ADVERSARIALLY SELF-SUPERVISED PRE-TRAINING IMPROVES ACCURACY AND ROBUSTNESS

Sylvestre-Alvise Rebuffi, Olivia Wiles, Evan Shelhamer & Sven Gowal
DeepMind, London
{sylvestre,sgowal}@deepmind.com

ABSTRACT

There is growing interest in learning visual representations that work well across distribution shifts as illustrated by the increasing number of IMAGENET evaluation sets. In this paper, we reconsider adversarial training, which is generally used as a defense against adversarial shifts, as a way to improve the pre-training of representations for transfer across tasks and natural shifts. In this study we combine adversarial training with different self-supervised pre-training methods such as bootstrap your own latent (BYOL), masked auto-encoding (MAE), and the auxiliary task of rotation prediction (RotNet). We show that the adversarial versions of these self-supervision methods consistently lead to better fine-tuning accuracy both in and out of distribution compared to standard self-supervision, even with nominal/non-adversarial fine-tuning. Furthermore we observe that, to reach best performance with adversarial self-supervised pre-training, (1) the optimal perturbation radius differs among pre-training methods, and (2) that the robust parameters of early layers need to be preserved during fine-tuning to avoid losing the benefits of adversarial pre-training. Finally, we show that there is not a single adversarial self-supervised method that dominates others across all variants, but that adversarial MAE is the best choice for in-distribution variants, and that adversarial BYOL is best for out-of-distribution variants.

1 INTRODUCTION

As deep networks continue to improve, with new architectures such as Vision Transformers (Dosovitskiy et al., 2020), the results on standard large-scale benchmarks like IMAGENET have begun to saturate and to overfit to their test set (Recht et al., 2019). As illustrated by the emergence of several IMAGENET variants, the interest of the community has started to shift towards training models performing well on the standard evaluation set but which are also robust across distribution shifts (Hendrycks et al., 2019b; 2020; Wang et al., 2019; Geirhos et al., 2018; Hendrycks & Dietterich, 2018).

In parallel with the development of novel architectures, the classification performance of networks has been pushed by the advent of new self-supervised learning methods. Indeed, in recent work in computer vision with Masked autoencoder (He et al., 2022) and natural language processing with BERT (Devlin et al., 2018), transfer learning by first pre-training then fine-tuning dominates the accuracy of simple fully-supervised training across tasks. However, sheer accuracy on a standard evaluation set is not the only metric for a model, and *robustness* to distribution shifts in particular is a key concern for deployment. In separate threads, recent work has highlighted the potential to improve transfer by either adversarial pre-training with perturbed inputs (Salman et al., 2020) or self-supervised pre-training with auxiliary outputs and losses (Gidaris et al., 2018; He et al., 2022). For transfer effects beyond accuracy alone, there is evidence that self-supervised pre-training can not only rival the accuracy of supervised pre-training, but that it can also deliver fairer and more general representations without relying on hand-labeled annotations and their possible biases (Goyal et al., 2022). In this work, we join the threads of adversarial and self-supervised learning to devise a new pre-training scheme for *self-adversarial learning* based on MAE, and empirically show that combining any pre-training method in our study with adversarial training achieves more accurate and more robust transfer. Overall, our contributions are as follows:

- We create an adversarial version of the self-supervision method MAE, and further examine the existing adversarial BYOL and adversarial RotNet on both in and out of distribution benchmarks. As adversarial attacks depend on the training loss, we detail in Section 2 and Appendix B the specific adversarial training scheme for each self-supervised method.
- We empirically demonstrate that adversarial training consistently improves self-supervised pre-training for all the studied methods, with a significant boost in accuracy on IMAGENET and its variants. Furthermore, adversarial pre-training reduces the performance gaps among self-supervised methods. Surprisingly, even the older RotNet rivals the accuracy of more recent and strong methods like MAE when both are trained adversarially.
- We identify the key factors in the pre-training and fine-tuning procedures to reach best performance. We study the influence of the attack and perturbation radius used during pre-training on the downstream classification performance. For the fine-tuning stage, we show the impact of layer-wise learning rate decay on preserving the robust filters learned during pre-training and how it leads to better performance on out of distribution benchmarks.
- We show in a fine-grained study over IMAGENET variants that there is not a single adversarial self-supervised method which performs best on all the variants but that self-supervised methods specialize to certain shifts.

2 SELF-ADVERSARIAL TRAINING

We improve pre-training by attacking the label-free losses of self-supervised learning. That is, the model is pre-trained with an adversarial version of self-supervision, and then fine-tuned on the task of interest by standard supervised learning. As adversarial attacks modify the loss for training, we create a separate setup to adversarially train each self-supervised method. In the following, we first define adversarial training as background, and then explain its application to self-supervision by masked auto-encoding (MAE). We recall in Appendix B the existing adversarial RotNet and BYOL.

3 BACKGROUND: ADVERSARIAL TRAINING

Adversarial training seeks to find a model that is robust to small perturbations that reside within an ℓ_p-norm ball. Madry et al. (2018) propose to find the parameters $\boldsymbol{\theta}$ of such a model by solving a min-max problem at each training step:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathcal{L}^{\text{AT}}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}, y) \right] \tag{1}$$

where pairs of samples $\boldsymbol{x}$ and labels y are sampled from the data distribution $\mathcal{D}$. $\mathcal{L}^{\text{AT}}_{\boldsymbol{\theta}}$ is a suitable loss function (such as the cross-entropy loss for classification tasks) using the output of the model parametrized by $\boldsymbol{\theta}$. $\mathbb{S}$ denotes the constrained space of perturbations. For ℓ_p norm-bounded perturbations of size ϵ the adversarial set of perturbations is defined as $\mathbb{S}_p = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_p \leq \epsilon\}$. In the rest of the manuscript we will use ϵ_p to denote ℓ_p norm-bounded perturbations of size ϵ (e.g., $\epsilon_\infty = 4/255$). To solve the inner optimization problem, Madry et al. (2018) use Projected Gradient Descent (PGD), which computes the adversarial perturbation in K gradient ascent steps with step size α. For an arbitrary adversarial loss $\mathcal{L}^{\text{AT}}$, we denote as $\text{PGD}^K_{\mathcal{L}^{\text{AT}}}(\boldsymbol{x}, y)$ the inner optimization with K steps defined as

$$\boldsymbol{\delta}^{(k+1)} \leftarrow \text{proj}_{\mathbb{S}}\left(\boldsymbol{\delta}^{(k)} + \alpha\,\text{sign}\left(\nabla_{\boldsymbol{\delta}^{(k)}} \mathcal{L}^{\text{AT}}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{(k)}, y)\right)\right) \tag{2}$$

where $\boldsymbol{\delta}^{(0)}$ is randomly sampled within $\mathbb{S}$, and where $\text{proj}_{\mathbb{A}}(\boldsymbol{a})$ projects a point $\boldsymbol{a}$ back onto a set $\mathbb{A}$, $\text{proj}_{\mathbb{A}}(\boldsymbol{a}) = \text{argmin}_{\boldsymbol{a}'\in\mathbb{A}} \|\boldsymbol{a}-\boldsymbol{a}'\|_2$.

3.1 ADVERSARIAL MASKED AUTOENCODER

Masked autoencoder (MAE), as introduced by He et al. (2022), is a self-supervised technique based on regression. Random patches of the input image are masked out and the autoencoder is tasked to predict the missing pixels based on the remaining visible patches. The network architecture is composed of an encoder $e(\cdot;\boldsymbol{\theta})$ that operates on the visible patches and a lightweight decoder $d(\cdot;\boldsymbol{\theta})$

that reconstructs the masked patches from the latent representation. Given an image $\boldsymbol{x}$ we decompose it into a batch of patches $\boldsymbol{p}$. We randomly sample a mask $\boldsymbol{m}$ which is equal to 1 for a visible patch and to 0 for a masked out patch. We feed the patches $\boldsymbol{p}$ and the mask $\boldsymbol{m}$ to the auto-encoder $d \circ e$ and use the mean squared error (MSE) error to match the output of the autoencoder with the normalized input patches. This loss is only computed on masked patches, so MAE minimizes the following loss:

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{MAE}} \propto (1-\boldsymbol{m}) \left\| d \circ e(\boldsymbol{p}, \boldsymbol{m}; \boldsymbol{\theta}) - \frac{\boldsymbol{p} - \mu(\boldsymbol{p})}{\sigma(\boldsymbol{p})} \right\|_2^2. \tag{3}$$

where μ and σ are respectively the mean and standard deviation functions over width, length and channel dimensions. At the end of the training, the decoder is discarded and the encoder representation $e(\boldsymbol{p}, \mathbf{1}; \boldsymbol{\theta})$ of fully visible images $\boldsymbol{x}$ with patches $\boldsymbol{p}$ is fine-tuned on downstream tasks.

To adapt MAE to adversarial training, we need to design an adversarial attack which maximizes the disagreement between the autoencoder output and the normalized input patches by perturbing the input images. The input image is broken up into patches before being passed to the MAE loss, so we need to decide how to attack the patches. We observe that MAE's loss uses the input patches $\boldsymbol{p}$ in two different ways depending on whether the patch is visible or not. Visible patches are given as input to the autoencoder whereas masked patches are used for the reconstruction target. In our design, we propose to only attack the visible patches of the image, as *only* the visible patches impact the output of the autoencoder. The visible patches are independent of the masked ones, so perturbing the masked ones would modify the target, but the autoencoder has no way to ascertain which masked patches had been perturbed and thereby how to be robust to that perturbation. Hence, we propose to use

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{A-MAE}} = (1-\boldsymbol{m}) \left\| d \circ e(\boldsymbol{p} + \boldsymbol{\delta}, \boldsymbol{m}; \boldsymbol{\theta}) - \frac{\boldsymbol{p} - \mu(\boldsymbol{p})}{\sigma(\boldsymbol{p})} \right\|_2^2. \tag{4}$$

where we use $\text{PGD}^K_{\mathcal{L}^{\text{A-MAE}}}$ to approximate the perturbation in K gradient ascent steps and we optimize the model parameters using this adversarial loss where the attacked visible patches are fed to the autoencoder. We emphasize that the regression task of adversarial MAE is much more challenging than that of standard MAE as the autoencoder has to reconstruct the original missing patches based on perturbed visible patches. Hence, this task is a combination of both inpainting and denoising.

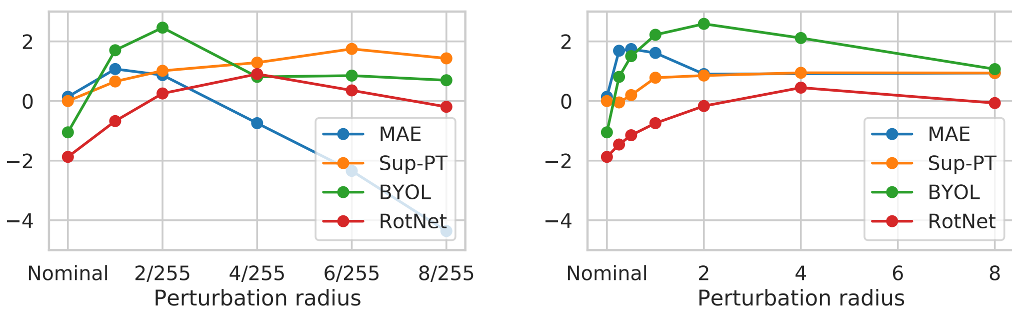
4 Experimental results

4.1 Adversarial training improves pre-training

All methods benefit from adversarial training. We study the fine-tuning performance of adversarially trained self-supervised models which are fine-tuned on clean ImageNet images. We report in Figure 1 their average classification performance on ImageNet and its seven variants. We compare models which are adversarially pre-trained with three different self-supervised methods and attacked with either ℓ_∞ or ℓ_2 attacks and varying perturbation radii. We also add models adversarially pre-trained with full supervision, which are then fine-tuned again with full supervision but on clean images.

First, we observe that all the curves increase when moving to the right, improving over their starting points which correspond to the nominal self-supervised methods. Thus, all the pre-training methods benefit from adversarial training, as all methods obtain significantly higher average accuracies than when nominally pre-trained. Second, all the curves reach positive values so all the adversarially self-supervised methods achieve better classification performance than a model nominally trained from scratch on ImageNet with full supervision. This is not the case for nominal BYOL and nominal RotNet whose points are below 0. Finally, self-supervision with adversarial training bridges the performance gap between methods, as the best adversarial RotNet result, which already performs better than nominal MAE, is only 1.68% in average accuracy below the best adversarial BYOL result.

Finding the optimal attack. When comparing the two panels of Figure 1 (as their y-axis are aligned), we notice that adversarial RotNet and adversarial supervised pre-training achieve their best results when using ℓ_∞ attacks. On the contrary, the best adversarial MAE result with a ℓ_2 attack is +0.67% better than the best average accuracy with a ℓ_∞ attack. Adversarial BYOL achieves similar



(a) ℓ_∞ adversarial training

(b) ℓ_2 adversarial training

Figure 1: Influence of the perturbation radius. We report the average classification performance over IMAGENET variants for four adversarial pre-training methods using ℓ_∞ (left panel) and ℓ_2 (right panel) attacks with various perturbation radii. Sup-PT corresponds to supervised pre-training with the true class labels. For better comparison, we (1) subtract off the average accuracy obtained by a model nominally trained from scratch on IMAGENET with full supervision and (2) align the y-axis of both panels.

best average accuracy with both ℓ_∞ and ℓ_2 attacks and is the method which benefits the most from adversarial training with a +3.64% improvement for $\epsilon_2 = 2$ over nominal BYOL. Secondly, we observe that the optimal perturbation radius differs for the various pre-training methods. Indeed, if we focus on the left panel with ℓ_∞ attacks, MAE reaches its maximum for $\epsilon_\infty = 1/255$, BYOL for $\epsilon_\infty = 2/255$, RotNet for $\epsilon_\infty = 4/255$ and adversarial supervised for $\epsilon_\infty = 6/255$. We hypothesize that there exist different optimal perturbations radii because these pre-training methods use different training losses, which might be more or less sensitive to adversarial perturbations. Indeed, the regression task of adversarial MAE of reconstructing patches of the clean image becomes extremely difficult when the non-masked surrounding patches can be perturbed with a strong perturbation radius. In comparison, classification losses used for adversarial supervised pre-training and RotNet are less sensitive to the perturbation radius.

4.2 FINE-GRAINED ANALYSIS

Influence of the pre-training methods. In subsection 4.1, we observed that all the adversarial pre-training methods have better average performance on IMAGENET and its variants than training from scratch on IMAGENET . Delving into a more fine-grained analysis, we study the per variant performance of these models. We report in Figure 2 the results of the models with the best average accuracies for the different pre-training methods and attacks (nominal, ℓ_∞ or ℓ_2). When comparing the ℓ_∞ and ℓ_2 columns to the nominal columns, we see that adversarial pre-training improves over nominal pre-training for all the methods and for all the variants. More interestingly, we notice that there is not a single method which works best on all the variants. Indeed, if we look at the last two columns of Figure 2, we observe that adversarial MAE performs better than adversarial BYOL on standard ImageNet, IN-V2, IN-Real, IN-A and IN-Sketch but worse on IN-R, Conflict Stimuli and IN-C. Additionally, the fine-tuned performance on the standard ImageNet test set is not a sufficient indicator of performance for pre-training methods, as these methods show different strengths and weaknesses depending on the variant. To illustrate this point, we observe for ℓ_2 attacks that adversarial supervised pre-training and adversarial RotNet have opposite behaviours, with supervised pre-training performing better on domains closer to the original test set whereas adversarial RotNet performs relatively better on IN-R, IN-Sketch and Conflict Stimuli.

Influence of the layer-wise learning rate decay. For all the pre-training methods, we use the same fine-tuning procedure proposed in He et al. (2022) which prevents earlier layers from changing too much during fine-tuning thanks to layer-wise learning rate decay. In this setting the learning rate of the k to last transformer block is obtained by multiplying the nominal learning rate, which is the learning rate applied to the last layer, by a factor γ^{-k} where γ is the layer-wise learning rate decay. Notably, $\gamma = 1$ boils down to standard full-finetuning and $\gamma = 0$ to training a classifier layer on top of a frozen feature extractor. We study the impact of this hyperparameter in Figure 3 where we report

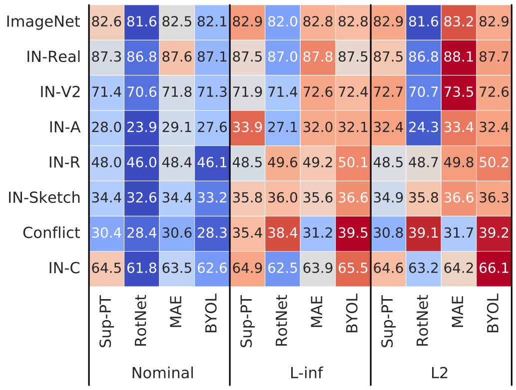


Figure 2: Influence of the pre-training methods and attacks. We report the accuracy on IMAGENET variants of the models with the best average accuracies in Figure 1 for the different pre-training methods and attacks. The three groups of columns (from left to right) correspond respectively to pre-training nominally, with ℓ_∞ attacks and with ℓ_2 attacks. The colours are row normalized and red means better.

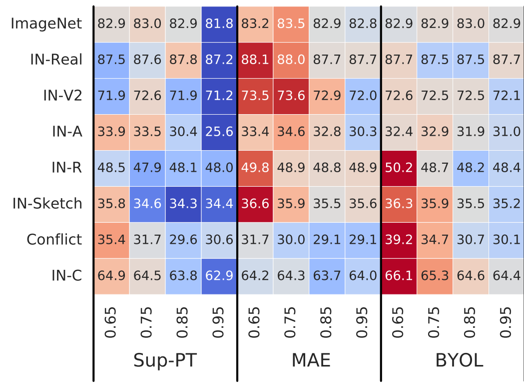


Figure 3: Influence of the layer-wise learning rate decay. We report the accuracy on IMAGENET variants when fine-tuning with various layer-wise learning rate decay from different pre-training models. The three groups of columns correspond to the pre-training models achieving the best average performance over variants in Figure 1: supervised with $\epsilon_\infty = 6/255$, MAE with $\epsilon_2 = 0.5$ and BYOL with $\epsilon_2 = 2$. The colours are row normalized and red means better.

the per variant performance when varying the layer-wise learning rate decay for the three adversarial pre-training methods achieving the best average performance over variants, namely supervised with $\epsilon_\infty = 6/255$, MAE with $\epsilon_2 = 0.5$ and BYOL with $\epsilon_2 = 2$. Similarly for the three methods, we observe that some variants benefit more from the smallest decay $\gamma = 0.65$ such as IN-R, IN-Sketch, IN-C and more significantly Conflict Stimuli. A small layer-wise learning rate decay acts as a ”soft” freezing of the early layers which have a much smaller effective learning rate than later layers. Thus keeping early layers close to the robust filters learned during the adversarial pre-training phase is helpful for variants which are the most outside of the IMAGENET training distribution. Conversely, variants such as IN-V2 or IN-A, closer to the training distribution, obtain better results with a larger decay $\gamma = 0.75$.

A possible explanation is that preserving the robust early layers during nominal fine-tuning can retain some of the robustness learnt during adversarial pre-training and transfer this robustness on the fine-tuned task. To illustrate this, we report in Figure 5 (in the appendix) the robust test accuracy on IMAGENET of models nominally fine-tuned with various layer-wise learning rate decays from the same network pre-trained using adversarial BYOL with $\epsilon_2 = 2$. We observe that soft freezing the early layers with a smaller layer-wise learning rate decay (blue curve) during fine-tuning leads to a higher robustness on the downstream classification task. This transferred robustness could explain the better performance on the variants which are the most outside of the IMAGENET training distribution.

5 CONCLUSION

In this work we have shown that adversarial training consistently improves self-supervised pre-training. In fact, not only models fine-tuned from adversarial versions of self-supervised methods have better performance on the standard evaluation set of IMAGENET but they also do significantly better in the face of distribution shifts with strong improvements on the IMAGENET variants. Furthermore, adversarial training narrows the performance gap between self-supervised methods as even adversarial RotNet can compete (on average) with more recent methods such as MAE.

In a fine-grained analysis over IMAGENET variants, we have shown that the various self-supervision methods specialize on certain distribution shifts and that there is not a single method which performs best on all the variants. These observations open up new interesting directions for future work about whether there are potential self-supervised methods that could do well on many (or even all) types of distribution shift.

REFERENCES

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020a.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020b. URL `https://arxiv.org/pdf/2003.12862`.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020c.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

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.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. *CVPR*, 2018.

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. *arXiv preprint arXiv:2010.11929*, 2020.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5414–5423, 2021.

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34:21480–21492, 2021.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/pdf?id=Bygh9j09KX`.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2020a.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020b. URL `https://arxiv.org/pdf/2010.03593`.

Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised Adversarial Robustness for the Low-label, High-data Regime. In *International Conference on Learning Representations*, 2021a. URL `https://openreview.net/pdf?id=bgQek2O63w`.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy Mann. Improving Robustness using Generated Data. *arXiv preprint arXiv:2110.09468*, 2021b. URL `https://arxiv.org/pdf/2110.09468`.

Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision Models Are More Robust And Fair When Pretrained On Uncurated Images Without Supervision. *arXiv preprint arXiv:2202.08360*, 2022. URL `https://arxiv.org/pdf/2202.08360`.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

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*, pp. 16000–16009, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/pdf?id=HJz6tiCqYm`.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019b.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *arXiv preprint arXiv:2006.16241*, 2020. URL `https://arxiv.org/pdf/2006.16241`.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. volume 33, pp. 16199–16210, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. volume 33, pp. 2983–2994, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

Chengzhi Mao, Amogh Gupta, Vikram Nitin, Baishakhi Ray, Shuran Song, Junfeng Yang, and Carl Vondrick. Multitask learning strengthens adversarial robustness. In *European Conference on Computer Vision*, pp. 158–174. Springer, 2020.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit Pairing Methods Can Fool Gradient-Based Attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of Tricks for Adversarial Training. *arXiv preprint arXiv:2010.00467*, 2020. URL `https://arxiv.org/pdf/2010.00467`.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing Data Augmentation to Improve Adversarial Robustness. *arXiv preprint arXiv:2103.01946*, 2021. URL `https://arxiv.org/pdf/2103.01946`.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do Adversarially Robust ImageNet Models Transfer Better? In *NeurIPS*, 2020. URL `https://arxiv.org/pdf/2007.08489`.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *CVPR*, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

A RELATED WORK

Adversarial Robustness. Adversarial training improves the robustness of supervised learning by perturbing its inputs during training (Madry et al., 2018; Kurakin et al., 2016), and as posed by Madry et al. (2018) it remains one of the most successful defenses to attack as measured in standard evaluations (Croce et al., 2021). It has been augmented in different ways—by changes to the attack optimization (e.g., by incorporating momentum (Dong et al., 2018)), loss (e.g., logit pairing (Mosbach et al., 2018)), model architecture (e.g., feature denoising (Xie et al., 2019)), and data augmentation (e.g., leveraging synthetic examples (Rebuffi et al., 2021; Gowal et al., 2021b))—and thoroughly analyzed (Gowal et al., 2020b; Pang et al., 2020). During training, adversarial perturbations are generated by counter-optimizing the supervised loss of the *main* task, and they do not take into account *auxiliary* tasks, such as the losses provided by self-supervised learning. For evaluation, adversarial training has almost exclusively been studied for robustness, and not transfer, and studying its transfer (Salman et al., 2020) has considered only supervised and not self-supervised pre-training. Our work adapts adversarial training from supervised learning to self-supervised learning, which necessitates the careful design and evaluation of adversarial schemes to address each type of self-supervised output and loss. We report the first results for the adversarial training of self-supervision by MAE.

Self-supervised Training. Self-supervised learning enables pre-training on unlabeled data by designing auxiliary tasks and losses that provide their own labels. Unsupervised learning uses the data as its own "labels" for prediction, generation, or reconstruction. Pre-training without labels has attracted widespread attention for its potential to reduce effort and raise accuracy: unlike supervised pre-training, such methods do not require time-consuming and expensive labeling, and they can learn more transferable representations from the input itself than potentially limited or biased labels (Ericsson et al., 2021; Goyal et al., 2022). Auxiliary tasks and losses for vision often transform the image, then supervise the transformation or its inverse, by for example recognizing rotations (Gidaris et al., 2018), colorizing (Zhang et al., 2016), locating shuffled patches (Doersch et al., 2015; Noroozi & Favaro, 2016), or clustering (Caron et al., 2018). Contrastive learning more generally defines positive and negative pairs as transformations of the same or different images, then optimizes to differentiate between positives and negatives (Oord et al., 2018; Chen et al., 2020c; He et al., 2020) or to simply bring together positive pairs (Grill et al., 2020; Chen & He, 2021; Richemond et al., 2020). Unsupervised learning by reconstruction and generation includes masking then reconstructing or generating image patches (He et al., 2022; Pathak et al., 2016) or autoregressively generating neighboring pixels (Chen et al., 2020a). To complement progress on the invention and tuning of self-supervised and unsupervised losses, we demonstrate that casting such losses into adversarial counterparts can further improve the robustness and transferability of the learned representations, and do so without the supervised task labels that are needed for standard adversarial training.

Self-supervision for Robustness. Self-supervised and multi-task losses have been shown to improve robustness in combination with supervised training (Hendrycks et al., 2019a; Mao et al., 2020). To improve robustness without full supervision, recent work has investigated adversarial training on unlabeled data by robust optimization of self-supervised losses. Chen et al. (2020b) experiment with adversarial self-supervised classification, including RotNet, followed by adversarial or nominal fine-tuning. Adversarial contrastive learning by RoCL (Kim et al., 2020), ACL (Jiang et al., 2020), and AdvCL (Fan et al., 2021) augment contrastive pairs with adversarial perturbations to improve robustness to attack for pre-training and fine-tuning on CIFAR-10/100. Bootstrap your own robust latents (BYORL) (Gowal et al., 2021a) extends BYOL by perturbing its positives, and shows both improved adversarial robustness and nominal accuracy on the training dataset in the regime of limited labeled data. However, these works each study a single self-supervised loss in isolation, on smaller datasets like CIFAR-10/100, and with smaller and less accurate models than the current state-of-the-art for vision. We contribute to this line of work with a broader and deeper examination of robustness and transfer, and report results for pre-training and fine-tuning across a variety of self-supervised methods and datasets at IMAGENET scale with the stronger ViT-B16 architecture.

B More Self-Adversarial Training methods

B.1 Adversarial RotNet

Gidaris et al. (2018) propose RotNet, a self-supervised learning method which pre-trains the model by using a fully supervised pretext task based on rotation prediction. The unlabeled images are randomly rotated by 0°, 90°, 180° or 270° degrees before being fed to the network, and the network is trained to predict which rotation was applied to each image. This is a standard classification problem with four classes corresponding to the four rotations. Similar to Chen et al. (2020b), RotNet can straightforwardly be adapted to adversarial training by modifying the adversarial risk of equation 1 for the rotation prediction task:

$$\arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},.)\in\mathcal{D}} \mathop{\mathbb{E}}_{y\sim\mathcal{U}[\{0,\ldots,270\}]} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta}}(\mathrm{rot}(\boldsymbol{x},y)+\boldsymbol{\delta},y)\right] \tag{5}$$

where $\mathcal{L}^{\mathrm{CE}}$ is the cross-entropy loss, the label y corresponds to a rotation randomly sampled among the four possible rotations 0°, 90°, 180° and 270° degrees and $\mathrm{rot}(\boldsymbol{x}, y)$ is the function which rotates the sample $\boldsymbol{x}$ by y degrees. In adversarial RotNet, the adversarial perturbations are optimized to fool the network into predicting the wrong rotation.

B.2 Adversarial BYOL

Grill et al. (2020) propose BYOL, a self-supervised learning method based on two networks: an online and a target network. The goal of the online network is to predict the target network representation of the same image under different augmented views. The target network is defined by an exponential moving average of the online network parameters. The online network is composed of three stages: an encoder $e(\cdot;\boldsymbol{\theta})$, a projector $g(\cdot;\boldsymbol{\theta})$ and a predictor $q(\cdot;\boldsymbol{\theta})$. We denote by $\gamma = g\circ e$ the composition of the encoder and projector and by $\kappa = q\circ g\circ e$ the composition of the encoder, projector and predictor. The target network has the same architecture as the online network but skips the predictor and uses as weights $\boldsymbol{\xi}$, an exponential moving average of the weights $\boldsymbol{\theta}$. Given an image $\boldsymbol{x}$, and two augmentations $t, t' \sim \mathcal{T}$ sampled from a set of augmentations BYOL produces two augmented views $\boldsymbol{v} = t(\boldsymbol{x})$ and $\boldsymbol{v}' = t'(\boldsymbol{x})$. The first view passes through the online network, producing a representation $\boldsymbol{h} = e(\boldsymbol{x};\boldsymbol{\theta})$ and a projection $\boldsymbol{z} = g(\boldsymbol{h};\boldsymbol{\theta})$. The second view similarly passes through the target network, producing a target projection $\boldsymbol{z}' = \gamma(\boldsymbol{v}';\boldsymbol{\xi})$. Finally, given an online prediction $q(\boldsymbol{z};\boldsymbol{\theta}) = \kappa(\boldsymbol{v};\boldsymbol{\theta})$ (which should be predictive of the target projection), BYOL minimizes the loss

$$\mathcal{L}^{\mathrm{BYOL}}_{\boldsymbol{\theta}} \propto -\frac{\kappa(\boldsymbol{v};\boldsymbol{\theta})^T\boldsymbol{z}'}{\|\kappa(\boldsymbol{v};\boldsymbol{\theta})\|_2\cdot\|\boldsymbol{z}'\|_2}. \tag{6}$$

At the end of training, everything but e and $\boldsymbol{\theta}$ is discarded and only the representation $e(\boldsymbol{x};\boldsymbol{\theta})$ of an image $\boldsymbol{x}$ is used by downstream applications.

Similar to Gowal et al. (2020a), we adapt BYOL to the adversarial setting by performing the attacks through the online network. There are still two views $\boldsymbol{v} = t(\boldsymbol{x})$ and $\boldsymbol{v}' = t'(\boldsymbol{x})$ of the same image $\boldsymbol{x}$. Now, while the second view goes through the target network unmodified to produce a target projection $\boldsymbol{z}' = \gamma(\boldsymbol{v}';\boldsymbol{\xi})$, the first view is further augmented via an adversarial attack. To maximize the loss in Equation 6, the optimal perturbation has to minimize the cosine similarity between the online prediction $\kappa(\boldsymbol{v}+\boldsymbol{\delta};\boldsymbol{\theta})$ and target projection $\boldsymbol{z}'$:

$$\max_{\boldsymbol{\delta}\in\mathbb{S}} -\frac{\kappa(\boldsymbol{v}+\boldsymbol{\delta};\boldsymbol{\theta})^T\boldsymbol{z}'}{\|\kappa(\boldsymbol{v}+\boldsymbol{\delta};\boldsymbol{\theta})\|_2\cdot\|\boldsymbol{z}'\|_2}. \tag{7}$$

Similar to the other methods, the optimal perturbation is approximated by using PGD and then we minimize the loss in Equation 6 where the attacked view is given to the online network. As in the original BYOL, we can symmetrize the procedure by feeding $\boldsymbol{v}'$ to the online network and $\boldsymbol{v}$ to the target network. The adversarial attack is then executed on $\boldsymbol{v}'$ instead of $\boldsymbol{v}$ and tries to minimize the cosine similarity between the online prediction $\kappa(\boldsymbol{v}'+\boldsymbol{\delta};\boldsymbol{\theta})$ and target projection $\gamma(\boldsymbol{v};\boldsymbol{\xi})$.

C Experimental setup

Architecture. We base our studies on the B16 variant of the Vision Transformer (ViT-B16) of Dosovitskiy et al. (2020). Furthermore, to be consistent across pre-training methods, we use for each method the same modified ViT architecture proposed by He et al. (2022) for fine-tuning MAE. In this architecture, the linear head is not applied to the classification token but to the mean of the final tokens except the classification token. When pre-training with BYOL, we follow Grill et al. (2020); Gowal et al. (2020a) and use MLPs with hidden dimension 4096 and output dimension 256 for the projector and predictor networks on top of the ViT. Regarding the decoder of the MAE, we use 8 transformer layers with 16 heads and a hidden dimension of 2048 in the MLPs.

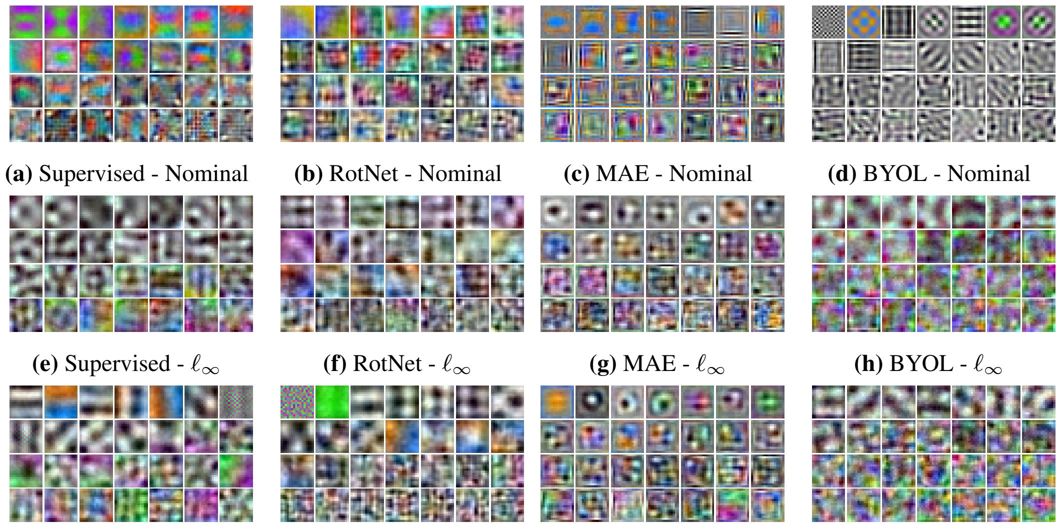
Attacks. We consider several attacks and perturbation radiuses when pre-training on adversarial samples: ℓ_∞ -bounded attacks with radius $\epsilon \in \{1/255, 2/255, 4/255, 6/255, 8/255\}$ and ℓ_2 -bounded attacks with radius $\epsilon \in \{0.25, 0.5, 1, 2, 4, 8\}$. During training we compute the adversarial perturbations with 2 steps Projected Gradient Descent (Madry et al., 2018) named PGD^2 where we use a gradient descent update with a fixed step size of $5\epsilon/8$.

Training. For fully supervised and MAE pre-training we use the hyperparameters described in He et al. (2022). For BYOL, we adversarially pre-train the model by using the training pipeline of Gowal et al. (2020a). For RotNet, we use the same hyperparameters as for supervised pre-training but without using CutMix and MixUp. Regarding fine-tuning, we fine-tune for 100 epochs with batch size 512, using AdamW with learning rate 0.0005 and weight decay 0.05 and we use the same data augmentations as in He et al. (2022). Furthermore, we sweep over the layer-wise learning rate decay within $\{0.65, 0.75, 0.85, 0.95\}$.

Datasets. We evaluate the fine-tuning performance of the various pre-training methods on the ImageNet dataset (Russakovsky et al., 2015) and its variants to measure their generalization across distribution shifts. We consider ImageNet-A (Hendrycks et al., 2019b), ImageNet-R (Hendrycks et al., 2020), ImageNet-Sketch (Wang et al., 2019), Conflict Stimuli (sometimes called ImageNet -Stylized) (Geirhos et al., 2018) and ImageNet-C (Hendrycks & Dietterich, 2018). For both training and evaluation we use images at 224×224 resolution. We also study the transfer learning performance of the pre-trained models on smaller datasets: Cifar-10, Cifar-100 (Krizhevsky, 2009), SUN-397 (Xiao et al., 2010), RESISC-45 (Cheng et al., 2017) and DMLab (Beattie et al., 2016). For these smaller datasets we rescale the images to 224×224 resolution without preserving aspect ratio and we apply random horizontal flipping as data augmentation.

D Visualizing filters.

We visualize filters to qualitatively explore the differences between the features learned for models trained with adversarial or non adversarial self-supervised pre-training. We visualize the ViT embedding layer of the pre-training models which achieve the best average accuracy on ImageNet and its variants after fine-tuning nominally on ImageNet . We extract the first principal components of the standardized embedding weights. Then we reshape and rescale these principal components to $16 \times 16 \times 3$ RGB images which we plot in Figure 4. First, we notice that the filters (first row) for the different nominal pre-training methods are visually diverse. Interestingly, when combining these methods with adversarial training, we observe that filters learned with adversarial samples (second and third rows) are visually very different from the nominal filters (first row) and that these adversarial samples are much more similar among methods, especially between Supervised, RotNet and BYOL. When comparing the last two rows, we see that ℓ_∞ and ℓ_2 perturbations result in similarly looking embedding filters.



(a) Supervised - Nominal **(b)** RotNet - Nominal **(c)** MAE - Nominal **(d)** BYOL - Nominal

(e) Supervised - ℓ_∞ **(f)** RotNet - ℓ_∞ **(g)** MAE - ℓ_∞ **(h)** BYOL - ℓ_∞

(i) Supervised - ℓ_2 **(j)** RotNet - ℓ_2 **(k)** MAE - ℓ_2 **(l)** BYOL - ℓ_2

Figure 4: Visualizing filters. First 28 principal components of the embedding filters of ViT-B16 pre-trained nominally (first row) or adversarially (second and third rows) by various pre-training methods: Supervised, RotNet, MAE, and BYOL (columns from left to right).

E Transfer learning performance

Transfer learning details. For completeness we evaluate the transfer learning performance of the adversarial pre-training methods from ImageNet to smaller datasets. For all the pre-training methods and attack types we select as initialization the models that achieved the best average performance over ImageNet variants in the previous subsections. Regarding the optimization, we compare the masked autoencoder procedure of He et al. (2022) with AdamW and layer-wise learning rate decay which we used in the previous subsections and the transfer learning procedure proposed in Steiner et al. (2021) with SGD with momentum 0.9, a batch size of 512, gradient clipping at global norm 1, no weight decay, a total of 2500 training steps and a learning rate of 0.01 attained after a linear ramp-up of 500 steps followed by a cosine decay.

Results. We report the results in Table 1 where we observe that adversarial training consistently improves the transfer learning performance of the various pre-training methods with an improvement of the average accuracy of +0.27%, +1.95%, +0.86% and +0.47% for supervised pre-training, RotNet, MAE and BYOL respectively when changing from nominal to ℓ_2 attacked pre-training. Secondly, ImageNet supervised pre-training is the best performing method on all of the datasets except DMLab whereas BYOL and MAE achieve the highest accuracy on ImageNet and its variants, so there is no strict correlation between the fine-tuning performance on the pre-training dataset (here ImageNet) and other transfer datasets. Finally, while both optimizers perform similarly on average for supervised pre-training and BYOL, we observe that RotNet and MAE perform much better with AdamW and layer-wise learning rate decay. This indicates that these two methods benefit from preserving the early layers learned during the pre-training phase.

F Additional table and figure

Table 1: Transfer learning. We compare the transfer learning performance of models pre-trained with or without attacks on the four studied pre-training tasks. We select the pre-trained models that achieve the best average downstream performance over IMAGENET variants: supervised with $\epsilon_\infty = 6/255$ and $\epsilon_2 = 4$, RotNet with $\epsilon_\infty = 4/255$ and $\epsilon_2 = 4$, MAE with $\epsilon_\infty = 1/255$ and $\epsilon_2 = 0.5$ and BYOL with $\epsilon_\infty = 2/255$ and $\epsilon_2 = 2$. We evaluate two fine-tuning optimizers on several datasets (headers in green) and we report the average over datasets in the last rows (orange header). MAE-ℓ_2 performs the best among adversarial self-supervised methods and matches supervised adversarial pre-training without any labels.

SETUP	SUPERVISED			ROTNET			MAE			BYOL		
	Nominal	ℓ_∞	ℓ_2	Nominal	ℓ_∞	ℓ_2	Nominal	ℓ_∞	ℓ_2	Nominal	ℓ_∞	ℓ_2
CIFAR-10												
SGD	98.78%	98.97%	**99.08%**	94.79%	96.90%	96.79%	96.77%	97.78%	98.00%	97.72%	98.80%	98.96%
AdamW	98.55%	98.69%	**98.76%**	96.14%	97.29%	97.43%	98.03%	98.30%	98.46%	98.12%	98.69%	98.67%
CIFAR-100												
SGD	90.96%	91.60%	**92.53%**	78.65%	83.00%	83.48%	83.28%	86.39%	86.68%	86.91%	90.56%	91.10%
AdamW	90.71%	90.59%	**91.07%**	82.55%	85.22%	86.45%	88.15%	89.24%	89.74%	88.62%	90.61%	91.05%
SUN-397												
SGD	76.53%	74.26%	**76.62%**	59.79%	61.52%	63.95%	69.88%	69.98%	71.82%	74.82%	75.02%	75.31%
AdamW	77.59%	76.13%	**78.01%**	68.01%	69.09%	70.27%	76.39%	76.56%	77.31%	76.47%	75.65%	75.96%
RESISC-45												
SGD	**96.32%**	95.31%	96.17%	94.56%	94.29%	95.14%	94.69%	93.85%	95.30%	95.07%	95.15%	95.46%
AdamW	**96.94%**	96.52%	96.92%	95.15%	95.30%	95.93%	96.62%	96.44%	**96.94%**	96.44%	95.89%	96.08%
DMLAB												
SGD	73.63%	**74.56%**	74.49%	63.73%	69.01%	68.89%	70.22%	73.86%	73.76%	70.93%	72.90%	71.81%
AdamW	74.88%	74.61%	75.27%	66.80%	68.60%	68.31%	76.32%	76.90%	**77.34%**	72.13%	73.40%	72.39%
AVERAGE												
SGD	87.24%	86.94%	**87.78%**	78.30%	80.94%	81.65%	82.97%	84.37%	85.11%	85.09%	86.49%	86.53%
AdamW	87.73%	87.31%	**88.01%**	81.73%	83.10%	83.68%	87.10%	87.49%	87.96%	86.36%	86.85%	86.83%

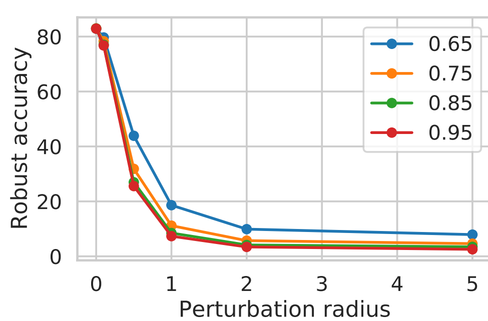


Figure 5: Influence of layer-wise learning rate decay on preserving robustness. We report the robust test accuracy on IMAGENET under ℓ_2 attacks with different perturbation radii of models nominally fine-tuned with various layer-wise learning rate decays. All the models are fine-tuned from the same network pre-trained using adversarial BYOL with $\epsilon_2 = 2$. Using a smaller layer-wise learning rate decay during fine-tuning leads to a higher robustness on the downstream classification task, thus preserving some of the robustness learned during pre-training.