




Rescuing Easy Samples in Self-Supervised Pretraining

Qin Wang¹^a, Kai Krajsek²^b and Hanno Scharr¹^c

¹IAS-8: Data Analytics and Machine Learning, Forschungszentrum Jülich, Germany

²Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, Germany
{qi.wang, k.krajsek, h.scharr}@fz-juelich.de

Keywords: Self-Supervised Learning, Augmentation, Vision Transformer.

Abstract: Many recent self-supervised pretraining methods use augmented versions of the same image as samples for their learning schemes. We observe that 'easy' samples, i.e. samples being too similar to each other after augmentation, have only limited value as learning signal. We therefore propose to rescue easy samples and make them harder. To do so, we select the top k easiest samples using cosine similarity, strongly augment them, forward-pass them through the model, calculate cosine similarity of the output as loss, and add it to the original loss in a weighted fashion. This method can be adopted to all contrastive or other augmented-pair based learning methods, whether they involve negative pairs or not, as it changes handling of easy positives, only. This simple but effective approach introduces greater variability into such self-supervised pretraining processes, significantly increasing the performance on various downstream tasks as observed in our experiments. We pretrain models of different sizes, i.e. ResNet-50, ViT-S, ViT-B, or ViT-L, using ImageNet with SimCLR, MoCo v3, or DINOv2 training schemes. Here, e.g., we consistently find to improve results for ImageNet top-1 accuracy with a linear classifier establishing new SOTA for this task.

1 INTRODUCTION


Self-supervised learning (SSL) from unlabeled data is the most common approach for foundation model pretraining (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; Chen et al., 2021; Oquab et al., 2023; Caron et al., 2021; He et al., 2022). Specifically, SSL is a technique that, in the ideal case, learns a task agnostic image feature representation on a *pretext task* with unlabeled data that subsequently can be used on other *downstream tasks*.


Some current SSL techniques like SimCLR (Chen et al., 2020a), MoCo (He et al., 2020; Chen et al., 2020b; Chen et al., 2021), and DINO (Oquab et al., 2023; Caron et al., 2021) make use of image pairs. For example, SimCLR (Chen et al., 2020a) allows to derive image representations from unlabeled data by contrasting the representations of augmented versions of the same image, denoted 'positive pairs' and of different images, denoted 'negative pairs'. A crucial step in this process is to carefully craft positive as well as negative pairs for meaningful comparisons. Specifically, positive pairs are formed by means of data aug-


mentation, i.e. a chain of image transformations, e.g. geometrical transformations like cropping as well as pixel wise transformations like color jitter, is applied to an image to generate different 'views' of the same semantic content. It has been shown that the type of augmentations plays a crucial role for the quality of the learned image feature representation (Chen et al., 2020b). If the hand-crafted augmentations are not sufficiently diverse, the positive pairs may become too alike, leading to the neural network training converging prematurely without learning valuable image representations (Cai et al., 2020).

Consequently, it is crucial for SSL methods to be robust against the effects of too-easy positive pairs. To this end, we introduce a novel loss term $L_{\text{top}k}$ acting on the top k easiest samples in a batch, only, by strongly augmenting them. This constitutes a selective regularization 'rescuing' these pairs for the pretraining. To do so, we adopt RandAugment (Cubuk et al., 2019b) to create strongly augmented views in addition to the standard augmentation techniques used in the baseline SSL methods.

In our experiments, we combine the SSL loss of the baseline technique with our regularization term $L_{\text{top}k}$. This proposed approach has yielded significant performance improvements, surpassing

^a <https://orcid.org/0009-0002-1505-2455>

^b <https://orcid.org/0000-0003-3417-161X>

^c <https://orcid.org/0000-0002-8555-6416>

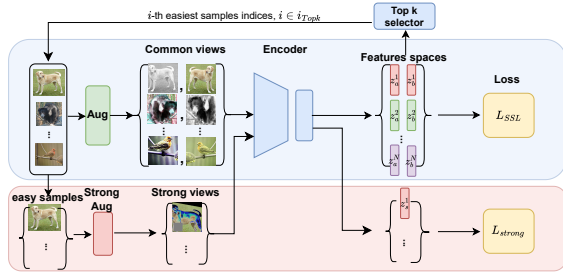


Figure 1: **Illustration of rescuing easy samples self-supervised pretraining framework.** Top k easiest samples are selected based on cosine similarity of features z_a and z_b from common views. The selected samples are strongly augmented, and input as additional strong views resulting in features z_s , used for computing L_{topk} (see (1)).

several established contrastive learning methods, not only on benchmarks like ImageNet (Russakovsky et al., 2015) but also across a range of downstream tasks, including classification and dense prediction tasks.

Our Contribution. We present

- a novel regularization term L_{topk} suitable for all augmented-views-based SSL methods,
- a sensitivity analysis of the involved hyperparameters and recommendations how to select them,
- experiments based on the most recent versions of SimCLR, MoCo, and DINO for a set of small to medium-sized network architectures,
- new SOTA for various downstream tasks.

2 RELATED WORK

2.1 Self-Supervised Learning

SSL encompasses a wide spectrum of methods, including information restoration (Larsson et al., 2016; Pathak et al., 2016; He et al., 2022), spatial context inference also denoted as pretext task learning (Doersch et al., 2015; Gidaris et al., 2018; Noroozi et al., 2018), canonical-correlation analysis methods (Andrew et al., 2013; Zbontar et al., 2021; Bardes et al., 2021), clustering methods, contrastive learning (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; Chen et al., 2021), self-distillation methods (Grill et al., 2020; Oquab et al., 2023; Caron et al., 2021), instance discrimination (Dosovitskiy et al., 2014; Wu et al., 2018) as well as generative approaches (Bengio et al., 2006; Springenberg, 2016; Donahue et al., 2017). Please note that this classification of SSL methods is not exclusive but might overlap. What all these methods

have in common is their ability to automatically derive an objective from unlabelled data.

The SSL methods applied in this work belong all to the group of multi-view invariance approaches that further can be classified into contrastive learning and self-distillation approaches. In order to define an objective for the learning process an image is transformed in two or more augmented views such that its semantic content is not changed. These different views are then mapped to the feature space by either the same or a different encoder for each view and an objective is formulated such that the feature vectors of the different views should be close in the feature space. The different methods differ in how they generate the different views, in the chosen model architecture, the way the encoder(s) are learned as well as the training objective.

One of the methods considered in this paper is the Momentum Contrast version 3 (MoCo v3 (Chen et al., 2021)) which has evolved from MoCo v2 (Chen et al., 2020b) and MoCo (He et al., 2020). The main idea of MoCo is the mapping of two different views by two separate encoders to the feature space and a contrastive loss is computed from the two views whereas the negative examples are obtained from a dynamic dictionary with a queue making the number of negative examples independent of the batch size. Only one encoder is updated using the contrastive loss whereas the parameters of the second encoder follow a moving average of the first one making it a self-distillation approach.

SimCLR (Chen et al., 2020a) proposed a similar idea but neglect the dynamic dictionary for negative examples but generated negative pairs from the current batch as well as abandon self-distillation approach but consider the same encoder for both views that are updated by means of gradient descent in each step. In addition, SimCLR considers an additional projection head before the contrastive loss and an extended data augmentation pipeline. MoCo v2 (Bardes et al., 2021) considered the extended data augmentation strategy as well as the projection head of SimCLR while MoCo v3 abandons the memory queue and replaces the convolution architecture by a Vision-Transformer (ViT).

The third SSL method under consideration here is DINOv2 (Oquab et al., 2023), which can be classified both as a self-distillation and an information restoration method. Its lineage can be traced back to the work of (Grill et al., 2020; Chen and He, 2021; He et al., 2022). The foundational idea from (Grill et al., 2020) involves feeding two different views into two decoders, a teacher and a student network and mapping the output of one encoder onto the output of the

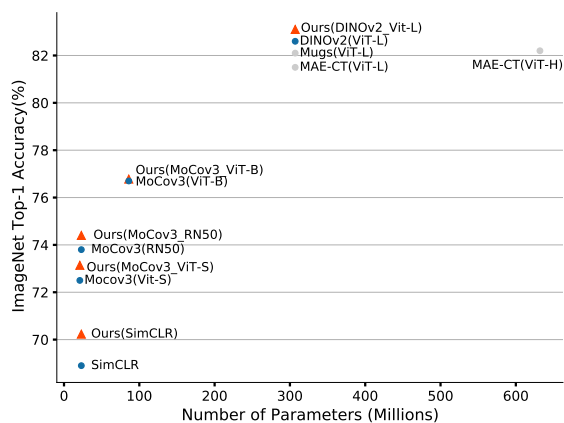


Figure 2: **Linear ImageNet top-1 accuracy.** Top-1 accuracy for linear classifier trained on frozen features from different SSL learning methods. Left: Model performance vs. number of parameters. We consistently improve baseline SSL methods (blue dots) with our adaptation (orange triangles). Grey dots are other SOTA SSL methods also not using additional training data. Right: values with [†] are taken from the original literature. Values with * are reproduced using their shared code. The experiments for DINOv2 are conducted three times with different seeds; mean and standard deviation are given.

second one, all without requiring negative examples. This process is safeguarded against collapsing due to the asymmetry of the encoders. DINOv2 has evolved from these approaches by incorporating a centering and softmax operation in the feature space, as detailed in (Caron et al., 2021). It also combines with masked image modeling (Zhou et al., 2022a) and introduces several techniques, such as regularizers, to stabilize training at large scale on curated imaging data.

2.2 Data Augmentations

Today’s de facto standard to generate positive pairs are data augmentation techniques, *i.e.* the original image is processed by a image processing pipeline where different geometric or pixelwise operations are applied to generate two or more views of the same semantic content. Consequently, applying objectives that foster corresponding features to be as close as possible to each other leads to image feature representations that are (nearly) invariant with respect to the transformations applied in the data augmentation pipeline. In supervised learning the type and strength of data augmentation has been vividly discussed as well as methods for automatically generating optimal data augmentation pipelines have been proposed (Cubuk et al., 2019b; Cubuk et al., 2019a). Mostly, in SSL the type and strength of the image transformations have not been focus of the publications until Chen *et al.* (Chen et al., 2020a) firstly examine the impact of different data augmentations in SimCLR, proposing a richer data augmentation pipeline as previous approaches. They have been later adopted also by others (Chen et al., 2020b; Zbontar et al., 2021; Bardes et al., 2021). Poole *et al.* (Poole et al., 2020)

systematically studied the influence of data augmentation both from a theoretical as well as from an empirical view and derived unsupervised and self-supervised approaches to synthesize optimal views following the InfoMin principle. In (Bordes et al., 2023) the influence of different data augmentations on different downstream tasks have been studied. In (Caron et al., 2020) more than two views have been explored leading to a more robust representation.

Combining strong and weaker augmentations so far got only little attention in the SSL literature. Wang *et al.* (Wang and Qi, 2021) combined strong and weaker augmentations and retrieved stronger augmentations from a comprehensive pool of instances by matching the distribution divergence between weakly and strongly augmented images. Unlike our approach, their method applies strong augmentation to all samples, potentially introducing a negative training effect of too-hard samples. We propose the top-*k* selection as a countermeasure.

Adaptive augmentation selection for all image pairs has been proposed by Zhang *et al.* (Zhang et al., 2023). They sample augmentations by probabilities that are derived from pretext task’s accuracies. In contrast to their approach, we apply strong augmentation where needed, only, *i.e.* we select pairs that lead to ‘too close’ features and therefore are to ‘too easy’, and re-adds them after applying stronger data augmentations.

To the best of our knowledge, our approach is the first one adapting augmentations for easy pairs, only, thus ‘rescuing’ them for training in a targeted fashion avoiding negative effects of too-easy samples, while also avoiding effects of too-hard samples.

Input : Number of steps S , batch size N ,
base encoder network $f(\cdot)$,
projection head $g(\cdot)$, loss scale
factor s , number k of images to
select for strong augmentation, SSL
loss function SSL .

Output: Trained encoder network $f(\cdot)$.

for $steps = 1$ to S **do**

```

    Sample a minibatch  $x = \{x_n\}_{n=1}^N$ ;
    Draw 2 sets of augmentations  $t_a \sim T$ ,
     $t_b \sim T$ ;
     $y_a \leftarrow t_a(x) \in \mathcal{R}^{N \times 3 \times X \times Y}$ ;
     $y_b \leftarrow t_b(x) \in \mathcal{R}^{N \times 3 \times X \times Y}$ ;
    Compute embeddings:
     $z_a \leftarrow g(f(y_a)) \in \mathcal{R}^{N \times D}$ ;
     $z_b \leftarrow g(f(y_b)) \in \mathcal{R}^{N \times D}$ ;
    Compute SSL loss:  $L_{SSL} \leftarrow SSL(z_a, z_b)$ ;
    Compute the cosine similarity  $CS$ ;
     $sim \leftarrow CS(z_a, z_b) \in \mathcal{R}^N$ ;
    Get the top  $k$  indices:
     $i_{topk} \leftarrow topk(sim, k).indices \in \mathbb{N}^k$ ;
     $z_s \leftarrow g(f(RandAugment(x_{i_{topk}}))) \in$ 
     $\mathcal{R}^{k \times D}$ ;
     $L_{topk} \leftarrow -mean(CS(z_a[i_{topk}], z_s))$ ;
    Total loss:  $L \leftarrow L_{SSL} + sL_{topk}$ ;
    Update networks  $f, g$  to minimize  $L$ ;

```

end

return encoder network $f(\cdot)$, and discard
 $g(\cdot)$.

Algorithm 1: Main Learning Algorithm.

3 METHOD

Pseudocode for our adaptive augmentation regularization for SSL pretraining is shown as Algorithm 1. Please note the symbols being defined there. A graphical illustration of our method is shown in Figure 1.

Current SSL algorithms involve the joint embedding of images distorted with common augmentations T , following the contrastive learning framework of SimCLR (Chen et al., 2020a). The self-supervised loss $SSL(z_a, z_b)$ is computed according to different methods, given two (or more) image features z_a and z_b , computed from two augmentations $t_a(x)$ and $t_b(x)$ of each image in minibatch $x \in \mathcal{R}^{N \times 3 \times X \times Y}$. We consider RGB color images of size $X \times Y$ and batch size N . Our innovation lies in a regularization loss term L_{topk} incorporating RandAugment (Cubuk et al., 2019b) to introduce an additional heavily augmented view to the input images of the k most similar positive pairs according to $sim = CS(z_a, z_b)$, where CS is the cosine similarity. This allows us to introduce $L_{topk} = -mean(CS(z_a[i_{topk}], z_s))$ weighted by a scale

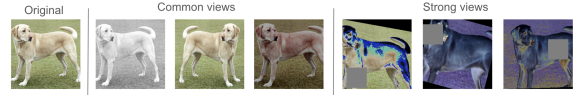


Figure 3: **Examples of common augmentation and strong augmentation.** The left image is a centercropped original image, the middle three images are common views generated by common self-supervised learning augmentations, and the right 3 images are distorted with RandAugment and Cutout.

factor s , using heavily augmented views of the k images with index i_{topk} according to sim . We elaborate this method in the following sections in more detail.

Strong Augmentations. The amalgamation of common augmentations for contrastive learning encompasses random resized crop, horizontal flip, color jitter, grayscale, Gaussian blur, solarization, and equalization. While some methods(He et al., 2020; Zbontar et al., 2021) employ asymmetric augmentation strategies to encourage the learning of more diverse features and prevent pretraining collapse, our approach with RandAugment introduces even more variant features. RandAugment includes common augmentations such as contrast, brightness, color jitter, equalization, sharpness, as well as spatial augmentations like translation, shear, rotation. Additionally, cutout is included in our repertoire of strong augmentations. We follow the augmentation strategy of RandAugment similar to (Sohn et al., 2020) to randomly select transformations for each input. Specifically, we include 14 types of augmentations with different severity for selecting randomly with pre-defined range as strong augmentation. See Figure 3 for examples of common views and strong views.

The strong augmentation pipeline is anticipated to generate more diverse images for self-supervised pretraining. To validate this assumption, we assess the cosine similarity of features extracted from a pre-trained neural network, specifically ResNet-50 trained on ImageNet, between common views and strongly augmented views. The mean cosine similarity score for common views over the entire dataset is 0.7069, whereas for strongly augmented views, it is 0.6690. Therefore, in this scenario, the strong augmentation pipeline is expected to yield more diverse images compared to common augmentations. Consequently, we can augment the dataset to include more challenging positive examples by employing the strong augmentation approach.

Top K Selector. Introducing strong augmentations to a neural network training increases data variability,

but this does not always yield positive effects (Wang and Qi, 2021). When the input samples already exhibit sufficient diversity, there may be no need to subject images to intense augmentation. In such cases, preserving the diverse samples becomes crucial as they play a vital role in enhancing the learning capabilities of self-supervised learning (SSL) methods, facilitating the acquisition of improved representations (Cai et al., 2020).

In our approach, we therefore refrain from uniformly applying strong augmentation to all images within a mini-batch. Instead, we selectively employ strong augmentation on the top k most similar views. The selection process is based on cosine similarity, where each image sample x^i in the mini-batch x is individually evaluated with the cosine similarity $z_a^i z_b^i / (|z_a^i| |z_b^i|)$ of its feature vectors $z_a^i \in \mathcal{R}^D$ and $z_b^i \in \mathcal{R}^D$ of its two commonly augmented views y_a^i and y_b^i . D is the output dimension of the projection head $g(\cdot)$. Cosine similarity is computed for the features within the common positive pair, only, not for negative pairs. Computing the cosine similarity score vector $sim \in \mathcal{R}^N$ therefore only adds linear complexity in batch size N .

The cosine similarity scores vector serves as the basis for selection. If the score for the cosine similarity of the same image ranks within the top k in the minibatch, we opt to heavily augment that particular image using RandAugment plus Cutout and input it into the neural network. The features corresponding to the top k strongly augmented views are subsequently used for computing the regularization loss L_{topk} .

Consistency Regularization. To encourage the invariance of strong augmented views and common augmented views, we introduce the regularization term L_{topk} to the loss function. The regularization term is defined as follows,

$$L_{\text{topk}} = -\text{mean}(CS(z_a, z_s)) \quad (1)$$

Here, CS is cosine similarity, z_a and z_s denote the features from common view and from strong augmented view respectively.

We incorporate cosine similarity CS as a regularization term in our approach. This choice aligns with SimCLR and MoCo, where cosine similarities are computed for subsequent use in contrastive loss calculations. Additionally, in the case of DINOv2, although cosine similarities are not explicitly computed, the features are readily available for cosine similarity computation. Our objective, centered on promoting invariances from multiple perspectives, leads us to adopt the negative cosine similarity as the

loss function. In our implementation, we adapt the SSL loss function by introducing the additional regularization term

$$L = L_{SSL} + sL_{\text{topk}} \quad (2)$$

Here, L_{SSL} denotes the SSL loss akin to SimCLR, MoCo and DINOv2, while L_{topk} represents the regularization term as shown in (1). The parameter s scales the impact of L_{topk} to balance between the stronger augmentation and the original SSL loss. A sensitivity analysis for hyperparameter s is shown in Section 5.

4 EXPERIMENTAL RESULTS

Following the protocols from the previous work (Chen et al., 2020a) (He et al., 2020) (Oquab et al., 2023), we conduct the downstream experiments on commonly used SSL evaluation datasets to evaluate the performance of pretrained neural networks. We show our results on ImageNet-1K (Deng et al., 2009), classification datasets, as well as dense prediction datasets.

Training Data. We pretrain the network with ImageNet-1K train partition without labels. Labels are only used for evaluation purposes in the downstream tasks.

Baseline SSL Methods. We incorporate our method into a series of current state-of-the-art (SOTA) self-supervised learning (SSL) techniques, including SimCLR, MoCo v3, and DINOv2. For SimCLR, we adhere to the ResNet-50 architecture configuration. In the case of MoCo v3, our methods are implemented on various architectures, namely ResNet-50, ViT-Small, and ViT-Base. In the context of DINOv2, we follow the configuration for the ViT-Large model, pretraining exclusively on ImageNet-1K. This diverse set of SSL methods and architectures allows us to comprehensively evaluate the effectiveness of our approach across different self-supervised learning frameworks.

In the results we name findings using our method according to the underlying baseline SSL method, e.g. 'Ours(SimCLR)' or 'Ours(MoCo v3)' etc. and specify the network architecture in addition, where needed.

Detailed Experiment Settings. We keep the same settings as in the baseline publications wherever possible. Specifically, we apply the following settings:

- **SimCLR** We use ResNet-50 as the architecture of the base encoder and optimize it using LARS (You et al., 2017) with learning rate 4.8 (i.e. $0.3 \times N/256$) and weight decay of 10^{-6} . We train at batch size $N = 4096$ for 1000 epochs. Furthermore, we use linear warmup for the first 10 epochs and decay the learning rate with cosine decay schedule. All sensitivity experiments use the same settings but train for 100 epochs, only.
- **MoCo v3** We use ResNet-50, ViT-S and ViT-B as architectures of the base encoder. For ResNet-50, we use the same optimiser settings as above for SimCLR. However, we follow the original literature to pretrain with 800 epochs. For ViT-S and ViT-B, we use AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0024 and weight decay 0.1. We train at batch size $N = 4096$ for 300 epochs. All sensitivity experiments use ViT-S settings for 300 epochs.
- **DINOv2** We use ViT-L as the architecture of the base encoder. The specific version of ViT-L is ViT-L/16. We train the base encoder with batch size $N = 2048$ for 100 epochs (i.e. 12500 iterations) with square rooted scale learning rate 0.004 and weight decay 0.04. All sensitivity experiments use above settings.

Hyperparameters. We use different hyperparameters for different baseline SSL methods according to the sensitivity analysis given in Section 5. For SimCLR we use $k = 128$ and $s = 0.75$, for MoCo v3 $k = 64$ and $s = 0.05$, and DINOv2 $k = 16$ and $s = 1$. All other hyperparameters are selected as given in the base SSLs publications, see paragraph above.

4.1 Linear Evaluations on ImageNet-1K

A common evaluation protocol for self-supervised learning model is linear probing (Balestriero et al., 2023). As usual, to evaluate the linear probing performance on ImageNet-1K, we froze the respective encoder f pretrained with our method, and finetune a linear classifier on top of the encoder with the ImageNet-1K training set. Then the results are evaluated with the validation set of ImageNet-1K. The main results are shown in Figure 2. We observe that our method based on MoCo v3 achieves the highest top-1 accuracy (74.4%) among the ResNet-50-based methods, outperforming SimCLR (69.1%) and Ours(SimCLR) (70.2%). Our methods based on DINOv2 achieves the best top-1 accuracy (83.1%) among all the SSL methods establishing new SOTA for this task. Our methods boost all the base SSL

algorithms we tested. It improved SimCLR by 1.1 %, MoCo v3(ResNet-50) by 0.6 % and DINOv2 by 0.5%. Besides, compared to DINOv2, our methods exhibits a lower standard deviation (± 0.02), indicating robustness in its performance across multiple experiments.

4.2 Transfer Learning on Downstream Tasks

Typical downstream tasks include augmentation invariant tasks (classification) and equivariant dense prediction tasks (detection and segmentation). We test our approach for both cases in the following.

Image Classification with Fixed Features. We follow the experimental settings as given for SimCLR (Chen et al., 2020a) and MoCo v3 (He et al., 2020). To this end, we train a linear classifier g for each task on frozen pretrained networks f . For evaluation, we use a range of classification tasks given by the datasets Food-101 (Bossard et al., 2014), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), SUN397 (Xiao et al., 2010), FGVC Aircraft (Maji et al., 2013), Describable Textures Dataset (DTD) (Cimpoi et al., 2014), Oxford-IIIT Pets (Parkhi et al., 2012), Oxford 102 Flowers (Nilsback and Zisserman, 2008) and PASCAL VOC 2007 (Everingham et al., 2010).

The results for SimCLR vs. Ours(SimCLR) are depicted in Figure 4a for ResNet-50, for MoCo v3 vs. Ours(MoCo v3) in Figure 4b for ViT-B, and Figure 4c shows the same plot for DINOv2 vs. Ours(DINOv2) for ViT-L. The respective performance values are shown in Table 1.

From the plots and table, we can infer that a network pretrained with adaptive strong augmentations outperforms baseline SSL methods across the majority of downstream classification datasets. However, the picture is not completely consistent across all base SSL methods. While our method exhibits comparable or improved performance to the SimCLR base method on 9 out of the 10 tasks, it does not surpass it on the VOC2007 dataset. Notably, our approach improves the baseline on datasets such as Cifar, Caltech, and Food101 by a significant margin (more than 1%). In contrast to SimCLR, results for VOC2007 are improved when using MoCo v3 or DINOv2, however, in both cases results for DTD, SUN397, and Food-101 do not improve.

Transfer Learning on Dense Prediction. We evaluate transfer learning performance across multiple dense prediction tasks. Specifically, we employ a linear classifier trained on a pre-trained network using

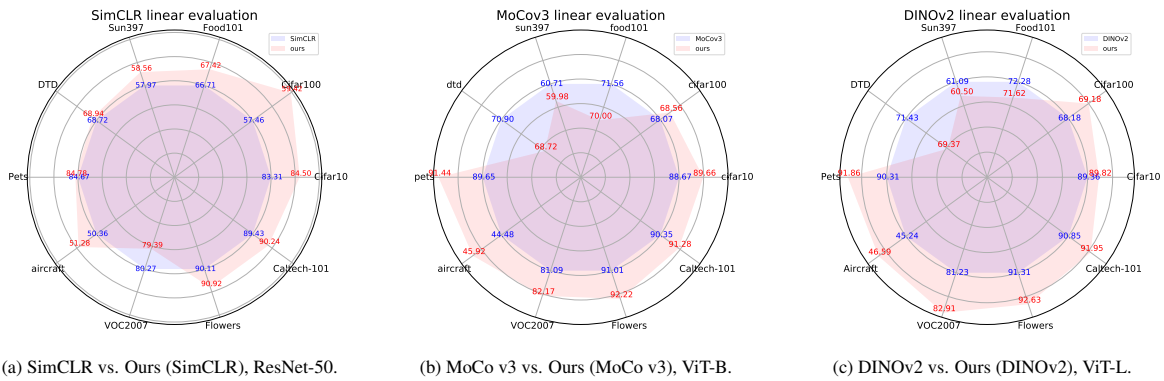


Figure 4: **Transfer learning on classification tasks.** Radii are equally spaced to indicate performance differences clearly. Numbers can also be found in Table 1.

Table 1: **Transfer learning on classification tasks.** Bold values indicate best results within direct comparison of Ours vs. SSL methods, underlined values overall best.

Methods	Cifar10	Cifar100	Food101	SUN397	DTD	Pets	Aircraft	VOC2007	Flowers	Caltech101
SimCLR	83.31	57.46	66.71	57.97	68.72	84.67	50.36	80.27	90.11	89.43
Ours(SimCLR)	84.50	59.42	67.42	58.56	68.94	84.78	51.28	79.39	90.92	90.24
MoCo v3	88.67	68.07	71.56	60.71	70.90	89.65	44.48	81.09	91.01	90.00
Ours(MoCo v3)	89.66	68.56	70.00	59.98	68.72	91.44	45.92	82.17	92.22	91.28
DINOv2	89.36	68.18	72.28	61.09	71.43	90.31	45.24	81.23	91.31	90.85
Ours(DINOv2)	89.82	69.18	71.62	60.50	69.37	91.86	46.59	82.91	92.63	91.95

both the PASCAL VOC dataset (Everingham et al., 2010) and the MS-CoCo dataset (Lin et al., 2015). Detailed results can be found in Table 2.

We observe for SimCLR and MoCo v3 that Ours consistently outperforms baseline methods across almost all evaluation metrics in tasks such as VOC07+12 detection, COCO detection, and COCO instance segmentation. In the domain of VOC07+12 detection, our adaptive augmentations self-supervised learning (SSL) method exhibits a marginal yet discernible enhancement in AP_{all} , AP_{50} , and AP_{75} compared to the baseline SSL methods. This suggests that Ours excels across various levels of precision. For DINOv2 half of the measures were improved. The underlined values in Table 2 reveal, that our method improves the best seen values in 6 out of 9 performance measures, where DINOv2 delivers the other 3 best or on par results.

In conclusion, our approach improves upon the baseline SSL methods in various dense prediction tasks, showcasing superior performance in precision, object detection, and instance segmentation tasks across different datasets. The observed enhancements are not only consistent but also statistically significant, underlining the effectiveness of our proposed method.

5 SENSITIVITY ANALYSIS

Our methods needs two hyperparameters to be tuned suitably, 'scale factor' s and number k of strongly augmented samples per mini-batch. We utilize the ImageNet-1K linear probing assessment (see Section 4.1) to report the efficacy of SSL methods. Given that loss values vary across different SSL methods, we conduct separate investigations per method.

Sensitivity to Scale Factor S . The scale factor s serves to balance the SSL loss and our novel regularization loss based on heavy augmentation (see (2)). Naturally $s \geq 0$ is a real positive number, where we expect a break down of training performance for too large s . We explore the impact of this hyperparameter on SSL pretraining, keeping the other hyperparameters fixed, specifically we set $k = 64$ for SimCLR and MoCo v3 and $k = 32$ for DINOv2. In Figure 5 we observe the expected decline for high s for all methods, however at quite different values of s . This is not unexpected, as the base SSL losses are different in their characteristics and amplitudes. For SimCLR we get $s = 0.75$ for ResNet-50, for MoCo v3 $s = 0.025$ for ViT-S and $s = 0.05$ for ResNet-50 and ViT-B (not shown), and for DINOv2 $s = 1$ (ViT-L) as best values. However, methods seem not to be too sensitive to the exact values of s , as their performance influ-

Table 2: **Transfer learning on dense prediction tasks.** Bold values indicate best results within direct comparison of Ours vs. baseline SSL methods, underlined values overall best. Hyperparameters have not been tuned for these tasks.

Method	VOC07+12 det			CoCo det			CoCo instance seg		
	AP_{all}	AP_{50}	AP_{75}	AP^{bb}	AP_{50}^{bb}	AP_{75}^{bb}	AP^{mk}	AP_{50}^{mk}	AP_{75}^{mk}
SimCLR	43.9	77.0	48.7	33.8	69.3	27.6	16.3	52.8	39.3
Ours(SimCLR)	46.4	77.9	43.9	34.7	69.9	29.5	16.4	53.5	38.9
MoCo v3	50.8	80.5	54.9	39.3	58.9	42.5	34.4	55.8	36.5
Ours(MoCo v3)	51.4	80.8	55.9	41.0	61.3	44.4	35.4	57.5	37.5
DINOv2	51.9	81.3	57.0	44.3	66.8	44.1	23.9	55.0	45.8
Ours(DINOv2)	52.2	81.1	57.3	43.9	66.9	43.9	24.2	54.5	45.8

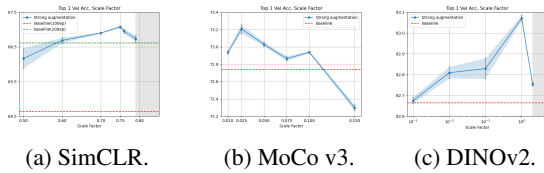


Figure 5: **Sensitivity to scale factor s .** Linear probing on ImageNet-1K compared to SSL baseline (red dashed line for 100 epochs and green dashed line for 200 epochs training, Our methods are trained for 100 epochs, only). The grey shaded areas denote pretraining collapse here. We evaluate the network three times, mean and standard deviation are reported in the graph.

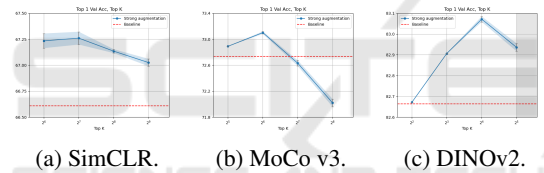


Figure 6: **Sensitivity analysis for hyperparameter k .** Linear probing on ImageNet-1K. Red dashed lines indicate baseline SSL performance. The grey shaded area indicates pretraining collapse here.

ence remains positive, i.e. above the baseline indicated as red dashed line, in a reasonably wide range around the maximum. This allows for relatively wide-spaced searches when trying to find the best s for a new method.

Sensitivity to Top K Parameter K . We investigate the sensitivity of our method to the hyperparameter k , controlling the number of samples to be strongly augmented. For the experiments we set s to the best values we found in the respective sensitivity analysis, above. From the plots in Figure 6 we see that the parameter k is best at $k = 64$ for SimCLR where training batch size is $N = 4096$, for MoCo v3 best at $k = 64$ with $N = 4096$, and for DINOv2 it is $k = 16$, where $N = 2048$. For DINOv2 also $k = 32$ performs almost as good as $k = 16$, indicating that a ratio of $N/k \approx 64$ may be a suitable rule of thumb.

6 CONCLUSIONS

Our novel strong-augmentation top k loss term is designed to be easily included in training methods that make use of augmented versions of the same sample. To apply it properly, we observed that one needs to tune two hyperparameters s and k , both being mildly sensitive. However, we found quite different optimal values for s for the investigated SSL methods SimCLR, MoCo v3, and DINOv2. Simply selecting some value that previously worked for an unrelated method may therefore be inappropriate. The parameter k seems to be better behaved and may be selected as $k = N/64$ as a rule of thumb, where N is the batch-size.

In this paper, we experiment with pretraining on ImageNet-1K, only, in order to keep needed compute (and CO₂-footprint) within reasonable limits. However, experiments show, that using our additional loss term improves performance of most downstream tasks, in some cases establishing new SOTA.

From Figure 2 we observe, that strongest improvements are achieved on smallest models and that improvements on larger models are statistically significant but sometimes small. This opens the question how well the found improvements transfer to larger models trained on larger datasets. This question cannot be answered by our current experiments and we plan to address this in future studies.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

REFERENCES

- Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA. PMLR.
- Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Milon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsivash, H., LeCun, Y., and Goldblum, M. (2023). A cookbook of self-supervised learning.
- Bardes, A., Ponce, J., and Vireg, Y. L. (2021). Vireg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2006). Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press.
- Bordes, F., Balestriero, R., and Vincent, P. (2023). Towards democratizing joint-embedding self-supervised learning.
- Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*.
- Cai, T. T., Frankle, J., Schwab, D. J., and Morcos, A. S. (2020). Are all negatives created equal in contrastive instance discrimination?
- Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924.
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations.
- Chen, X., Fan, H., Girshick, R., and He, K. (2020b). Improved baselines with momentum contrastive learning.
- Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE.
- Chen, X., Xie, S., and He, K. (2021). An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9620–9629, Los Alamitos, CA, USA. IEEE Computer Society.
- Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. (2014). Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019a). Autoaugment: Learning augmentation policies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*.
- Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2019b). Randaugment: Practical automated data augmentation with a reduced search space.
- Deng, J., Socher, R., Fei-Fei, L., Dong, W., Li, K., and Li, L.-J. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255.
- Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*.
- Donahue, J., Krähenbühl, P., and Darrell, T. (2017). Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.
- Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 15979–15988. IEEE Computer Society. Publisher Copyright: © 2022 IEEE.; 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022 ; Conference date: 19-06-2022 Through 24-06-2022.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning.
- Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

- Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *Computer Vision – ECCV 2016*, pages 577–593.
- Lehner, J., Alkin, B., Fürst, A., Rumetshofer, E., Miklautz, L., and Hochreiter, S. (2023). Contrastive tuning: A little help to make masked autoencoders forget.
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.
- Maji, S., Kannala, J., Rahtu, E., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft.
- Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*.
- Noroozi, M., Vinjimoor, A., Favaro, P., and Pirsiavash, H. (2018). Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision.
- Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. (2012). Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544.
- Poole, B., Sun, C., Schmid, C., Krishnan, D., Isola, P., and Tian, Y. (2020). What makes for good views for contrastive representation learning? In *NeurIPS*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence.
- Springenberg, J. T. (2016). Unsupervised and semi-supervised learning with categorical generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Wang, X. and Qi, G.-J. (2021). Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:5549–5560.
- Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3733–3742. Computer Vision Foundation / IEEE Computer Society.
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492.
- You, Y., Gitman, I., and Ginsburg, B. (2017). Large batch training of convolutional networks.
- Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research*, pages 12310–12320. PMLR.
- Zhang, Y., Zhu, H., and Yu, S. (2023). Adaptive data augmentation for contrastive learning.
- Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. (2022a). iBOT: Image BERT Pre-Training with Online Tokenizer. In *International Conference on Learning Representations (ICLR)*.
- Zhou, P., Zhou, Y., Si, C., Yu, W., Ng, T. K., and Yan, S. (2022b). Mugs: A multi-granular self-supervised learning framework.