Representation Learning by Detecting Incorrect Location Embeddings

Sepehr Sameni, Simon Jenni, Paolo Favaro

1 Computer Vision Group, University of Bern, Switzerland
2 Adobe Research

sepehr.sameni@unibe.ch, jenni@adobe.com, paolo.favaro@unibe.ch

Abstract

In this paper, we introduce a novel self-supervised learning (SSL) loss for image representation learning. There is a growing belief that generalization in deep neural networks is linked to their ability to discriminate object shapes. Since object shape is related to the location of its parts, we propose to detect those that have been artificially misplaced. We represent object parts with image tokens and train a ViT to detect which token has been combined with an incorrect positional embedding. We then introduce sparsity in the inputs to make the model more robust to occlusions and to speed up the training. We call our method DILEMMA, which stands for Detection of Incorrect Location EMbeddings with MAsked inputs. We apply DILEMMA to MoCoV3, DINO and SimCLR and show an improvement in their performance of respectively 4.41%, 3.97%, and 0.5% under the same training time and with a linear probing transfer on ImageNet-1K. We also show full fine-tuning improvements of MAE combined with our method on ImageNet-100. We evaluate our method via fine-tuning on common SSL benchmarks. 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.

Introduction

In computer vision, deep learning models trained on small labeled datasets can benefit greatly from supervised pre-training on datasets such as ImageNet (Girshick et al. 2014). Even more surprisingly, (He et al. 2020) showed that it is possible to pre-train with unlabeled data (with MoCo) and outperform pre-training with supervised learning on several downstream tasks. This led to the rapid development of several Self-Supervised Learning (SSL) methods, such as (Caron et al. 2020; Chen et al. 2020a,b; He et al. 2020).

Representations obtained via SSL have the ability to generalize to downstream tasks such as object classification, detection, and segmentation (Deng et al. 2009; Everingham et al. 2009; Krizhevsky 2009). Recent work suggests that representations with a shape bias generalize better to these tasks than those with a texture bias (Geirhos et al. 2018; Tartaglini, Vong, and Lake 2022).

In particular, the expectation is that image representations may do better in the transfer learning to a shape-based task, such as pose classification of Yoga82 (see Fig. 1). Thus, we investigate whether adding a regularization loss that is sensitive to shape to a state-of-the-art SSL method, might lead to better representation learning.

We propose DILEMMA, which is short for Detection of Incorrect Location EMbeddings with MAsked inputs. In our experiments we integrated DILEMMA with SimCLR (Chen et al. 2020a), DINO (Caron et al. 2021) and MoCoV3 (Chen, Xie, and He 2021), and find a consistent improvement across the majority of downstream tasks.

With DILEMMA, the image representation is encouraged to differentiate shapes thanks to two main components: 1) A binary classification loss to detect the correct/incorrect positions of object parts, and 2) the use of randomized input sparsity, so that every subset of object parts contributes to the whole image representation. The first component is a concept already proposed in the context prediction (Doersch, Gupta, and Efros 2015) and jigsaw puzzle SSL methods (Noroozi and Favaro 2016). It takes also inspiration from ELECTRA (Clark et al. 2020a), where some text tokens are replaced by a weak generator and a discriminator is trained to detect them. The second component, is also a concept that has been exploited in VATT (Akbari et al. 2021) and MAE (He et al. 2021) to reduce the computational workload of training with ViTs (Dosovitskiy et al. 2020).

More in detail, as shown in Fig. 2, we split an image into a grid of tiles, map them to tokens, and combine them with positional embeddings. Then, we corrupt the positional embeddings of a fraction of the tokens before we feed them to a ViT. In our DILEMMA loss, we classify the tokens into those with correct and incorrect positional embeddings. The sparsification of the input can be implemented in a ViT sim-
Our contributions can be summarized as follows:

- We introduce DILEMMA, a novel SSL regularization loss that enhances the shape discriminability of image representations; it is based on the detection of misplaced positional embeddings with a ViT and the use of sparsity in the input;
- We propose to randomly sparsify the inputs and to use a student-teacher architecture to: 1) reduce the memory storage, 2) close the gap between training and test data, 3) speed up the training;
- DILEMMA boosts the performance of MoCoV3, SimCLR, DINO, and MAE under the same computational budget.

Related Work

Self-Supervised Learning for Image Representations.

Self-supervised learning gained popularity as a form of unsupervised learning, where pretext tasks leverage supervision signals obtained without human labor. Some classic examples are the classification of image patch locations (Doersch, Gupta, and Efros 2015; Noroozi and Favaro 2016), the reconstruction of color channels (Zhang, Isola, and Efros 2016) or image patches (Pathak et al. 2016), or the recognition of various image transformations (Gidaris, Singh, and Komodakis 2018; Jenni and Favaro 2018). 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 domain gaps between pre-training and transfer.

Contrastive Learning.

Efforts to scale up and improve instance discrimination (Dosovitskiy et al. 2015; Wu et al. 2018a) as a self-supervised pre-training task have established contrastive learning (Chen et al. 2020a; He et al. 2020; Oord, Li, and Vinyals 2018) 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 (He et al. 2020; Chen et al. 2020b), some remove the need for explicit negative pairs (Grill et al. 2020; Chen and He 2020), and others extend the set of positives beyond data-augmentation through clustering (Caron et al. 2020) or nearest-neighbors in feature space (Dwibedi et al. 2021). Another line of work considers contrastive pre-training strategies tailored to dense prediction tasks (O Pinheiro et al. 2020; Wang et al. 2021; Xiao et al. 2021; Xie et al. 2021b; Li et al. 2021b; Liu et al. 2021a). More recently, contrastive methods leverage vision transformer architectures (Dosovitskiy et al. 2020; Liu et al. 2021b), e.g., by adapting existing approaches (Chen, Xie, and He 2021; Xie et al. 2021a), tailoring architectures (Li et al. 2021a), or novel objectives (Caron et al. 2021). In our approach, we show that several well-established contrastive baselines (Chen, Xie, and He 2021; Caron et al. 2021; Chen et al. 2020a) can be improved 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 (Vaswani et al. 2017) 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 (Devlin et al. 2018) have recently also been adapted to the image domain (Bao, Dong, and Wei 2021; Zhou et al. 2021; He et al. 2021; Zhou et al. 2021). 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 (Clark et al. 2020a,b). 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. (Fang et al. 2022) use BEiT (Bao, Dong, and Wei 2021) for that purpose. The method of DABS (Tamkin et al. 2021) 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. MP3 (Zhai et al. 2022) also predicts the position of all the tokens like jigsaw (Noroozi and Favaro 2016) with a ViT. A technique that has proven very beneficial to improve the training efficiency of vision transformers is token dropping (Akbari et al. 2021; He et al. 2021; El-Nouby et al. 2021; Chen et al. 2022). 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.

Training DILEMMA

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 (Grill et al. 2020) to \( x \) and obtain \( \hat{x}_1 \) and \( \hat{x}_2 \). Similarly to ViT, each input \( \hat{x}_1 \) and \( \hat{x}_2 \) is divided in \( 14 \times 14 \) tiles, flattened and projected to \( N \) tokens \( t_{1,i}, t_{2,i} \in \mathbb{R}^D \), \( \forall i \in U \equiv \{1, \ldots, N\} \), through a linear projection. We then combine each token \( t_{i,j} \), with a positional embedding \( p_i \) in \( \mathbb{R}^D \), which can be either learned or fixed.

As in MoCoV3 (Chen, Xie, and He 2021), we define a Student \( S \) and a Teacher \( T \) ViTs (Dosovitskiy et al. 2020), 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 \( t_{1,1}, \ldots, t_{1,N} \) with the corresponding positional embeddings \( p_{1,1}, \ldots, p_{1,N} \). The Student instead receives as input a sparse set \( M \subset U \) of tokens \( t_{2,i}, i \in M \). For a randomized fraction of these tokens \( B \subset M \) the corresponding positional embeddings \( q_{i}, i \in M \) are
The Teacher network takes the complete set of tiles as input (dense) and without mismatches in the positional embeddings for each token. The Student takes only a subset of the tiles as input (sparse) and some tiles have incorrect positional embeddings. The Student is then trained under two losses: one is the contrastive loss of the class tokens (CLS) between the Teacher and the Student, and the other is the DILEMMA binary cross-entropy for each token.

Combining DILEMMA and Contrastive Learning

The DILEMMA loss can be integrated with other SSL losses. Here we describe the integration with the contrastive loss, but other choices follow an identical procedure.

The contrastive loss is defined as

\[
\mathcal{L}_{\text{CNT}} = \mathbb{E}_{x} \left[ L_{\text{CE}} \left( S_{0} \left( \{ q_{j} \odot t_{2,j} \}_{j \in M \cup \{0\}} \right), T_{0} \left( \{ p_{j} \odot t_{1,j} \}_{j=0,...,N} \right) \right) \right],
\]

where

\[
L_{\text{CE}} \left( A, V \right) = -2\tau \sum_{n} z_{n} \log \text{softmax} \left( \frac{A_{n} V_{n}}{\tau} \right)
\]

and \( A \) and \( V \) are \( G \times m \) matrices, with \( m \) the minibatch size and \( G \) the vector size after the projection \( Y \) (see eq. (1)), \( z_{j} \) is the one-hot vector with 1 at the \( j \)-th position and the index \( n \) indicates the class token within the minibatch.

When we combine both the DILEMMA and the contrastive losses into a single cost we obtain

\[
\mathcal{L}_{\text{UNION}} = \lambda_{\text{DILEMMA}} \mathcal{L}_{\text{DILEMMA}} + \mathcal{L}_{\text{CNT}},
\]

which we minimize and where \( \lambda_{\text{DILEMMA}} > 0 \) is a hyper parameter which we always set to 0.4.

Implementation

Architecture. We use Vision Transformers (ViT) (Dosovitskiy et al. 2020) with a patch size of \( 16 \times 16 \) pixels and an input image size of \( 224 \times 224 \) pixels, which gives a total of \( (224/16)^2 = 196 \) tokens. Due to computational limitations, we mostly use the small variant of the Vision Transformer.
Pre-training Setup. For our main model, we pre-train DILEMMA on ImageNet-1K (Deng et al. 2009) with the exact same hyper-parameters of MoCoV3 using three GeForce RTX 3090 GPUs for 100 epochs with a base batch size of 345. We set $\theta$ to 0.4 and the probability of positional embedding mismatch $\theta = 0.2$. We use sparsity ratios of 0%, 40%, 55%, 65% with $\times 1, \times 2, \times 3, \times 4$ base batch size and disable the DILEMMA loss when the input is dense.

To show the compatibility of the proposed method with other SSL methods, we also added two short runs for SimCLR and DINO with multi-cropping. For the DINO experiments, we used ViT-Base to show that DILEMMA scales to larger models. Since input sparsity allows for faster training, we also report results of DILEMMA variants with equal training time as the baselines.

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 and perform a coarse grid search over learning rates, batch sizes, and whether to normalize the data before feeding it to the linear layer or not (similarly to the added BatchNorm layer (Ioffe and Szegedy 2015) in MAE (He et al. 2021)). In contrast, DINO (Caron et al. 2021), obtains its representation by concatenating the CLS token of the last four attention layers of the network.

Experiments

We evaluate the use of DILEMMA on several datasets, compare it to state-of-the-art (SotA) SSL baselines, and perform ablations to show the role of each loss component. In each table, where we compare to an SSL baseline, we indicate the baseline with a method name (e.g., MoCoV3 (Chen, Xie, and He 2021)) and use a $+$ \{DILEMMA/sparsity\} to indicate that the baseline immediately above is combined with just sparsity or with the DILEMMA loss, which includes sparsity. We compare these two cases to show the added benefit of the DILEMMA positional classification loss over the lone sparsity.

Classification on ImageNet-1K

We show that DILEMMA leads to better representations for ImageNet-1K than prior SotA methods. Since this dataset has been used as a reference in SSL, it allows an easy comparison with previous work. In all tested cases, DILEMMA shows a consistent and significant improvement over the baseline it has been integrated with. Notice that the improvement due to the positional loss, relative to the use of sparsity, becomes more significant with a longer training.

$k$-NN and Linear Probing. In Table 1, we evaluate the quality of the ImageNet-1K pre-trained features. We either use a weighted $k$ nearest neighbor classifier (we always use $k = 20$) (Wu et al. 2018b) or a simple linear layer on top of a frozen backbone and frozen features. Since the use of sparsity has the added benefit of reducing the computational load at each iteration, we also show the actual training time. For example, with a ViT-Base/16 model and multi-crop, DINO $+$ DILEMMA (denoted with the $\dagger$ symbol) trains for 60 epochs in about the same time DINO trains for 45 epochs. This gives a significant advantage in performance. Furthermore, DILEMMA outperforms the baseline methods even if trained for the same number of epochs. The improvement under the same number of epochs is about 1 – 2% due to sparsity and 0.15 – 0.33% due to the positional classification loss for the $k$-NN evaluation. Similarly, it is about 2 – 3% due to sparsity and 0.06 – 0.46% due to the positional classification loss for our linear probing. Notice that the boost...
Table 2: Transfer learning for image classification on 13 datasets. The † variants are trained for the same duration as the corresponding (non-sparse) baselines. – Position is the case of MoCoV3, where the input tokens do not have the corresponding positional embeddings.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet-1%</th>
<th>ImageNet-10%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k-NN</td>
<td>Linear</td>
</tr>
<tr>
<td>MoCoV3</td>
<td>38.70</td>
<td>87.35</td>
</tr>
<tr>
<td>+Sparsity</td>
<td>43.29</td>
<td>89.25</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>44.43</td>
<td>89.55</td>
</tr>
<tr>
<td>+Sparsity†</td>
<td>44.64</td>
<td>89.89</td>
</tr>
<tr>
<td>+DILEMMA†</td>
<td>46.02</td>
<td>90.29</td>
</tr>
</tbody>
</table>

Table 3: Low-shot learning on ImageNet-1K. The † variants are trained for the same duration as the corresponding (non-sparse) baselines. In the Single-Crop case, DINO is shown only as a reference.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet-1%</th>
<th>ImageNet-10%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k-NN</td>
<td>Linear</td>
</tr>
<tr>
<td>MoCoV3</td>
<td>38.48</td>
<td>43.69</td>
</tr>
<tr>
<td>+Sparsity</td>
<td>40.02</td>
<td>45.44</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>41.64</td>
<td>47.95</td>
</tr>
<tr>
<td>+Sparsity†</td>
<td>42.42</td>
<td>48.34</td>
</tr>
<tr>
<td>+DILEMMA†</td>
<td>45.62</td>
<td>51.58</td>
</tr>
</tbody>
</table>

Downstream Tasks

We evaluate DILEMMA on several datasets to assess its generalization capability across different classification and detection tasks. While DILEMMA improves the performance over the baselines in all the datasets, the most significant improvement seems to occur for more shape-based tasks, such as pose classification. The evaluation on object segmentation, which is a dense downstream task, illustrates the representation captured by the non-CLS tokens.

Transfer Learning. In Table 2, we evaluate the transfer capability of our representations for image classification on several datasets. We use: Aircraft (Maji et al. 2013), Caltech101 (Fei-Fei, Fergus, and Perona 2004), Cars (Krause et al. 2013), CIFAR10 (Krizhevsky 2009), CIFAR100 (Krizhevsky 2009), DTD (Cimpoi et al. 2014), Flowers102 (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Van Gool 2014), INat19 (INaturalist 2019 competition dataset), Pets (Parkhi et al. 2012), STL10 (Coates, Ng, and Lee 2011), SVHN (Netzer et al. 2011), and Yoga82 (Verma et al. 2020). We train a linear layer on top of the frozen features to accelerate the process. DILEMMA performs well in transfer learning across all datasets and significantly more on datasets with shape-based tasks, such as Yoga82 (Verma et al. 2020) (for yoga position classification).
We also try to measure approximately how much shape matters in each dataset. We evaluate MoCoV3 with tokens without their position embedding. For simplicity, we use the same pre-trained MoCoV3 used throughout the experiments (although one should ideally use a MoCoV3 trained without position embeddings). We indicate this case with —Position in Table 2. Without position embedding, these features are equivalent to a bag of features. We can see that the improvement due to DILEMMA relative to the baseline MoCoV3 follows the corresponding relative degradation due to the bag of features representation. This suggests that DILEMMA tends to generalize better on datasets with shape-based tasks.

**Semantic Segmentation on ADE20K.** In Table 4, we show the evaluation of DILEMMA on semantic segmentation. This 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 evaluated by fine-tuning the model with an extra decoder. For that we use UPerNet (Xiao et al. 2018) on the ADE20K (Zhou et al. 2017) dataset and train the model for 160K iterations with a batch size of 2 for ViT-Base and 8 for ViT-Small. We also follow the evaluation protocol of iBOT (Zhou et al. 2021) and just train a linear layer (for 160K iterations and a batch size of 16) for semantic segmentation with a frozen backend to directly assess the per-token representation. The results show that DILEMMA is also better than the baseline models for dense classification tasks. It yields remarkable mIoU improvements of 4.6% against MoCoV3 and of 5.2% against DINO in the linear settings and under the same training time.

**Unsupervised Object Segmentation.** In Table 5, we evaluate the single frame object segmentation task. We use the mask generated from the attention of the CLS token (thresholded to keep 0.9 of the mass) as in DINO (Caron et al. 2021), and report the Jaccard similarity between the ground truth and the mask evaluated on the validation set of PASCAL-VOC12 (Everingham et al. 2009). For the videos, we use the DAVIS-2017 video instance segmentation benchmark (Pont-Tuset et al. 2017) and by following the protocol introduced in Space-time by Jabri et al. (Jabri, Owens, and Efros 2020) we segment scenes via the nearest neighbor propagation of the mask. In these evaluations, the role of the positional classification loss seems to be more important than sparsity alone.

**Humanoid Vision Engine Benchmark.** We also use the newly introduced HVE (Ge et al. 2022) to evaluate our shape bias in Table 6. In HVE Shape dataset, the input images are only the depth map of the foreground object which only contains shape information. We see that DILEMMA outperforms the base model which confirms our hypothesis that DILEMMA can focus on shape. For the HVE Texture, only four grey scaled random crops of the foreground object are concatenated and fed as input, so predicting the right class requires high texture discriminability. Results on HVE Texture show that DILEMMA’s better shape understanding did not harm the texture discriminability.

**Robustness against Background Change.** Following the background challenge evaluation metric (Xiao et al. 2020), 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 7, 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.B.T. (Only Background with foreground replaced with Tiled background)) the model performs correctly and does not just rely on the background for image classification.

**Ablations**

In these experiments, we want to validate empirically a number of choices: 1) we ask how much the trained model...
Table 7: Robustness of pre-trained models against background changes. The ↑ models are trained for a number of epochs, such that the total training time (see column Time) is the same as for the baseline methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Background Change</th>
<th>Clean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M.N.(↑)</td>
<td>M.R.(↑)</td>
</tr>
<tr>
<td>MoCoV3</td>
<td>64.52</td>
<td>65.68</td>
</tr>
<tr>
<td>+Sparsity</td>
<td>65.53</td>
<td>67.75</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>65.19</td>
<td>68.37</td>
</tr>
<tr>
<td>+Sparsity↑</td>
<td>66.25</td>
<td>69.60</td>
</tr>
<tr>
<td>+DILEMMA↑</td>
<td>68.86</td>
<td>71.16</td>
</tr>
<tr>
<td>DINO</td>
<td>65.56</td>
<td>68.94</td>
</tr>
<tr>
<td>+Sparsity</td>
<td>67.68</td>
<td>71.28</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>69.58</td>
<td>71.85</td>
</tr>
<tr>
<td>+Sparsity↑</td>
<td>69.38</td>
<td>73.75</td>
</tr>
</tbody>
</table>

Table 8: Token dropping policy. Results are evaluated on IN100.

<table>
<thead>
<tr>
<th>Sampling Method</th>
<th>k-NN</th>
<th>Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importance Based</td>
<td>71.88</td>
<td>76.76</td>
</tr>
<tr>
<td>Random</td>
<td>73.98</td>
<td>77.78</td>
</tr>
</tbody>
</table>

Table 9: Random Dropping Ratio. Results on the left are evaluated on IN100 and on the right on IN-1K.

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>k-NN</th>
<th>Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>0% (Dense)</td>
<td>76.16</td>
<td>77.50</td>
</tr>
<tr>
<td>75%</td>
<td>73.98</td>
<td>77.78</td>
</tr>
<tr>
<td>Random</td>
<td>74.46</td>
<td>78.82</td>
</tr>
</tbody>
</table>

is robust to occlusions (sparsity) and positional errors; 2) whether the selection of tokens should be random or guided; 3) whether the ratio of dropped tokens should remain constant in time or instead vary; 4) what the relevance of the positional classification loss is; 5) the impact of the number of positional errors used during training; 6) whether other design variations are more effective than DILEMMA.

Ablation studies are conducted either on ImageNet100 (IN100) or ImageNet-1K (IN-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 IN-1K experiments we train all models for 50 epochs with MoCoV3 unless stated otherwise.

**Token Dropping Policy.** In Table 8, we compare the case of dropping the tokens that are less important based on the attention of the teacher network (Li et al. 2021c) compared to randomly dropping the tokens. Results show that simple random dropping works well and there is no need to introduce extra complexity to the policy.

**Randomized Dropping Ratio.** In Table 9, we verify that a randomized dropping ratio is better than a constant one. We conducted two experiments: one on IN100 and one on IN-1K. The results show that a randomized dropping ratio performs better than a constant one. On the more difficult IN-1K dataset, just applying sparsity is worse than using the dense model. Only with a random dropping ratio can the sparse model outperform the dense model.

**Position Classification Loss.** In Table 10, we verify that

the position classification loss helps, by training a dense model with position mismatch detection. Surprisingly, even though the Mismatch Detection (MD) (i.e., the average classification accuracy of the token locations – see “MD Acc.” in Table 10) is easily solved (it achieves 100% in the dense case), the dense model can still improve the performance of the model on a downstream task. The performance improvement for a task like in Yoga$_{a2}$, which requires a better understanding of shape, is quite significant both with the dense and randomized sparsity inputs.

Mismatch Probability. The probability of a positional embedding mismatch $\theta$ is one of the hyper-parameters of DILEMMA. Early in our experiments, we found out that 20% is much better than 15% (which is used by Electra (Clark et al. 2020a)), probably due to the higher information redundancy in images compared to text. In Table 11, we show that $\theta = 30\%$ yields worse performance than the default $\theta = 20\%$.

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 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 12 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 Yoga$_{a2}$.

Timing. To show the efficiency of the proposed method, we ran SimCLR, MoCoV3, DINO with and without multi-crop on 4 GPUs and reported the epoch times in Table 13.

Combining with MAE. To show the general applicability of our proposed method to masked models, we misplaced some of the MAE (He et al. 2021) inputs and added DILEMMA loss to the encoder of MAE in addition to the reconstruction loss of the decoder. Both MAE and DILEMMA are trained for 200 epochs on IN-100 (using the exact same hyperparameters of the official repository) and results in Table 14 show that we can outperform MAE both in terms of linear probe and finetuning.

Longer Pretraining. We pretrain MoCoV3 and DILEMMA for 1000 epochs on IN-100 and evaluate their linear performance to see whether the benefits of DILEMMA still hold for longer pretrainings. Results in Table 15 show that indeed DILEMMA always performs better than the baseline even with longer pretraining.

Weaker Data Augmentations. One of the most important factors for the performance of contrastive learners is the data augmentation. In this short experiment (50 epochs of pretraining, and 70 epochs of linear training) we only used random resized cropping (like MAE (He et al. 2021)) on IN-1K for both MoCoV3 and DILEMMA. Linear probe accuracy of DILEMMA is 44.48% and for MoCoV3 it is 29.65% (Note that a 100 epoch pretrained ResNet-50 (He et al. 2016) with SimCLR (Chen et al. 2020a) gets 33.1% accuracy). This huge gap shows that DILEMMA is a generic method for representation learning and does not completely depend on the contrastive component of the loss.

Conclusions

We introduced a novel SRL method based on a position 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. Since 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

This work was supported by grant 200020_188690 of the Swiss National Science Foundation (SNSF) and an Adobe award.

Table 13: Timing and memory usage of training ViT-Small models with four RTX Geforce 3090 GPUs. MC stands for Multi-Crop. † models use ViT-Base

<table>
<thead>
<tr>
<th>Method</th>
<th>BatchSize</th>
<th>EpochTime</th>
<th>MaxMem(GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>640</td>
<td>21:35</td>
<td>23.65</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>1680</td>
<td>18:20 (×0.85)</td>
<td>23.17</td>
</tr>
<tr>
<td>MoCoV3</td>
<td>656</td>
<td>49:08</td>
<td>23.41</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>1664</td>
<td>32:11 (×0.65)</td>
<td>23.55</td>
</tr>
<tr>
<td>DINO</td>
<td>576</td>
<td>37:57</td>
<td>22.73</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>1184</td>
<td>24:13 (×0.64)</td>
<td>22.79</td>
</tr>
<tr>
<td>DINO(MC)†</td>
<td>144</td>
<td>3:07:21</td>
<td>22.85</td>
</tr>
<tr>
<td>+DILEMMA†</td>
<td>216</td>
<td>2:15:52 (×0.72)</td>
<td>23.69</td>
</tr>
</tbody>
</table>

Table 14: Combining DILEMMA with MAE. Results are evaluated on IN100 after pretraining for 200 epochs and using ViT-Base

<table>
<thead>
<tr>
<th>Method</th>
<th>Linear</th>
<th>Finetune</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>37.30</td>
<td>82.60</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>39.06</td>
<td>83.30</td>
</tr>
</tbody>
</table>

Table 15: Longer Pretraining on IN-100. Linear accuracies on IN100 after pretraining for 300 and 1000 epochs

<table>
<thead>
<tr>
<th>Method</th>
<th>300 Epochs</th>
<th>1000 Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCoV3</td>
<td>77.50</td>
<td>79.76</td>
</tr>
<tr>
<td>+DILEMMA</td>
<td>78.82</td>
<td>81.26</td>
</tr>
</tbody>
</table>
References


