**ADVERSARIALY SELF-SUPERVISED PRE-TRAINING IMPROVES ACCURACY AND ROBUSTNESS**

Sylvestre-Alvise Rebuffi, Olivia Wiles, Evan Shelhamer & Sven Gowal
DeepMind, London
{svyvestre,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 improve pre-training by attacking the label-free losses of self-supervised learning. That is, the
where pairs of samples \( x \) and labels \( y \) are sampled from the data distribution \( D \). \( L^\theta_\delta \) is a suitable loss function (such as the cross-entropy loss for classification tasks) using the output of the model parametrized by \( \theta \). \( S \) denotes the constrained space of perturbations. For \( \ell_p \) norm-bounded perturbations of size \( \epsilon \) the adversarial set of perturbations is defined as \( S_p = \{ \delta | ||\delta||_p \leq \epsilon \} \). In the rest of the manuscript we will use \( \epsilon_p \) to denote \( \ell_p \) norm-bounded perturbations of size \( \epsilon \) (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 \( \alpha \). For an arbitrary adversarial loss \( L^\theta \), we denote as \( \text{PGD}_K^\epsilon (x, y) \) the inner optimization with \( K \) steps defined as

\[
\delta^{(k+1)} \leftarrow \text{proj}_S \left( \delta^{(k)} + \alpha \text{sign} \left( \nabla_{\delta^{(k)}} L^\theta_\delta (x + \delta^{(k)}, y) \right) \right)
\]

where \( \delta^{(0)} \) is randomly sampled within \( S \), and where \( \text{proj}_A (a) \) projects a point \( a \) back onto a set \( A \), 
\[
\text{proj}_A (a) = \arg \min_{a' \in A} ||a - a'||_2.
\]

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 \( e(\cdot; \theta) \) that operates on the visible patches and a lightweight decoder \( d(\cdot; \theta) \).
that reconstructs the masked patches from the latent representation. Given an image $x$ we decompose it into a batch of patches $p$. We randomly sample a mask $m$ which is equal to 1 for a visible patch and to 0 for a masked out patch. We feed the patches $p$ and the mask $m$ 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:

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

where $\mu$ and $\sigma$ 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 $e(p, \theta)$ of fully visible images $x$ with patches $p$ 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 $p$ 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

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

where we use PGD$_{\text{A-MAE}}^K$ 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 $\ell_\infty$ or $\ell_2$ 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 $\ell_\infty$ attacks. On the contrary, the best adversarial MAE result with a $\ell_2$ attack is +0.67% better than the best average accuracy with a $\ell_\infty$ attack. Adversarial BYOL achieves similar
best average accuracy with both $\ell_\infty$ and $\ell_2$ 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 $\ell_\infty$ 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, $\ell_\infty$ or $\ell_2$). When comparing the $\ell_\infty$ and $\ell_2$ 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 $\ell_2$ 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 $\gamma^{-k}$ where $\gamma$ 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...
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 $\epsilon_{l} = 0.5$ and BYOL with $\epsilon_{v} = 2$. 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 $\epsilon_{v} = 2$. 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.

---

**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 $\ell_{\infty}$ attacks and with $\ell_{2}$ 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 $\epsilon_{l} = 0.5$ and BYOL with $\epsilon_{v} = 2$. The colours are row normalized and red means better.
REFERENCES


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.


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 $0^\circ$, $90^\circ$, $180^\circ$ or $270^\circ$ 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:

$$
\arg \min_{\theta} \mathbb{E}_{(x,y) \in D} \mathbb{E}_{y \sim \mathcal{U}([0,\ldots,270])} \left[ \max_{\delta \in \mathcal{S}} L_{CE}^{\theta}(\text{rot}(x, y) + \delta, y) \right] 
$$

where $L_{CE}$ is the cross-entropy loss, the label $y$ corresponds to a rotation randomly sampled among the four possible rotations $0^\circ$, $90^\circ$, $180^\circ$ and $270^\circ$ degrees and $\text{rot}(x, y)$ is the function which rotates the sample $x$ 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 $e(\cdot; \theta)$, a projector $g(\cdot; \theta)$ and a predictor $q(\cdot; \theta)$. We denote by $\gamma = g \circ q \circ e$ 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 $\xi$, an exponential moving average of the weights $\theta$. Given an image $x$, and two augmentations $t, t' \sim T$ sampled from a set of augmentations BYOL produces two augmented views $v = t(x)$ and $v' = t'(x)$. The first view passes through the online network, producing a representation $h = e(x; \theta)$ and a projection $z = g(h; \theta)$. The second view similarly passes through the target network, producing a target projection $z' = \gamma(v'; \xi)$. Finally, given an online prediction $q(z; \theta) = \kappa(v; \theta)$ (which should be predictive of the target projection), BYOL minimizes the loss

$$
L_{BYOL}^{\theta} \propto -\frac{\kappa(v; \theta)^T \gamma(z')}{\|\kappa(v; \theta)\|_2 \cdot \|z'\|_2}. 
$$

At the end of training, everything but $e$ and $\theta$ is discarded and only the representation $e(x; \theta)$ of an image $x$ 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 $v = t(x)$ and $v' = t'(x)$ of the same image $x$. Now, while the second view goes through the target network unmodified to produce a target projection $z' = \gamma(v'; \xi)$, 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 $\kappa(v + \delta; \theta)$ and target projection $z'$:

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

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 $v'$ to the online network and $v$ to the target network.
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 radiiuses when pre-training on adversarial samples: $\ell_{\infty}$-bounded attacks with radius $\epsilon \in \{1/255, 2/255, 4/255, 6/255, 8/255\}$ and $\ell_2$-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 $224 \times 224$ 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 $224 \times 224$ 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 $\ell_{\infty}$ and $\ell_2$ perturbations result in similarly looking embedding filters.
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 I 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 $\ell_2$ 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 $\text{IMAGENET}$ 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-$\ell_2$ performs the best among adversarial self-supervised methods and matches supervised adversarial pre-training without any labels.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Supervised</th>
<th>RotNet</th>
<th>MAE</th>
<th>BYOL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\ell_\infty$</td>
<td>$\ell_2$</td>
<td>$\ell_\infty$</td>
<td>$\ell_2$</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>98.78%</td>
<td>98.97%</td>
<td>99.00%</td>
<td>94.79%</td>
</tr>
<tr>
<td>AdamW</td>
<td>98.55%</td>
<td>98.69%</td>
<td>98.76%</td>
<td>96.14%</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>90.96%</td>
<td>91.60%</td>
<td>92.51%</td>
<td>78.65%</td>
</tr>
<tr>
<td>AdamW</td>
<td>90.71%</td>
<td>90.59%</td>
<td>91.07%</td>
<td>82.55%</td>
</tr>
<tr>
<td>SUN-397</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>76.53%</td>
<td>74.26%</td>
<td>76.62%</td>
<td>59.79%</td>
</tr>
<tr>
<td>AdamW</td>
<td>77.59%</td>
<td>76.13%</td>
<td>78.01%</td>
<td>68.01%</td>
</tr>
<tr>
<td>RESISC-45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>96.32%</td>
<td>95.31%</td>
<td>96.17%</td>
<td>94.56%</td>
</tr>
<tr>
<td>AdamW</td>
<td>96.94%</td>
<td>96.52%</td>
<td>96.92%</td>
<td>96.92%</td>
</tr>
<tr>
<td>DMLAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>73.63%</td>
<td>74.56%</td>
<td>74.49%</td>
<td>63.73%</td>
</tr>
<tr>
<td>AdamW</td>
<td>74.88%</td>
<td>74.61%</td>
<td>75.27%</td>
<td>66.80%</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>87.24%</td>
<td>86.94%</td>
<td>87.78%</td>
<td>78.30%</td>
</tr>
<tr>
<td>AdamW</td>
<td>87.73%</td>
<td>87.31%</td>
<td>88.01%</td>
<td>81.73%</td>
</tr>
</tbody>
</table>

Figure 5: Influence of layer-wise learning rate decay on preserving robustness. We report the robust test accuracy on $\text{IMAGENET}$ under $\ell_2$ 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.