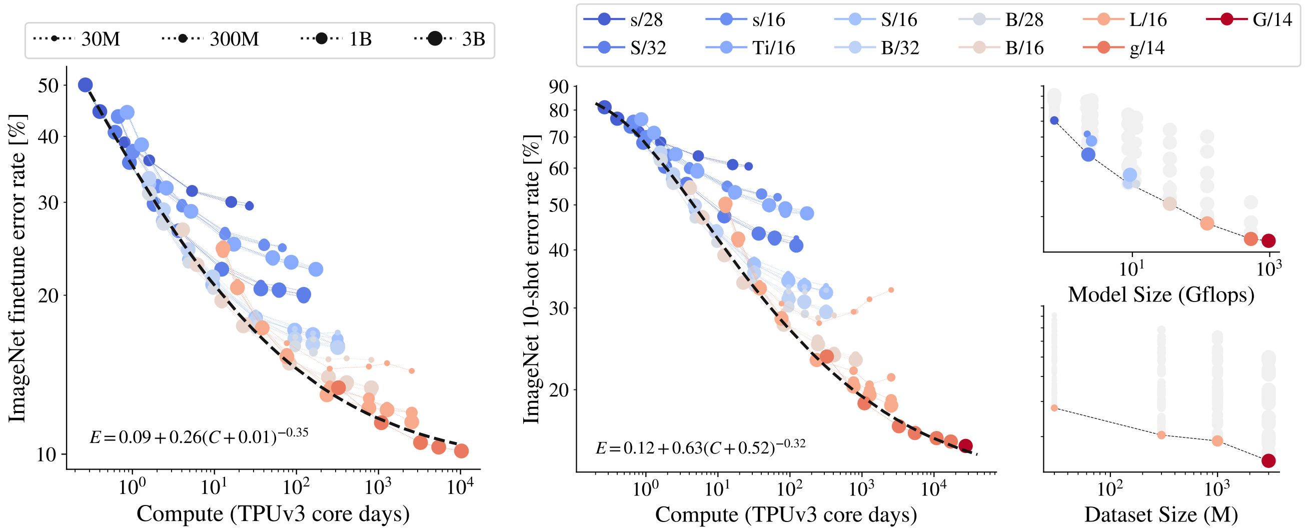

Figure 2. **Left/Center**: Representation quality, measured as ImageNet finetune and linear 10-shot error rate, as a function of total training compute. A saturating power-law approximates the Pareto frontier fairly accurately. Note that smaller models (blue shading), or models trained on fewer images (smaller markers), saturate and fall off the frontier when trained for longer. **Top right**: Representation quality when bottlenecked by model size. For each model size, a large dataset and amount of compute is used, so model capacity is the main bottleneck. Faintly-shaded markers depict sub-optimal runs of each model. **Bottom Right**: Representation quality by datasets size. For each dataset size, the model with an optimal size and amount of compute is highlighted, so dataset size is the main bottleneck.

2. Core Results

We first present our main results on scaling trends, before presenting detailed architecture and training protocol improvements in Section 3. In the following experiments, we train several ViT models on both public ImageNet-21k [13] dataset and privately gathered images, up to three billion weakly-labelled images. We vary the architecture size, number of training images, and training duration. All models are trained on TPUv3, thus total compute is measured in TPUv3 core-days. To evaluate the quality of the representation learned by the models, we measure (i) few-shot transfer via training a linear classifier on frozen weights, (ii) transfer via fine-tuning the whole model on all data, both to multiple benchmark tasks.

2.1. Scaling up compute, model and data together

Figure 2 shows both the 10-shot linear evaluation and finetuning evaluation on ImageNet [13]. Similar trends on other datasets, Oxford IIIT Pets [27], CIFAR-100 [23], and Caltech-UCSD Birds [46] are presented in the Appendix, Figure 9. For each combination of model size and data size we pre-train for various numbers of steps. In Figure 2, connected points represent the same model trained for a different number of steps. We make the following observations.

First, *scaling up compute, model and data together improves representation quality*. In the left plot and center plot, the lower right point shows the model with the largest size, dataset size and compute achieving the lowest error rate. However, it appears that at the largest size the models starts to saturate, and fall behind the power law frontier (linear relationship on the log-log plot in Figure 2).

Second, *representation quality can be bottlenecked by model size*. The top-right plot shows the best attained performance for each model size. Due to limited capacity, small models are not able to benefit from either the largest dataset, or compute resources. Figure 2, left and center, show the Ti/16 model tending towards a high error rate, even when trained on a large number of images.

Third, *large models benefit from additional data, even beyond 1B images*. When scaling up the model size, the representation quality can be limited by smaller datasets; even 30-300M images is not sufficient to saturate the largest models. In Figure 2, center, the error rate of L/16 model on the the 30M dataset does not improve past 27%. On the larger datasets, this model attains 19%. Further, when increasing the dataset size, we observe a performance boost with big models, but not small ones. The largest models even obtain a performance improvement the training set size grows from 1B to 3B images (Figure 2, bottom right). For small models, however, such as Ti/16 or B/32, increasing the dataset size does not help. For example, in Figure 2, left and center, all of the curves for Ti/16 overlap, showing that this model achieves the same performance irrespective of the dataset size.

2.2. Double-saturating power law

Figure 2, left and center, show the Pareto frontier of representation quality versus training compute. The frontier contains the models with the best allocation of compute to model shape and training duration.

For over two orders of magnitude of compute, the relation-

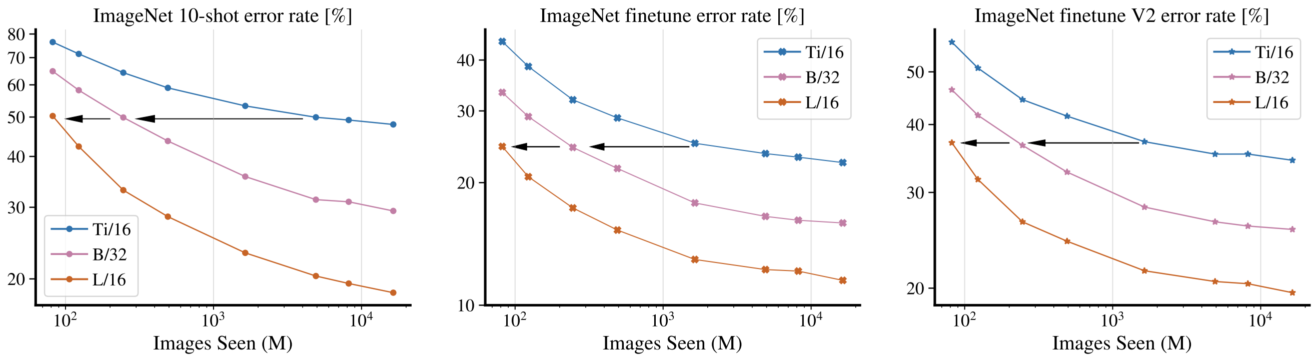

Figure 3. Error rate on ImageNet, with respect to images seen during pre-training. Big models are more sample efficient, which is consistent across diverse setups: few-shot transfer on the frozen representations, fine-tune the network on ImageNet, and evaluate the fine-tuned models on the v2 test set.

ship between compute and performance follows a power-law ($E = aC^b$), resulting in a straight line on the log-log plot. However, we observe "saturation" at both ends of the compute spectrum. At the higher end of compute, the largest models do not tend towards zero error-rate. If we extrapolate from our observations, an infinite capacity model will obtain a non-zero error. This effect has also been observed for generative models [18]. The authors of [18] refer to this residual error as the "irreducible entropy" of the task. Since we plot error rate, the information-theoretic interpretation does not apply, but our observations support the notion of fundamental performance ceilings for ImageNet [4]. In terms of the law, this saturation corresponds to an additive constant to the error rate: c in $E = aC^{-b} + c$.

At the lower end of the compute spectrum, we see a saturation for smaller models; the performance of the smallest model is better than that would be predicted by a power-law. This saturation occurs because even trivial solutions can achieve non-zero error. For example, predicting the majority class (almost zero compute) will achieve an accuracy related to its occurence frequency in the test set. This lower bound is not observed in [18], either because their smallest model is large enough to avoid this region, or because log-loss saturates at worse performances than accuracy (it will saturate eventually). This saturation corresponds to a shift in the x-axis: d in $E = a(C + d)^{-b} + c$. This constant indicates that the zero-compute model will still obtain non-zero accuracy.

2.3. Big models are more sample efficient

Figure 3 shows the representation quality with respect to the total number of images "seen" (batch size times number of steps) during pre-training. In addition to ImageNet fine-tuning and linear 10-shot results on the public validation set, we also report results of the ImageNet fine-tuned model on the ImageNet-v2 test set [32] as an indicator of robust generalization. Three ViT models pre-trained on three billion images are presented in this plot.

We observe that *bigger models are more sample efficient*, reaching the same level of error rate with fewer seen images. For 10-shot, the Ti/16 model needs to see nearly 100 times more images to match the representation quality of the L/16 model. When fine-tuning, this factor reduces from 100 to about 20. Our results suggest that with sufficient data, training a larger model for fewer steps is preferable. This observation mirrors results in language modelling and machine translation [21, 25].

2.4. Do scaling laws still apply on fewer images?

We extend the study to much fewer images, ranging from one million to 13 millions on the public ImageNet-21k

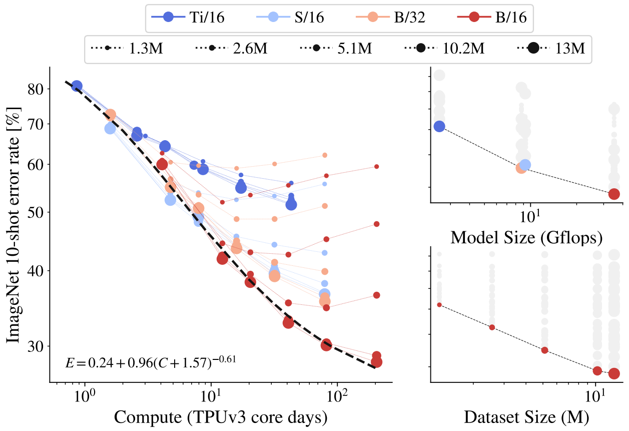

Figure 4. Results on the ImageNet-21k dataset. **Left**: Representation quality, measured as ImageNet linear 10-shot error rate, as a function of total training compute. The double-saturating power law still applies. **Right**: Representation quality by model sizes and dataset sizes.

Table 1. The results for ViT-G/14, compared to the previous state-of-the-art models.

Benchmark	**ImageNet**	**INet V2**	**INet ReaL**	**ObjectNet**	**VTAB (light)**
NS (Eff.-L2) [48]	88.3	80.2	-	68.5	-
MPL (Eff.-L2) [28]	90.2	-	**91.02**	-	-
CLIP (ViT-L/14) [30]	85.4	75.9	-	**72.3**	-
ALIGN (Eff.-L2) [20]	88.6	70.1	-	-	-
BiT-L (ResNet) [22]	87.54	-	90.54	58.7	76.29
ViT-H/14 [15]	88.55	-	90.72	-	77.63
Our ViT-G/14	**90.45±0.03**	**83.33±0.03**	90.81±0.01	70.53±0.52	**78.29±0.53**

dataset. In Figure 4 left, we found that the double-saturation power law *still applies*, when varying model sizes, dataset sizes and compute resources. This indicates that the conclusions from the study generalizes well, and can guide future design choices for vision transformer architectures. In Figure 4 right, we observe similar behaviors that the model performance are bottlenecked by the dataset size. When scaling up compute, model and data together, one gets the best representation quality.

2.5. ViT-G/14 results

We trained a large Vision Transformer, ViT-G/14, which contains nearly two billion parameters. Section 3.6 details the architecture's shape. We evaluate the ViT-G/14 model on a range of downstream tasks, and compare it to recent state-of-the-art results. We fine-tune on ImaegNet, and report ImageNet [33], ImageNet-v2 [32], ReaL [4], and ObjectNet [2] accuracies. In addition, we report transfer learning result on the VTAB-1k benchmark consisting of 19 tasks [52].

Figure 1 shows the few-shot transfer results on ImageNet. ViT-G/14 outperforms the previous best ViT-H/14 model [15] by a large margin (more than 5%), attaining *84.86% accuracy with 10 examples per class*. Ten images per class is less than 1% of ImageNet data (13 examples per class), as commonly used in self-supervised and semi-supervised learning [51]. For reference, Figure 1 shows three state-of-the-art self-supervised learning models, SimCLR v2 [9] and BYOL [16], using 1% of ImageNet data, DINO [8] using 20 examples per class. Note, however, that these approaches are quite different: ViT-G/14 uses large source of weakly-supervised data, and is pre-trained only once and transferred to different tasks. Meanwhile, the self-supervised learning models use unlabeled but in-domain data for pre-training, and target a single task.

Table 1 shows the results on the remaining benchmarks. ViT-G/14 achieves *90.45% top-1 accuracy on ImageNet*, setting the new state-of-the art. On ImageNet-v2, ViT-G/14 improves 3% over the Noisy Student model [48] based on EfficientNet-L2. For ReaL, ViT-G/14 outperforms ViT-H [15] and BiT-L [22] by only a small margin, which indicates again that the ImageNet classification task is likely reaching its saturation point. For ObjectNet, ViT-G/14 outperforms BiT-L [22] by a large margin, and is 2% better than Noisy Student, but is about 2% behind CLIP [30]. Note that, unlike the other methods, CLIP does not fine-tune on ImageNet, and evaluates directly on ObjectNet, this likely improves its robustness. Finally, when transferring the ViT-G/14 model to VTAB, it gets consistently better results with just a single hyper parameter across all tasks. The state-of-the-art on VTAB using a heavyweight per-task hyperparameter sweep is 79.99 [20], we leave running a heavy sweep with ViT-G/14 to future work.

3. Method details

We present a number of improvements to the ViT model and training. These improvements are mostly simple to implement, and can significantly improve memory-utilization and model quality. They allow us to train ViT-G/14 using data-parallelism alone, with the entire model fitting on a single TPUv3 core.

3.1. Decoupled weight decay for the "head"

Weight decay has a drastic effect on model adaptation in the low-data regime. We conduct an study of this phenomena at a mid-size scale.

We find that one can benefit from decoupling weight decay strength for the final linear layer ("head"), and for the remaining weights ("body") in the model. Figure 5 demonstrates this effect: we train a collection ViT-B/32 models on JFT-300M, each cell corresponds to the performance of different head/body weight decay values. The diagonal corresponds to using the same value for both decays. One can observe that the best performance appears off-diagonal (i.e. with a decoupled weight decay for the head and body). Interestingly, we observe that high weight decay in the head decreases performance on the pre-training (upstream) task (not shown), despite improving transfer performance.

We do not have a complete explanation of this phenomena. However, we hypothesize that a stronger weight decay in the

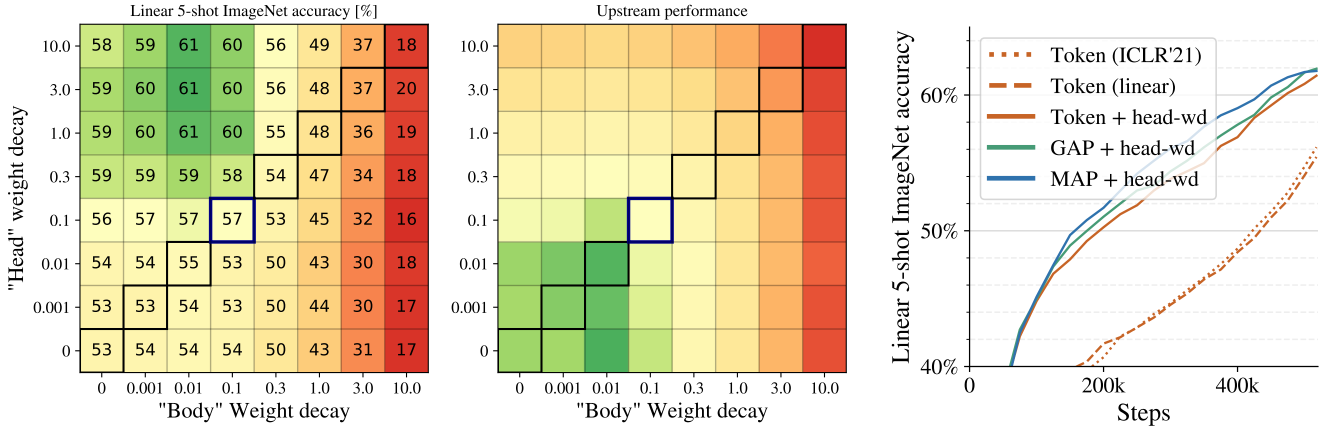

Figure 5. **Left and middle**: The dependence of 5-shot ImageNet accuracy and upstream performance depends on the weight decay strength. Normally, a single weight decay value is applied to all weights (corresponds to the diagonal on the heatmaps). We show that by using weight decay values for the "head" and the rest of the weights one significantly improves few-shot transfer performance. **Right**: Few-shot performance on ImageNet for different types of head. A high weight decay on the head works equally well for all of them.

head results in representations with larger margin between classes, and thus better few-shot adaptation. This is similar to the main idea behind SVMs [11]. This large decay makes it harder to get high accuracy during upstream pre-training, but our main goal is high quality transfer.

3.2. Saving memory by removing `[class]` token

The largest VIT model from [15] uses 14 × 14 patches with 224 × 224 images. This results in 256 visual "tokens", where each one corresponds to an image patch. On top of this, ViT models have an extra `[class]` token, which is used to produce the final representation, bringing the total number of tokens to 257.

For ViT models, current TPU hardware pads the token dimension to a multiple of 128, which may result in up to a 50% memory overhead. To overcome this issue we investigate alternatives to using the extra `[class]` token. In particular, we evaluate global average pooling (GAP) and multihead attention pooling (MAP) [24] to aggregate representation from all patch tokens. We set the number of heads in MAP to be equal to the number of attention heads in the rest of the model. To further simplify the head design we remove final non-linear projection before the final prediction layer, which was present in the original ViT paper.

To choose the best head, we perform a side-by-side comparison of a `[class]` token and GAP/MAP heads. Results are summarized in Figure 5 (right). We find that all heads perform similarly, while GAP and MAP are much more memory efficient due to the aforementioned padding considerations. We also observe that non-linear projection can be safely removed. Thus, we opt for the MAP head, since it is the most expressive and results in the most uniform architecture. MAP head has also been explored in [41], in a different context for better quality rather than saving memory.

3.3. Scaling up data

For this study, we use the proprietary JFT-3B dataset, a larger version of the JFT-300M dataset used in many previous works on large-scale computer vision models [15,22,36]. This dataset consists of nearly 3 billion images, annotated with a class-hierarchy of around 30k labels via a semi-automatic pipeline. Thus, the data and associated labels are noisy. We ignore the hierarchical aspect of the labels and use only the assigned labels as targets for multi-label classification via a sigmoid cross-entropy loss, following [15, 22].

We have conducted sensitive category association analysis as described in [1]. We measured (per label) the distribution of sensitive categories across the raw data, the cleaned data, the models trained on this data, and labels that were verified by human raters. Human raters additionally assisted in removing offensive content from the dataset.

Figure 6 shows an ablation of the effect of changing from JFT-300M to JFT-3B on model performance, even when scale is not increased. Figure 6, left shows linear 10-shot ImageNet performance evaluated throughout. We observe that JFT-3B results in a better model, even before the model has completely one epoch of JFT-300M. Therefore, overfitting JFT-300M is not the sole cause of the improvement. This difference can be seen even for the small B/32 model as well as the larger L/16. We fine-tune the models to the full ImageNet dataset (right), and confirm that these improvements transfer to a full fine-tuning setup. Overall, the change in dataset improves transfer to ImageNet by about 1% for both small and large models. Other than the performance improvement, training behavior is similar on JFT-300M and JFT-3B. Most importantly, JFT-3B allows us to scale up further with fewer concerns about overfitting and regularization.

Deduplication. We remove all images from the JFT-3B

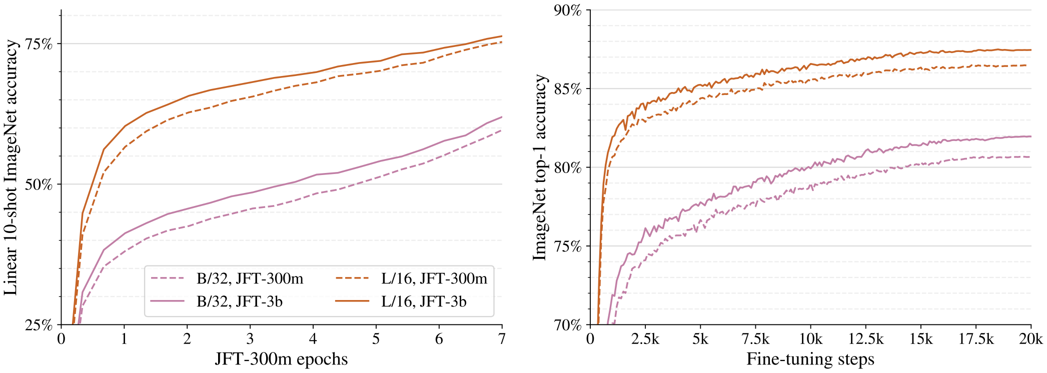

Figure 6. The effect of switching from JFT-300M to JFT-3B, without any further scaling. Both small and large models benefit from this change, by an approximately constant factor, both for linear few-shot evaluation (**left**) and transfer using the full dataset (**right**).

dataset that are near-duplicates of images from both train set and test set of datasets we evaluate on. Overall we identified and removed 927k duplicate images from JFT-3B.

3.4. Memory-efficient optimizers

When training large models, storage required for model parameters becomes a bottleneck. Our largest model, ViT-G, has roughly two billion parameters, which occupies 8 GiB of device memory. To make things much worse, the Adam optimizer that is commonly used for training Transformers, stores two additional floating point scalars per each parameter, which results in an additional two-fold overhead (extra 16 GiB). To tackle the overhead introduced by the Adam optimizer we explore two modifications.

Adam with half-precision momentum. We empirically observe that storing momentum in half-precision (`bfloat16` type) does not affect training dynamics and has no effect on the outcome. This allows to reduce optimizer overhead from 2-fold to 1.5-fold. Notably, storing the second momentum using half-precision resulted in a significant performance deterioration.

Adafactor optimizer. The above optimizer still induces a large memory overhead. Thus, we turn our attention to the Adafactor optimizer [34], which stores second momentum using rank 1 factorization. From practical point of view, this results in the negligible memory overhead. However, the Adafactor optimizer did not work out of the box, so we make the following modifications:

- We re-introduce the first momentum in half-precision, whereas the recommended setting does not use the first momentum at all.
- We disable scaling of learning rate relative to weight norms, a feature that is part of Adafactor.
- Adafactor gradually increases the second momentum from 0.0 to 1.0 throughout the course of training. In our preliminary experiments, we found that clipping the second momentum at 0.999 (Adam's default value) results in better convergence, so we adopt it.

The resulting optimizer introduces only a 50% memory overhead on top the space needed to store model's parameters.

We observe that both proposed optimizers perform on par with or slightly better than the original Adam optimizer. We are aware of other memory-efficient optimizers [31, 39], we leave the exploration to future work.

3.5. Learning-rate schedule

In our study we want to train each of the models for several different durations in order to measure the trade-off between model size and training duration. When using linear decay, as in [15], each training duration requires its own training run starting from scratch, which would be an inefficient protocol.

Inspired by [26], we address this issue by exploring learning-rate schedules that, similar to the *warmup* phase in the beginning, include a *cooldown* phase at the end of training, where the learning-rate is linearly annealed toward zero. Between the warmup and the cooldown phases, the learning-rate should not decay too quickly to zero. This can be achieved by using either a constant, or a reciprocal square-root schedule for the main part of training. Figure 7 (bottom) depicts several of these options, with a cooldown after approximately 200 k, 400 k, and 500 k steps. The upper half of Figure 7 shows the validation score (higher is better) for each of these options and their cooldowns, together with two linear schedules for reference. While the linear schedule is still preferable when one knows the training duration in advance and does not intend to train any longer, all three alternatives come reasonably close, with the advantage of allowing indefinite training *and* evaluating multiple training durations from just one run. For each of the schedules, we optimized the learning-rate and the exact shape. We have also briefly

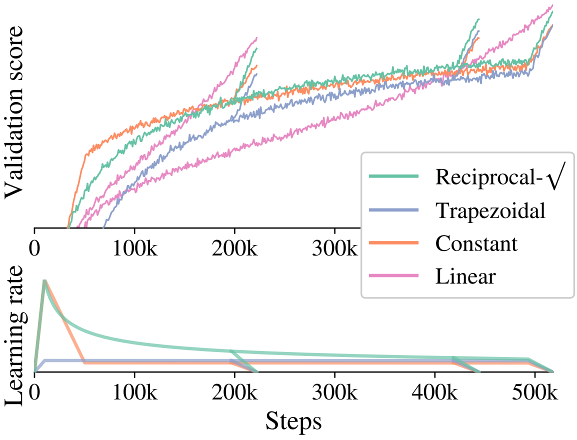

Figure 7. Various "infinite" learning-rate schedules, along with the finite linear one for reference.

tried cyclic learning-rate schedules, however they seemed to perform much worse and we have not investigated further. We therefore opt for the reciprocal square-root schedule.

3.6. Selecting model dimensions

ViT models have many parameters that control the model's shape, and we refer to the original publication for full details. Briefly, these include the *patch-size*, the number of encoder blocks (*depth*), the dimensionality of patch embeddings and self-attention (*width*), the number of attention *heads*, and the hidden dimension of MLP blocks (*MLP-width*). On top of this, we rely on the XLA compiler to optimize our models for runtime speed and memory footprint. Behind the scenes, XLA uses complex heuristics to compile a model into code for a specific hardware that trades off memory and speed optimally. As a result, it is hard to predict which model configurations will fit into memory on a single device.

Therefore we run an extensive simulation, where we instantiate a large amount of ViTs of various shapes, and attempt to train them for a few steps, without considering the quality. We vary the depth, width, heads, and MLP-width, but keep the patch-size at 14 px. In this way, we measure their speed and whether or not a given model fits into the device's memory. Figure 8 summarizes the result of this simulation. Each block corresponds to one model configuration, the shade of the block corresponds to its training speed (brighter is faster). Orange blocks show which original ViT models, without any of our modifications, fit. Green blocks then further include the memory savings described in Section 3.2 coupled with the half-precision Adam described in Section 3.4. Finally, blue blocks are with our modified AdaFactor optimizer. The shapes in the white area were not able to fit into memory in any setting. For space reasons, we show here only the models pertaining to the experiments presented, but note that with our modifications we were able to fit thin ViT models of a depth up to 100 encoder blocks.

Table 2. Model architecture details.

Name	**Width**	**Depth**	**MLP**	**Heads**	**Mio. Param**	**GFLOPs**	
						224^2	384^2
s/28	256	6	1024	8	5.4	0.7	2.0
s/16	256	6	1024	8	5.0	2.2	7.8
S/32	384	12	1536	6	22	2.3	6.9
Ti/16	192	12	768	3	5.5	2.5	9.5
B/32	768	12	3072	12	87	8.7	26.0
S/16	384	12	1536	6	22	9.2	31.2
B/28	768	12	3072	12	87	11.3	30.5
B/16	768	12	3072	12	86	35.1	111.3
L/16	1024	24	4096	16	303	122.9	382.8
g/14	1408	40	6144	16	1011	533.1	1596.4
G/14	1664	48	8192	16	1843	965.3	2859.9

The original Vision Transformer publication contains a study in Appendix D2 about the trade-offs between scaling the different components, concluding that it is most effective to scale all aspects (depth, width, MLP-width, and patch-size) simultaneously and by a similar amount. We follow this recommendation, and select shapes for ViT-g and ViT-G at the limit of what fits in memory accordingly, as shown in Figure 8 and summarized in Table 2.

4. Related Work

Smaller Vision Transformers Early work on Transformers for vision focused on small networks for CIFAR-10 [10]. The Vision Transformer [15], however, was proposed in the context of state-of-the-art medium and large-scale image recognition; the smallest model (ViT-B) containing 86M parameters. [40] present smaller ViT sizes for training from-scratch, down to ViT-Ti, with 5M parameters. New variants of ViT introduce smaller and cheaper architectures. For example, T2T-ViT [50] reduces the number of parameters and compute using a new tokenization and narrower networks. Pyramidal ViTs [45], designed for dense prediction tasks, follow a CNN-like pyramidal structure, that also reduces the size of the model. Hybrids of CNNs and Transformers typically allow smaller models to perform well, such as the ViT-CNN hybrid in [15], BoTNet [35], and HaloNet [43]. However, the other direction, increasing the scale of ViT, is less explored. While language Transformers are still much larger than Vision Transformers, understanding the scaling properties and the improvements introduced in this paper represent a step in this direction.

Scaling Laws [21] present a thorough study of the empirical scaling laws of neural language models. The authors

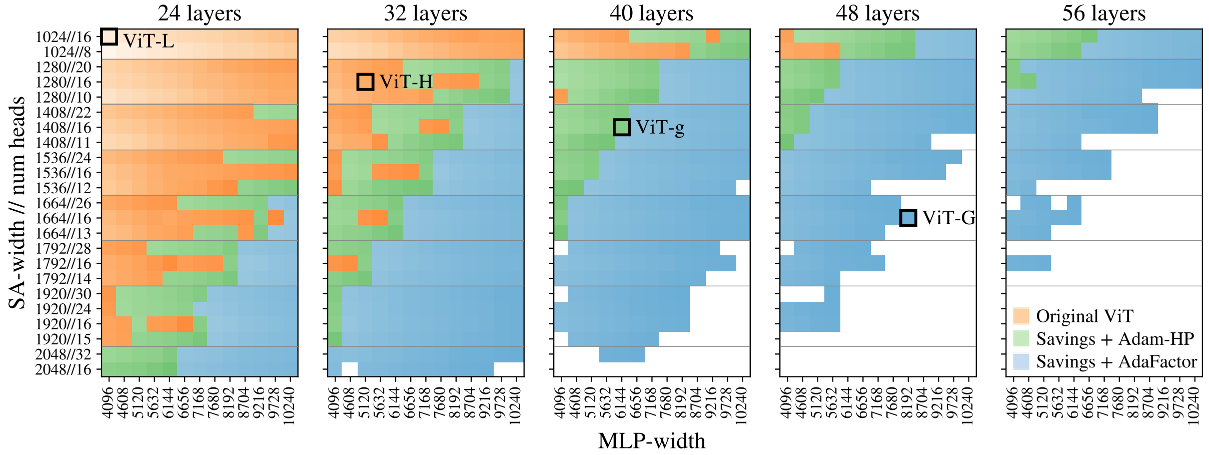

Figure 8. Combined results of the "Shapefinder" simulation for the original ViT in orange, our improvements together with half-precision Adam (*e.g*. ViT-g) in green, and finally with our modified AdaFactor in blue. White areas ran out of memory. The brightness of the dot corresponds to its relative training speed.

fit power laws that describe the relationships between compute, data size, model size, and performance. Following these laws, GPT-3, a 175B parameter language model was successfully trained [6]. [18] presents laws for autoregressive generative modelling in other modalities, including the generation of images. Our paper contains the first study of scaling laws for the discriminative modelling of images.

Scaling-up Vision Models Many papers scale up CNNs to attain improved performance. EfficientNets [37, 38] present a scaling strategy that balances compute between depth, width, and resolution and apply it to MobileNets. This strategy is revisited in [3, 47] to further improve the performance of ResNets [17]. Large CNNs have attained excellent performance in visual recognition, such as AmoebaNet-B(18, 512) (557M parameters) trained using GPipe pipeline parallelism [19], ResNeXt-101 32×48d (829M parameters) pre-trained on weakly-labelled Instagram images [26], EfficientNet-L2 (480M parameters) trained with ImageNet pseudo-labels on JFT-300M [49], and BiT-L-ResNet152x4 (928M parameters) pre-trained on JFT-300M [22]. Recently, [41, 53] explore strategies to scale the depth of ViTs. We are the first to scale Vision Transformers to even larger size and reache new state-of-the-art results doing so. The concurrent work [12] focuses on CNN and ViT hybrid architectures.

5. Discussion

Limitations. This work uses the proprietary JFT-3B dataset for the scaling laws study. To make our insights more reliable and generalizable, we verify that the scaling laws also apply on the public ImageNet-21k dataset.

Societal impact. A potential broader cost of this work is the energy required to perform the experiments in our scaling study, especially in training the largest ViT-G model. However, this cost may be amortized in two ways. First, such studies of scaling laws need only be performed once; We hope future developers of ViT models may use our results to design models that can be trained with fewer compute resources. Second, the models trained are designed primarily for transfer learning. Transfer of pre-trained weights is much less expensive than training from scratch on a downstream task, and typically reaches higher accuracy. Therefore, by transferring our models to many tasks, the pre-training compute is further amortized.

6. Conclusion

We demonstrate that the performance-compute frontier for ViT models with enough training data roughly follows a (saturating) power law. Crucially, in order to stay on this frontier one has to simultaneously scale compute and model size; that is, not increasing a model's size when extra compute becomes available is suboptimal. We also demonstrate that larger models are much more sample efficient and are great few-shot learners. Finally, we present a new training recipe, which allows one to efficiently train large and high-performing ViT models. Note, that our conclusions may not necessarily generalize beyond the scale we have studied and they may not generalize beyond the ViT family of models.

Acknowledgements We thank James Bradbury and Vivek Sharma for their help on using large-scale infrastructure; Alexey Dosovitskiy, Joan Puigcerver, Basil Mustafa, Carlos Riquelme for insightful discussions; Tom Duerig, Austin Tarango, Daniel Keysers, Howard Zhou, Wenlei Zhou, Yanan Bao for discussions on JFT; the Google Brain team at large for providing a supportive research environment.

References

[1] Osman Aka, Ken Burke, Alex Bäuerle, Christina Greer, and Margaret Mitchell. Measuring model biases in the absence of ground truth. *arXiv preprint arXiv:2103.03417*, 2021. 5

[2] Andrei Barbu, D. Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, J. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 4

[3] Irwan Bello, William Fedus, Xianzhi Du, Ekin D Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *arXiv preprint arXiv:2103.07579*, 2021. 8

[4] Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 3, 4

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1, 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 1

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

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 4

[10] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020. 7

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995. 5

[12] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *CoRR*, abs/2106.04803, 2021. 8

[13] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[15] 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. 1, 4, 5, 6, 7, 11

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8

[18] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 3, 8

[19] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, and zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *NeurIPS*, 2019. 8

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021. 4

[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 3, 7

[22] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, J. Puigcerver, Jessica Yung, S. Gelly, and N. Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *ECCV*, 2020. 4, 5, 8, 11

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 11

[24] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 5

[25] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 3

[26] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, September 2018. 6, 8

[27] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 2, 11

[28] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, and Quoc V. Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020. 4

[29] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. 11

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 4

[31] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In Christine Cuicchi, Irene Qualters, and William T. Kramer, editors, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, page 20. IEEE/ACM, 2020. 6

[32] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019. 3, 4

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 4

[34] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*, 2018. 6

[35] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. 7

[36] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *ICCV*, Oct 2017. 5

[37] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 8

[38] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10096–10106. PMLR, 2021. 8

[39] Hanlin Tang, Shaoduo Gan, Ammar Ahmad Awan, Samyam Rajbhandari, Conglong Li, Xiangru Lian, Ji Liu, Ce Zhang, and Yuxiong He. 1-bit adam: Communication efficient large-scale training with adam's convergence speed. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10118–10129. PMLR, 2021. 6

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 7

[41] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *CoRR*, abs/2103.17239, 2021. 5, 8

[42] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2020. 11

[43] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. *arXiv preprint arXiv:2103.12731*, 2021. 7

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1

[45] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 7

[46] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2, 11

[47] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *CoRR*, abs/2110.00476, 2021. 8

[48] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019. 4

[49] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, June 2020. 8

[50] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 7

[51] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019. 4

[52] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 4

[53] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *CoRR*, abs/2103.11886, 2021. 8