DILEMMA: Self-Supervised Shape and Texture Learning with Transformers

Sepehr Sameni
Institute of Computer Science
University of Bern
sepehr.sameni@inf.unibe.ch

Simon Jenni
Adobe Research
jenni@adobe.com

Paolo Favaro
Institute of Computer Science
University of Bern
paolo.favaro@inf.unibe.ch

Abstract

There is a growing belief that deep neural networks with a shape bias may exhibit better generalization capabilities than models with a texture bias, because shape is a more reliable indicator of the object category. However, we show experimentally that existing measures of shape bias are not stable predictors of generalization and argue that shape discrimination should not come at the expense of texture discrimination. Thus, we propose a pseudo-task to explicitly boost both shape and texture discriminability in models trained via self-supervised learning. For this purpose, we train a ViT to detect which input token has been combined with an incorrect positional embedding. To retain texture discrimination, the ViT is also trained as in MoCo with a student-teacher architecture and a contrastive loss over an extra learnable class token. We call our method DILEMMA, which stands for Detection of Incorrect Location EMbeddings with MAsked inputs. We evaluate our method through fine-tuning on several datasets and show that it outperforms MoCoV3 and DINO. Moreover, we show that when downstream tasks are strongly reliant on shape (such as in the YOGA-82 pose dataset), our pre-trained features yield a significant gain over prior work. Code will be released upon publication.

1 Introduction

In computer vision, deep learning models trained on small labeled datasets can benefit greatly from pre-training on labeled datasets such as ImageNet [34]. Even more surprisingly, MoCo [37] showed that it is possible to pre-train with unlabeled data and outperform pre-training with supervised learning on several downstream tasks. This led to the rapid development of several Self-Supervised Learning (SSL) methods [6, 8, 11, 37].

Representations obtained via SSL have the ability to generalize to downstream tasks such as object classification, detection, and segmentation [18, 27, 46]. Recent work suggests that representations with a shape bias generalize better than those with a texture bias [32, 64]. Such connection takes inspiration from developmental psychology studies, which have shown that children around the age of two prefer to rely more on shape rather than texture to classify novel objects [47]. This follows the philosophy where understanding deep learning models may help to understand mechanisms of human vision and vice versa [31]. However, as we discuss later and show in our experimental evaluation, the ability to generalize on shape-based tasks may not need to suppress texture discrimination.
The analysis of how much deep neural networks are biased towards shape or texture is still being debated [32, 64]. Geirhos et al. [32] propose new datasets and evaluation methods to measure the degree by which a trained model is biased more towards the shape or the texture of an object. Similarly, Tartaglini et al. [64] propose alternative versions of these datasets and experiments to evaluate shape bias as done in developmental psychology. These measures of shape bias are proposed as gauges of how well a pre-trained model can be transferred to other tasks. However, transfer learning methods may give unpredictable outcomes. We find that a model with a lower initial shape bias than another one, as measured in [32, 64], may do better in the transfer learning to a shape-based task, such as Yoga\textsubscript{82} (see Fig. 1). Thus, we suggest that predicting how well pre-trained models generalize to shape-based tasks may not have to trade off texture for shape discrimination or show a stronger bias towards shape.

Given the importance of texture discrimination in many tasks, and in particular in problems with natural categories, we propose a method to boost both shape and texture discriminability. Shape differentiation is encouraged by training a model to detect the correct positioning of object parts, a concept already proposed in the context prediction [20] and jigsaw puzzle SSL methods [55]. We also take inspiration from Electra [15], where some text tokens are replaced by a weak generator and a discriminator is trained to detect them, and the spotting artifacts method [43], which does the same in the visual domain. Texture discrimination is encouraged instead via a contrastive loss [30] as in MoCoV3 [13].

In our method (see Fig. 2), we split an image into a grid of tiles, map them to tokens, combine them with positional embeddings and then feed them to a ViT [22]. However, we corrupt the positional embeddings of a fraction of the tokens. Then, we train the ViT to classify the tokens into those with correct and incorrect positional embeddings. In this way, the pseudo-task of detecting corrupted locations forces the discriminator to learn meaningful associations between the texture in a token and its location relative to the other tokens.

The training of SSL methods and transformers poses a major challenge with its demanding computational requirements. One practical technique to reduce the computational load, which we adopt, is to sparsify the input tokens as in VATT [1] and MAE [36]. This technique serves also a second purpose: If we fed all tiles as input, the ViT could detect incorrect positional embeddings just by using the tile borders. This would not lead to the learning of shape-discriminative features. We avoid this behavior thanks to sparsity, because it is unlikely that the selected input tokens are adjacent.

We also use a teacher-student architecture as in MoCoV3 [13]. We sparsify only the input to the student network and instead feed all the tiles to the teacher network, since it is used only in evaluation mode and it does not have a significant impact on storage and computing resources. Moreover, the use of a complete set of tiles (a setting that we call dense) and without corrupted positional embeddings, allows the teacher to build a better reference for the student network. We call our method DILEMMA, which stands for “Detection of Incorrect Location EMbeddings with MAsked inputs.”

**Our contributions can be summarized as follows:**

- We introduce a novel SSL method that enhances the shape discriminability of features without sacrificing the ability to represent texture; it is based on the detection of misplaced positional embeddings with a ViT, and a contrastive loss as in MoCo.
• We show that current shape bias measures are not stable predictors of performance on shape-based downstream tasks.
• We propose to randomly sparsify the inputs to: 1) speed up the training, 2) reduce the memory usage, 3) close the gap between training and test times, 4) avoid degenerate learning
• We surpass the performance of MoCoV3 and DINO under the same computational budget.

2 Related Work

Self-Supervised Learning. Self-supervised learning gained popularity as a form of unsupervised learning where pretext tasks leverage supervision signals obtained automatically without human labor. Some classic examples are the classification of image patch locations [20, 55], the reconstruction of color channels [79] or image patches [59], or the recognition of various image transformations [33, 43, 44]. While prior patch-based methods inspired our approach of detecting wrongly placed image patches, ours is both simpler and performs better in transfer experiments. Furthermore, due to the input representation of ViTs (disjoint image patches) and our random sparse patch sampling, our approach suffers less from shortcuts or domain gaps between pre-training and transfer.

Contrastive Learning. Efforts to scale up and improve instance discrimination [23, 72] as a self-supervised pre-training task have established contrastive learning [8, 37, 57] as the most popular SSL approach in computer vision today. Several modifications of the basic recipe, i.e., learning to discriminate training instances up to data augmentations, have been proposed since. For example, some methods leverage momentum encoded samples for positive and negative sampling [37, 11], some remove the need for explicit negative pairs [35, 12], and others extend the set of positives beyond data-augmentation through clustering [6] or nearest-neighbors in feature space [24]. Another line of work considers contrastive pre-training strategies tailored to dense prediction tasks [56, 71, 76, 78, 49, 51]. More recently, state-of-the-art contrastive methods leverage novel vision transformer architectures [22, 52], e.g., by adapting existing approaches [13, 77], tailoring architectures [48], or novel objectives [7]. In our approach, we improve upon a well-established contrastive baseline [13] through the addition of a spatial reasoning task and by extending the set of image augmentations through randomized patch dropping.

Self-Supervised Pre-Training of Transformers. The success of the transformer architecture [67] in natural language is to a great extent due to large-scale self-supervised pre-training tasks. Successful pre-training strategies from NLP like masked token prediction [19] have recently also been adapted to the image domain [2, 81, 36, 81]. Our patch misplacement detection is similar to another type of pretext task in NLP, where the goal is to detect corrupted tokens, i.e., words replaced by an imperfect masked language model [15, 16]. However, a key difference in our approach is that we only tamper with the spatial position of the tokens and thus do not require a separate masked token prediction model. In parallel work, Fang et al. [28] use BEiT [2] for that purpose. The method of DABS [63] also uses the idea of patch misplacement, but it does not have a way to handle degenerate learning and it does not show performance improvements. A technique that has proven very beneficial to improve the training efficiency of vision transformers is token dropping [1, 36, 25, 10]. We extend this technique by randomizing the token dropping amount and including the case of no dropping to narrow the domain gap between pre-training and transfer.

3 Training DILEMMA

So far, the analysis of features obtained through representation learning has indirectly promoted the idea that features with a strong shape bias may have to forgo texture discriminability. For example, datasets used to evaluate shape bias are built under the premise that features should be “prone” to texture invariance, i.e., they could be easily fine-tuned to artificial datasets, where class discrimination is only based on the class shapes [32, 64]. However, texture carries also important class information (e.g., the zebra, jaguar or panda color patterns are very distinctive of their category). Thus, we advocate that shape discriminability should be boosted, while not sacrificing the ability to distinguish texture. To achieve this goal we introduce DILEMMA, a novel self-supervised learning method.
DILEMMA uses two fundamental losses, which we introduce in detail in section 3.1. We illustrate our training method in Fig. 2.

Let us define an image sample as \( x \in \mathbb{R}^{H \times W \times C} \), i.e., \( x \) has \( H \times W \) pixels and \( C \) color channels. We apply two data augmentations [35] to \( x \) and obtain \( \tilde{x}_1 \) and \( \tilde{x}_2 \). Similarly to ViT, each input \( \tilde{x}_1 \) and \( \tilde{x}_2 \) is divided in \( 14 \times 14 \) tiles, flattened and projected to \( N \) tokens \( t_{1,i}, t_{2,i} \in \mathbb{R}^D, \forall i \in U \subseteq \{1, \ldots, N\} \), through a linear projection. We then combine each token \( t_{i,:} \), with a positional embedding \( p_i \in \mathbb{R}^D \), which can be either learned or fixed.

As in MoCoV3 [13], we define a Student \( S \) and a Teacher \( T \) ViTs [22], where the Teacher, also called momentum encoder, is obtained through the exponential moving average (EMA) of the Student’s weights (thus, it is not trained). The Teacher receives as input all the tokens \( t_{1,i}, \ldots, t_{1,N} \) with the corresponding positional embeddings \( p_{1}, \ldots, p_{N} \). The Student instead receives as input a sparse set \( M \subseteq U \) of tokens \( t_{2,i} \), \( i \in M \). For a randomized fraction of these tokens \( B \subseteq M \) the corresponding positional embeddings \( q_i, i \in M \) are incorrect, i.e., \( q_i = p_i \) if \( i \in M \setminus B \) and \( q_i = p_j \) with \( j \in U \setminus M \), if \( i \in B \). We call the ratio \( \theta = |B|/|M| \in [0,1] \), between the cardinalities of \( B \) and \( M \), the probability of a positional embedding mismatch. We choose a different \( M \) and \( B \) sets for each sample at each iteration. We define a set of ground truth labels \( y_i = 0 \) (N) if \( i \in M \setminus B \) and \( y_i = 1 \) (Y) if \( i \in B \). The \( i \)-th output token from the Student is denoted with \( S_i(\{q_j \oplus t_{2,j}\}_{j \in M}) \). We indicate the extra classification token with \( i = 0 \) both at the input and output. Also, \( q_0, p_0 = 0 \), i.e., no location encoding.

Because of the sparsity in the input to the Student network, we also obtain a computational benefit. When we increase the sparsity of the input, we can also increase the mini batch size to fully utilize the GPU RAM. This is particularly significant with ViTs, because of their quadratic scaling with the number of tokens (the memory usage is \( O(N^2) \)). The fact that we can significantly increase the mini batch size is particularly effective with contrastive learners. Moreover, in this way it is also faster to train our model, because the average mini batch size is much larger than when using dense inputs (in our case it’s \( 2.5 \times \) more).

3.1 Enhancing both Shape and Texture Discriminability

We train our model (the Student network) to have both a shape and texture discriminability by using two losses: one is the location-congruence classification of tokens and the other is a contrastive loss.
between the student extra class token (CLS) and the corresponding class token of the Teacher. The first loss focuses on the shape, while the second focuses more on the texture [30]. The first loss can be described as

$$L_{\text{BCE}} = E_x \left[ \sum_{i \in M} y_i \log \left( \sigma(W_S i \{\{q_j \oplus t_{2,j}\}_{j \in M \cup \{0\}}\}) \right) \right] + (1 - y_i) \log \left( 1 - \sigma(W_S i \{\{q_j \oplus t_{2,j}\}_{j \in M \cup \{0\}}\}) \right),$$

where $E[.]$ is the expectation over image samples, $\sigma$ is the sigmoid function and $W$ is a linear projection. The second loss is instead the contrastive loss

$$L_{\text{CNT}} = E_x \left[ L_{\text{CE}} \left( S_0 \{\{q_j \oplus t_{2,j}\}_{j \in M \cup \{0\}}\}, T_0 \{\{p_j \oplus t_{1,j}\}_{j = 0, \ldots, N}\} \right) \right],$$

where

$$L_{\text{CE}} (A, V) = -2r \sum_n z_n \log \frac{A_n V}{\tau} \tag{3}$$

and $A$ and $V$ are $G \times m$ matrices, with $m$ the minibatch size and $G$ the vector size after the projection $W$; $z_j$ is the one-hot vector with 1 at the $j$-th position and the index $n$ indicates the class token within the minibatch.

Finally, we combine both losses into a single cost

$$L_{\text{DILEMMA}} = \lambda_{\text{DILEMMA}} L_{\text{BCE}} + L_{\text{CNT}},$$

which we minimize and where $\lambda_{\text{DILEMMA}} > 0$ is a tuning parameter.

As pointed out in Electra [15], we believe that our per-token positional-congruence loss $L_{\text{BCE}}$ provides a richer “dense” feedback to the model during training, as opposed to when only the CLS token is used in the contrastive loss, and this might help convergence. As in MoCoV3 we use a symmetrized loss.

### 3.2 Implementation

**Architecture.** We use Vision Transformers (ViT) [22] with a patch size of 16 x 16 pixels and an input image size of 224 x 224 pixels, which gives a total of $(224/16)^2 = 196$ tokens. Due to computational limitations, we only use the small variant of the Vision Transformer (ViT-S) which has 12 transformer blocks and 384 channels. Following MoCoV3 [13] we use 12 attention heads in each attention layer. This is different from most ViT-S implementations, which use 6 heads. This does not change the total number of parameters of the model, but incurs a slight speed penalty. We use a 3-layer MLP for the projection and prediction heads with synchronized batch normalization. We also freeze the weights of the patch embedding layer for better stability.

**Pre-training Setup.** We pre-train DILEMMA on ImageNet-1K [18] with the exact same hyper-parameters (including learning rate, learning rate scheduler, optimizer, and warm-up epochs) of MoCoV3 using three GeForce RTX 3090 GPUs for 100 epochs with a base batch size of 345. We set the $\lambda_{\text{DILEMMA}}$ to 0.4 and the probability of positional embedding mismatch $\theta = 0.2$. We use sparsity ratios of 0%, 40%, 55%, 65% with 1×, 2×, 3×, 4× base batch size and disable the DILEMMA loss when the input is dense. To compare with DINO [7], we trained a DINO network with a batch size of 480 for 100 epochs without multi-cropping. For the sake of completeness we also trained a DILEMMA for 150 epochs which takes the same amount of time as training MoCoV3 for 100 epochs.

**Linear Probing.** To evaluate the pre-trained features for image classification, we train a simple linear layer on top of frozen features, without any data augmentation (Linear$_F$). Note that it is different from the standard linear probing, and we opt to use this method for its simplicity and speed. It is also more aligned with the end goal of representation learning. In all the linear probing experiments, we use the embedding of the CLS token of the last layer (unlike in DINO [7], which uses the CLS token of the last four attention layers of the network and concatenates them) and perform a coarse grid search over learning rates, batch sizes and whether to normalize the data before feeding them to the linear layer or not (similarly to the added BatchNorm layer [41] in MAE [36]).
4 Experiments

We evaluate DILEMMA on several datasets, compare it to existing methods in SSL, perform ablations to show the role of each loss function and analyze its shape vs texture bias. We refer to MoCoV3 [13] as our baseline.

4.1 Classification on ImageNet-1K

**k-NN and Linear Probing.** In order to evaluate the quality of the pre-trained features, we either use a weighted $k$ nearest neighbor classifier (we always use $k = 20$) [73] or a simple linear layer on top of a frozen backbone and frozen features. In Table 1, DILEMMA outperforms the base model by 1.5% after 2/3 of the training time. It also outperforms DINO in similar settings when multi-crop training is disabled (multi-crop training could also improve the results of MoCoV3 and DILEMMA, but this is beyond the focus of this work). We also show DILEMMA trained for the same amount of time as MoCoV3 in gray and denote it with the ($\uparrow$) symbol. In either case, DILEMMA shows a consistent and significant improvement. Note that Table 1 shows all the results from other methods with their reported numbers. Reported numbers for the linear accuracy are based on a linear layer trained with data augmentation.

**Low-shot learning.** To simulate transfers to small datasets, we use the model pre-trained on the whole unlabeled ImageNet dataset and then train a linear layer on top of the frozen features of the 1% or 10% subsets [8] of ImageNet and then evaluate the results on the whole ImageNet validation set. Results in Table 2 show that DILEMMA is more label efficient than MoCoV3. Notice that in this implementation DILEMMA is based on MoCoV3, which, as was observed in DINO [7], has a consistently worse $k$-NN accuracy than DINO. Nonetheless, DILEMMA is able to almost compensate for the gap deficit.

**ImageNet-*.** To measure the out of distribution (OOD) generalization of the representation, we use a linear layer trained on the non augmented training set of ImageNet-1K and evaluate it on different OOD datasets. Results in Table 3 show that DILEMMA has a better OOD generalization than the baseline.

4.2 Downstream Tasks

**Semantic Segmentation on ADE20K.** Semantic segmentation is a task that strongly relates to the shape of objects. Thus, we expect to see a significant improvement from a boost in the shape discriminability. The semantic segmentation capability of self-supervised methods is usually
Table 4: Semantic Segmentation on ADE20K

| Method       | Seg. w/ Lin. | Seg. w/ UPerNet |
|--------------|--------------|-----------------|
|              | mIoU mAcc aAcc | mIoU mAcc aAcc   |
| MoCoV3       | 11.23 14.56 65.31 | 33.80 44.71 77.65 |
| DILEMMA      | 15.90 20.08 67.46 | 33.97 44.73 77.95 |

Table 5: Transfer learning for image classification

| Dataset          | MoCoV3 | DILEMMA |
|------------------|--------|---------|
| Aircraft         | 23.82  | 24.18   |
| Caltech101       | 79.11  | 80.72   |
| Cars             | 19.67  | 19.85   |
| CIFAR10          | 90.87  | 90.60   |
| CIFAR100         | 70.98  | 62.60   |
| DTD              | 59.84  | 58.97   |
| Flowers102       | 81.40  | 84.45   |
| Food101          | 57.89  | 58.97   |
| INat             | 74.22  | 78.82   |
| Pets             | 90.87  | 94.75   |
| STL10            | 32.73  | 38.41   |
| Yoga82           | 59.29  | 61.14   |

Evaluating by fine-tuning the model with an extra decoder. For that we use UPerNet [75] on the ADE20K [80] dataset and train the model for 64K iterations with a batch size of 12. We also follow the evaluation protocol of iBOT [81] and just train a linear layer (for 64K iterations and a batch size of 16) for semantic segmentation with a frozen backend to directly assess the per token representation. Results in Table 4 show that DILEMMA is also better than the base model for dense classification tasks and yields a remarkable mIoU gap of 4.6 percentage points between DILEMMA and MoCoV3 in the linear settings.

Transfer Learning. In order to evaluate the transfer capability of our representations we do image classification on a diverse set of datasets. We again train a linear layer on top of the frozen features to accelerate the process. The results are in Table 5. As can be seen DILEMMA performs well in transfer learning across all datasets and significantly outperforms the base model in Yoga82 [68] (a yoga position classification dataset). Correctly classifying Yoga82 images requires a solid understanding of object shape and texture alone is not sufficient.

Nearest Neighbor Retrieval. Following the evaluation protocol of DINO [7], we report the Mean Average Precision (mAP) for the Medium (M) and Hard (H) subsets of revisited Oxford and Paris retrieval datasets [61] using a k-NN retriever. Results in Table 6 show that our results are comparable to the baseline, but, as it is explained in iBOT [81], this evaluation metric is hyper-parameter sensitive (image resolution, whether to use multi-scale evaluation or not) and we do not investigate it further.

Unsupervised Object Segmentation. For single frame object segmentation we use the mask generated from the attention of the CLS token (thresholded to keep 0.9 of the mass) as in DINO [7] and report the Jaccard similarity between the ground truth and the mask evaluated on the validation set of PASCAL-VOC12 [27]. For the videos we use the DAVIS-2017 video instance segmentation benchmark [60] and by following the protocol introduced in Space-time by Jabri et al. [42] we segment scenes via the nearest neighbor propagation of the mask. Results in Table 7 show that DILEMMA performs well also in these dense tasks.

4.3 Shape vs Texture Bias

We also evaluate the shape bias of DILEMMA according to the metrics defined by Geirhos et al. [32] and Tartaglini et al. [64]. In Table 8 (Texture Bias column) we fine-tune our ViT model pre-trained with DILEMMA on ImageNet and then measure the shape vs texture bias on the Cue-Conflicting dataset [32]. The reported Texture Bias indicates how often the model has preferred class discrimination based on texture rather than shape. The Linear Accuracy is obtained through fine-tuning on half of the Cue-Conflicting dataset and then tested on the other half. In Fig. 3 we use instead a dataset [68], where the parameter Alpha indicates the degree of removal of the background (0 no removal, 1 full removal). These metrics could be used as predictors of the quality of pre-trained models. This would allow the ranking of models without the need for extensive experimental validation, which now requires the transfer to several downstream tasks. However, the results in
Table 6: Image retrieval. mAP on medium and hard subsets of the revisited Oxford and Paris retrieval datasets [61]

| Method    | ROX  | RPar |
|-----------|------|------|
| MoCoV3    | 26.02| 7.16 |
| DILEMMA   | 26.14| 7.26 |

Table 7: Unsupervised object segmentation. For DAVIS we report mean region similarity $J_m$ and mean contour-based accuracy $F_m$. For VOC12 we report Jaccard similarity $Jac_{sim}$.

| Method   | (J & F)_m | $J_m$ | $F_m$ | $Jac_{sim}$ |
|----------|------------|-------|-------|-------------|
| MoCoV3   | 58.28      | 57.46 | 59.09 | 46.50       |
| DILEMMA  | 60.00      | 57.99 | 62.02 | 48.89       |

Table 8: Shape Bias. The Texture Bias is evaluated on the Cue-Conflicting dataset [32]. The Linear Accuracy is obtained through fine-tuning on half of the Cue-Conflicting dataset and then tested on the other half.

| Method   | Texture Bias (%) | Linear Acc. |
|----------|------------------|-------------|
| MoCoV3   | 63.58            | 59.89       |
| DILEMMA  | 69.10            | 62.68       |

Figure 3: Shape bias Analysis [64]. Alpha indicates the transparency of the background (1 means full transparency).

Table 8 and Fig. 3 seem to contradict this conclusion. In fact, DILEMMA appears to be more texture biased than the baseline, but its performance on a wide range of new tasks and in particular on shape-based tasks seems to be consistently better (see all other experiments). In fact, in the Linear Accuracy column we see that as soon as DILEMMA is trained on a shape-based task, it delivers a better performance than the baseline. We argue that perhaps the ability of a model to classify objects based on shape does not necessarily imply that they must have a weaker texture discriminability. In conclusion, to have a better predictor for generalization based on shape, it would be more desirable to have a shape discriminability measure that is somehow unrelated to texture.

4.4 ViT Properties

In this section, we carry out experiments that are unique to the ViT architecture.

Robustness against Occlusions and Shuffle. Since our model was trained with sparse inputs, we should expect a gradual loss of performance with increased sparsity. Fig. 4 shows that the performance drop for MoCoV3 is more severe than in DILEMMA. Moreover, DILEMMA is able to preserve its accuracy with larger sparsity ratios. In Table 9 we see that DILEMMA can be fed with completely wrong position encodings and still obtain a reasonable classification accuracy compared to MoCoV3. We explain this result as the consequence of training the model with mismatched positional embeddings.

Robustness against Background Change. Following the background challenge evaluation metric [74], we compute the classification accuracy of the model on a subset of ImageNet (IN-9) by changing the background and foreground. As shown in Table 10, in O/N.F. (Only/No Foreground), M.S/R/N. (Mixed Same/Random/Next), where the foreground is visible or accurately masked out, we outperform the base model. When the foreground is not visible (O.BB. (Only Background with foreground box Blacked out) and O.BT. (Only Background with foreground replaced with Tiled background)) the model performs correctly and does not just rely on the background for image classification.
4.5 Ablations

Ablation studies are conducted either on ImageNet100 or ImageNet-1K. For the smaller dataset we train the dense models for 300 epochs and the sparse models for 450 epochs (with the same hardware and time settings). For ImageNet-1K experiments we train all models for 50 epochs unless stated otherwise.

**Image Size.** We compare a model trained on $112 \times 112$ images and a sparse model trained on 25% of the tokens of $224 \times 224$ images. Results in Table 11 show that simply feeding smaller images is worse than feeding sparse large inputs.

**Token Dropping Policy.** We tried dropping the tokens that were less important based on the attention of the teacher network [50] compared to randomly dropping the tokens. Results in Table 12 show that simple random dropping works well and there is no need to introduce extra complexity to the model.

**Random Dropping Ratio.** To verify that a random dropping ratio is better than a constant one, we conducted two experiments: one on IN100 and one on IN-1K. The results in Table 13 show that a random dropping ratio performs better than a constant one. On the more difficult IN-1K dataset just applying the sparsity is worse than using the dense model. Only with a random dropping ratio the sparse model can outperform the dense model.

**Mismatch Detection.** To verify that mismatch detection helps, we trained a dense model with the mismatch detection task. Results are in Table 14. Surprisingly, even though this is a trivial task (note that since only 20% of the positions are mismatched, a model can easily achieve an 80% accuracy in detecting the mismatches), the dense model can still improve the performance of the model. The performance improvement for a task like in Yoga$_{82}$, which requires a better understanding of shape, is quite significant.

**Skip Dense.** In order not to encourage the model to use the tile edge shortcut, we disabled the calculation of the DILEMMA loss when the sparsity ratio is zero (the input is dense). Table 15 shows that this choice helps (though only marginally). Thus, we use it in all of our main results.

**DILEMMA Variants.** We also tried some variants of DILEMMA. Instead of just detecting the misplaced tokens, we predict the right position (as a classification task of 196 classes). The other variant, which we call **Partial Jigsaw**, is to feed some tokens without position encoding and ask

---

**Figure 4: Token Dropping Analysis.** Classification accuracy of a pre-trained head as a function of input sparsity ratio.

| Method   | Correct | Random |
|----------|---------|--------|
| MoCoV3   | 63.8    | 26.1   |
| DILEMMA  | 65.3    | 45.1   |

**Table 9: Input shuffling effect.** Classification accuracy of a pre-trained head with correct or random position encodings.

| Method   | O.F.(↑) | M.S.(↑) | M.R.(↑) | M.N.(↑) | N.F.(↑) | O.BB.(↓) | O.BT.(↓) | IN-9(↑) |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| MoCoV3   | 77.26   | 78.05   | 64.96   | 64.07   | 38.02   | 9.36    | 10.72   | 91.53   |
| DILEMMA  | 77.75   | 79.43   | 67.63   | 64.84   | 38.79   | 8.64    | 9.33    | 91.75   |

**Table 10: Robustness of pre-trained models against background changes.**

**Table 11:** Performance of DILEMMA on ImageNet-1K with different image sizes.

---
the network to predict their position given the other (sparse) correctly position-encoded tokens. Lastly, instead of corrupting the position, one can corrupt the content of a patch. Instead of using complex methods like inpainting we simply horizontally flip some of the patches and use the binary cross-entropy as our loss. Table 16 shows that even though all of these methods do help in terms of shape discrimination, DILEMMA is the one with the best performance both on IN-1K and Yoga82.

Mismatch Probability. The probability of mismatch $\theta$ is one of the most important hyper-parameters of DILEMMA. Early in our experiments, we found out that 20% is much better than 15%. In Table 17 we show that 30% yields worse performance than the default 20%.

Timing. We measure the time for an epoch of pre-training on ImageNet100 with three GeForce RTX 3090 GPUs and the maximum batch size possible. Note that we multiply the number of batches proportional to the sparsity ratio and the reported number is for dense batches. Results in Table 18 show that DILEMMA is $1.5 \times$ faster than MoCoV3 due to a larger average batch size.

5 Conclusions

We introduced a novel SSL method based on a location classification pseudo-task and a contrastive loss. We showed that awareness of the relative location of tiles of the input image is important for generalization and in particular when fine-tuning on shape-based downstream tasks. We observe also that current indicators of shape bias for pre-trained models may not always be predictive of the performance of such models on novel shape-based tasks. Our method is based on the ViT architecture. We introduce sparsity in the input (i.e., dropping image tiles), to both speed up the training and also to avoid trivial degenerate learning.

Acknowledgments and Disclosure of Funding

This work was supported by grant 200020_188690 of the Swiss National Science Foundation.
Table 15: Skip Dense. Results are evaluated on IN100

| k-NN Linear | No Difference | Skip on Dense |
|-------------|---------------|---------------|
| 76.04       | 76.34         | 80.56         |

Table 16: Variants of the loss. Although the variants improve the performance wrt the base dense model, DILEMMA is the most effective one

| Task                | k-NN Linear | k-NN Linear |
|---------------------|-------------|-------------|
| Pos. Correction     | 54.77       | 58.95       |
| Partial Jigsaw      | 55.72       | 59.19       |
| Flip Detection      | 55.69       | 59.59       |
| DILEMMA             | 55.63       | 59.84       |

Table 17: Mismatch Probability

| IN-1K | Yoga82 |
|-------|--------|
| Prob. | k-NN Linear | k-NN Linear |
| 0.3   | 55.34  | 59.79  |
| 0.2   | 55.63  | 59.84  |

Table 18: Training Times

| Method | Architecture | EffectiveEpoch | BatchSize | k-NN Linear | Time(Sec.) | Batch Size |
|--------|--------------|----------------|-----------|-------------|------------|------------|
| DINO   | ViT-S        | 100x2          | 128       | 57.9        | 218        | 480        |
| MoCoV3 | ViT-S        | 100x2          | 345       | 60.1        | 335        | 345        |
| DILEMMA| ViT-S        | 100x2          | 345       | 61.7        | 223        | 345        |
| MoCoV3 | ViT-S        | 150x2          | 345       | 63.4        | -          | -          |
| DINO   | ViT-S        | 100x2          | 480       | 61.6        | -          | -          |
| DINO   | ViT-S        | 100x2          | 512       | 59.6        | -          | -          |
| DINO   | ViT-S        | 100x2          | 1024      | 59.9        | -          | -          |
| SimCLR | ViT-S        | 300x2          | 1024      | 69.0        | -          | -          |
| SwAV   | ViT-S        | 300x2          | 1024      | 66.6        | -          | -          |
| SwAV   | ViT-S        | 300x3.1        | 1024      | 64.7        | -          | -          |
| MoBY   | ViT-S        | 300x2          | 512       | 72.8        | -          | -          |
| MoCoV2 | ViT-S        | 300x2          | 1024      | 62.0        | -          | -          |
| MoCoV2 | ViT-S        | 300x3.1        | 1024      | 65.4        | -          | -          |
| MoCoV3 | ViT-S        | 300x2          | 1024      | 72.5        | -          | -          |
| MoCoV3 | ViT-S        | 4096           | 73.2      | 81.4        | -          | -          |
| MoCoV3 | ViT-S        | 600x2          | 1024      | 73.4        | -          | -          |
| TWIST  | ViT-S        | 300x3.1        | 1024      | 76.3        | -          | -          |
| DINO   | ViT-S        | 300x2          | 1024      | 67.9        | -          | -          |
| DINO   | ViT-S        | 100x3.5        | 512       | 69.3        | -          | -          |
| DINO   | ViT-S        | 300x3.1        | 1024      | 72.7        | -          | -          |
| DINO   | ViT-S        | 300x3.5        | 1024      | 73.3        | -          | -          |
| DINO   | ViT-S        | 800x3.8        | 1024      | 74.5        | -          | -          |
| DINO   | ViT-S        | 800x3.8        | 1024      | 74.5        | -          | -          |
| IBOT   | ViT-S        | 800x2          | 1024      | 72.4        | -          | -          |
| SplitMask| ViT-S      | 300x2          | 1024      | 75.2        | -          | -          |
| CIM    | ViT-S        | 300x1          | 2048      | -           | -          | -          |
| BEIT   | ViT-S        | 300x1          | 1024      | -           | -          | -          |
| CAE    | ViT-S        | 300x1          | 2048      | 50.8        | -          | -          |
| Supervised| ViT-S   | 300x1          | 1024      | -           | 79.8       | [66]        |

Table 19: ImageNet classification results on 224×224 Images w/o extra data. An epoch is calculated based on the number of full images processed during pre-training [81], and non integer multipliers indicate usage of multi cropping [6]. AUG means extra data augmentations (different from [35]), HED means more heads in the vision transformer compared to the base model, CLS indicates usage of more than just the last CLS token as the representation of the image, DAL indicates usage of DALLE’s encoder [62], and RRC indicates only random resized cropping and random horizontal flipping as data augmentations. † indicates a linear layer trained without data augmentation.
References

[1] Akbari, H., Yuan, L., Qian, R., Chuang, W.H., Chang, S.F., Cui, Y., Gong, B.: Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. Advances in Neural Information Processing Systems 34 (2021)

[2] Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)

[3] Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. Advances in neural information processing systems 32 (2019)

[4] Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020)

[5] Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: European Conference on Computer Vision (2014)

[6] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems 33, 9912–9924 (2020)

[7] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)

[8] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)

[9] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)

[10] Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., Wang, J.: Context autoencoder for self-supervised representation learning. arXiv preprint arXiv:2202.03026 (2022)

[11] Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)

[12] Chen, X., He, K.: Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566 (2020)

[13] Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9640–9649 (2021)

[14] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014)

[15] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)

[16] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Pre-training transformers as energy-based cloze models. arXiv preprint arXiv:2012.08561 (2020)

[17] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: AISTATS (2011)

[18] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

[19] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[20] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
[21] Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N.: Peco: Perceptual codebook for bert pre-training of vision transformers. arXiv preprint arXiv:2111.12710 (2021)

[22] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[23] Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE transactions on pattern analysis and machine intelligence 38(9), 1734–1747 (2015)

[24] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9588–9597 (2021)

[25] El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., Grave, E.: Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:2112.10740 (2021)

[26] El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., Jégou, H.: Xcit: Cross-covariance image transformers. ArXiv abs/2106.09681 (2021)

[27] Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88, 303–338 (2009)

[28] Fang, Y., Dong, L., Bao, H., Wang, X., Wei, F.: Corrupted image modeling for self-supervised visual pre-training. ArXiv abs/2202.03382 (2022)

[29] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. 2004 Conference on Computer Vision and Pattern Recognition Workshop pp. 178–178 (2004)

[30] Geirhos, R., Narayanappa, K., Mitzkus, B., Bethge, M., Wichmann, F., Brendel, W.: On the surprising similarities between supervised and self-supervised models. ArXiv abs/2010.08377 (2020)

[31] Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F.A., Brendel, W.: Partial success in closing the gap between human and machine vision. Advances in Neural Information Processing Systems 34 (2021)

[32] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)

[33] Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)

[34] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)

[35] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. ArXiv abs/2006.07733 (2020)

[36] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)

[37] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)

[38] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2021)

[39] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15262–15271 (2021)
[40] iNaturalist 2019 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2019 (2019)

[41] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)

[42] Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. Advances in neural information processing systems 33, 19545–19560 (2020)

[43] Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2733–2742 (2018)

[44] Jenni, S., Jin, H., Favaro, P.: Steering self-supervised feature learning beyond local pixel statistics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6408–6417 (2020)

[45] Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)

[46] Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)

[47] Landau, B., Smith, L.B., Jones, S.S.: The importance of shape in early lexical learning. Cognitive development 3(3), 299–321 (1988)

[48] Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021)

[49] Li, X., Zhou, Y., Zhang, Y., Zhang, A., Wang, W., Jiang, N., Wu, H., Wang, W.: Dense semantic contrast for self-supervised visual representation learning. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 1368–1376 (2021)

[50] Li, Z., Chen, Z., Yang, F., Li, W., Zhu, Y., Zhao, C., Deng, R., Wu, L., Zhao, R., Tang, M., et al.: Mst: Masked self-supervised transformer for visual representation. Advances in Neural Information Processing Systems 34 (2021)

[51] Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., Nadai, M.: Efficient training of visual transformers with small datasets. Advances in Neural Information Processing Systems 34 (2021)

[52] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)

[53] Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)

[54] Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing pp. 722–729 (2008)

[55] Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)

[56] O Pinheiro, P.O., Almahairi, A., Benmalek, R., Golemo, F., Courville, A.C.: Unsupervised learning of dense visual representations. Advances in Neural Information Processing Systems 33, 4489–4500 (2020)

[57] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

[58] Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)

[59] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)

[60] Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017)
[61] Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5706–5715 (2018)

[62] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021)

[63] Tamkin, A., Liu, V., Lu, R., Fein, D., Schultz, C., Goodman, N.: Dabs: A domain-agnostic benchmark for self-supervised learning. arXiv preprint arXiv:2011.12062 (2021)

[64] Tartaglini, A.R., Vong, W.K., Lake, B.M.: A developmentally-inspired examination of shape versus texture bias in machines. arXiv preprint arXiv:2202.08340 (2022)

[65] Tomasev, N., Bica, I., McWilliams, B., Buesing, L., Pascanu, R., Blundell, C., Mitrovic, J.: Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? arXiv preprint arXiv:2201.05119 (2022)

[66] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)

[67] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)

[68] Verma, M., Kumawat, S., Nakashima, Y., Raman, S.: Yoga-82: a new dataset for fine-grained classification of human poses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 1038–1039 (2020)

[69] Wang, F., Kong, T., Zhang, R., Liu, H., Li, H.: Self-supervised learning by estimating twin class distributions. arXiv preprint arXiv:2110.07402 (2021)

[70] Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems 32 (2019)

[71] Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3024–3033 (2021)

[72] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3733–3742 (2018)

[73] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 3733–3742 (2018)

[74] Xiao, K., Engstrom, L., Ilyas, A., Madry, A.: Noise or signal: The role of image backgrounds in object recognition. arXiv preprint arXiv:2006.09994 (2020)

[75] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018)

[76] Xiao, T., Reed, C.J., Wang, X., Keutzer, K., Darrell, T.: Region similarity representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10539–10548 (2021)

[77] Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., Hu, H.: Self-supervised learning with swin transformers. arXiv preprint arXiv:2105.04553 (2021)

[78] Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16684–16693 (2021)

[79] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision. pp. 649–666. Springer (2016)

[80] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021)