# Masked Siamese Networks for Label-Efficient Learning

Mahmoud Assran(✉), Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas

Meta AI (FAIR), New York, USA
massran@fb.com

**Abstract.** We propose Masked Siamese Networks (MSN), a self-supervised learning framework for learning image representations. Our approach matches the representation of an image view containing randomly masked patches to the representation of the original unmasked image. This self-supervised pre-training strategy is particularly scalable when applied to Vision Transformers since only the unmasked patches are processed by the network. As a result, MSNs improve the scalability of joint-embedding architectures, while producing representations of a high semantic level that perform competitively on low-shot image classification. For instance, on ImageNet-1K, with only 5,000 annotated images, our base MSN model achieves 72.4% top-1 accuracy, and with 1% of ImageNet-1K labels, we achieve 75.7% top-1 accuracy, setting a new state-of-the-art for self-supervised learning on this benchmark. Our code is publicly available at https://github.com/facebookresearch/msn.

**Keywords:** Self-supervised representation learning · Low-shot classification · Vision transformers · Siamese networks

## 1 Introduction

Self-Supervised Learning (SSL) has emerged as an effective strategy for unsupervised learning of image representations, eliminating the need to manually annotate vast quantities of data. By training large models on unlabeled data, SSL aims to learn representations that can be effectively applied to a downstream prediction task with few labels [15].

One of the core ideas of SSL is to remove a portion of the input and learn to predict the removed content [43]. Auto-regressive models and denoising auto-encoders instantiate this principle in vision by predicting the missing parts at the pixel or token level [3,5,12,27,50]. Masked auto-encoders in particular, which learn representations by reconstructing randomly masked patches from

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19821-2_26.


S. Avidan et al. (Eds.): ECCV 2022, LNCS 13691, pp. 456–473, 2022.
https://doi.org/10.1007/978-3-031-19821-2_26

an input, have been successfully applied in vision [5,27,52,55]. However, optimizing a reconstruction loss requires modelling low-level image details that are not necessary for classification tasks involving semantic abstraction. Thus, the resulting representations often need to be fine-tuned for semantic recognition tasks which can lead to overfitting in low-shot settings. Nevertheless, masked auto-encoders have enabled the training of large-scale models and demonstrated state-of-the-art performance when fine-tuning on large labeled datasets, with millions of labels [3,5,27,55].

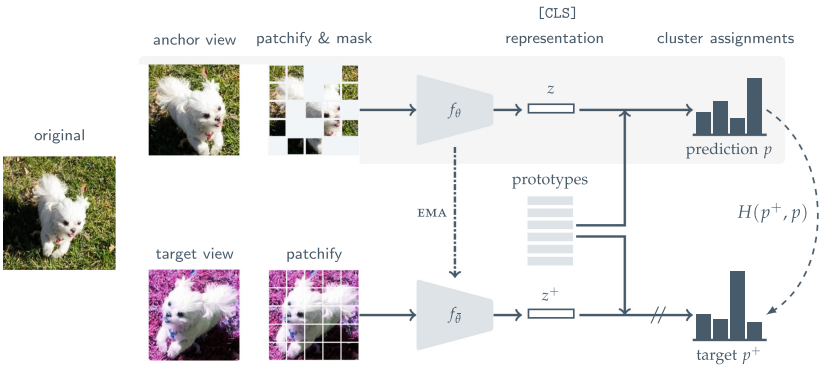



**Fig. 1. Masked Siamese Networks.** First use random data augmentations to generate two views of an image, referred to as the anchor view and the target view. Subsequently, a random mask is applied to the anchor view, while the target view is left unchanged. The objective is then to assign the representation of the masked anchor view to the same clusters as the representation of the unmasked target view. A standard cross-entropy loss is used as the criterion to optimize.

Joint-embedding architectures, on the other hand, avoid reconstruction. Approaches such as Siamese Networks [6,10,11,15,25,28,57] learn a representation by training an encoder network to produce similar embeddings for two different views of the same image [9,22]. Here the views are typically constructed by applying different image transforms—such as random scaling, cropping, and color jitter—to the input [41,53]. The inductive bias introduced by this invariance-based pre-training typically produces strong off-the-shelf representations of a high semantic level [11] but often disregards rich local structure that can be helpful to model.

In this work, we propose Masked Siamese Networks (MSNs), a self-supervised learning framework that leverages the idea of mask-denoising while avoiding pixel and token-level reconstruction. Given two views of an image, MSN randomly masks patches from one view while leaving the other view unchanged. The objective is to train a neural network encoder, parametrized with a vision transformer (ViT) [21], to output similar embeddings for the two views. In this procedure, MSN does not predict the masked patches at the input level, but

rather performs the denoising step implicitly at the representation level by ensuring that the representation of the masked input matches the representation of the unmasked one. Figure 1 shows a schematic of the method.

Empirically, we demonstrate that MSNs learn strong off-the-shelf representations that excel at low-shot prediction (cf. Fig. 2). In particular, MSN achieves good classification performance using 100× fewer labels than current mask-based auto-encoders [27,54]. In the standard 1% ImageNet low-shot classification task, an MSN-trained ViT-B/4 (using a patch size of 4x4 pixels) achieves 75.7% top-1 accuracy, outperforming the previous 800M parameter state-of-the-art convolutional network [14] while using nearly 10× fewer parameters (cf. Fig. 2a).

Since a good representation should not need many examples to learn about a concept [24], we also consider a more challenging evaluation benchmark for label-efficient low-shot classification [39,45], using from 1 labeled image per class up to 5 images per class (cf. Table 2). MSN also achieves state-of-the-art in that regime; e.g., with only 5 labeled images per class, we can pre-train a ViT-B with MSN on ImageNet-1K to achieve over 72% top-1 accuracy, surpassing the previous state-of-the-art method, DINO [11], by 8% top-1.

Similar to masked auto-encoders, MSNs also exhibit good computational scaling since only the unmasked patches are processed by the ViT encoder. For example, by randomly masking 70% of the patches, MSN uses half the computation and memory compared to an unmasked joint-embedding baseline. In practice, we pre-train a ViT-L/7 on as few as 18 AWS `p4d-24xlarge` machines. Without masking, the same job requires over 42 machines.

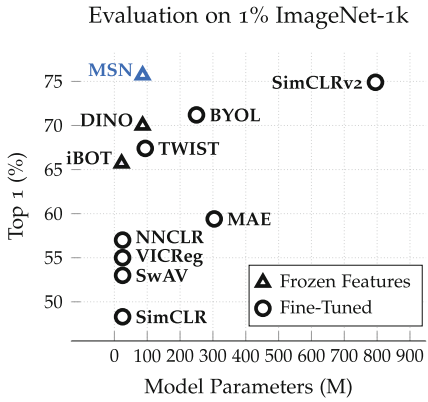
Finally, we also show that MSNs are competitive with prior works on other self-supervised benchmarks that use many labels for evaluation (e.g., fine-tuning, linear-evaluation, transfer learning).

## 2 Prerequisites

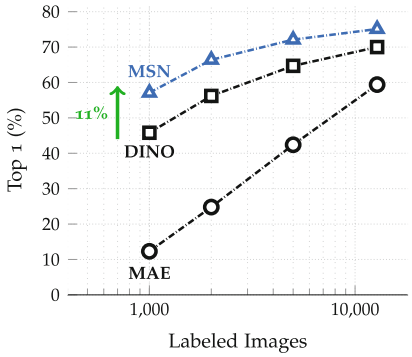
**Problem Formulation.** Consider a large collection of unlabeled images, $\mathcal{D} = (\mathbf{x}_i)_{i=1}^{U}$, and a small dataset of annotated images, $\mathcal{S} = (\mathbf{x}_{si}, y_i)_{i=1}^{L}$, with $L \ll U$. Here, the images in $\mathcal{S}$ may overlap with the images in the dataset $\mathcal{D}$. Our goal is to learn image representations by first pre-training on $\mathcal{D}$ and then adapting the representation to the supervised task using $\mathcal{S}$.

**Siamese Networks.** The goal of siamese networks [7,9], as they are used in self-supervised learning, is to learn an encoder that produces similar image embeddings for two views of an image. Specifically, given an encoder $f_\theta(\cdot)$ and two views $\mathbf{x_i}$ and $\mathbf{x_i^+}$ of an image, the encoder independently processes each view and outputs representations $z_i$ and $z_i^+$ respectively, referred to as the anchor representation and the target representation. The objective of siamese networks is to learn an encoder that is not sensitive to differences between views, so the representations $z_i$ and $z_i^+$ should match. In practice, the encoder $f_\theta(\cdot)$ is usually parameterized as a deep neural network with learnable parameters $\theta$.

The main challenge with siamese architectures is to prevent representation collapse in which the encoder produces a constant image embedding regardless of the input. Several approaches have been investigated in the literature. Contrastive losses explicitly push away embeddings of different images [9,15,28]. Information maximization approaches try to maximize the entropy of the average prediction [1,11] or spread out the embeddings uniformly on the surface of a sphere [10]. Asymmetric approaches rely on an asymmetric architectural choice such as stop-gradient operations and a momentum encoder [15,25] to prevent collapse. Other approaches try to decorrelate the vector components of the embeddings to minimize redundancy across samples [6,57].




(a) Evaluation using 1% of ImageNet-1K labels (∼13 imgs/class). Evaluation with *Frozen Features* corresponds to freezing the weights and training a logistic regression classifier with the available labeled samples. Evaluation with *Fine-Tuning* corresponds to adding a linear head and fine-tuning the model+head, end-to-end.




(b) Low-shot evaluation comparing MSN (ViT-L/7) to the best publicly available models in low-shot classification for DINO (ViT-B/8) and MAE (ViT-L/16). MSN and DINO use a linear probe, whereas MAE uses partial fine-tuning, where the last block of the pre-trained model along with a linear head are adapted.

**Fig. 2.** Low-shot Evaluation of self-supervised models, pre-trained on ImageNet-1K. (Left) MSN matches the previous 800M parameter state-of-the-art, while using a model that is 10× smaller, and no fine-tuning. (Right) MSN achieves good classification performance using less labels than current mask-based auto-encoders.

**Vision Transformer.** We use a standard Vision Transformer (ViT) architecture [21] as the encoder. Vision Transformers first extract a sequence of non-overlapping patches of resolution $N \times N$ from an image. Next, they apply a linear layer to extract patch tokens, and subsequently add learnable positional embeddings to them. An extra learnable `[CLS]` token is added to the sequence. This token aims to aggregate information from the full sequence of patches [11,21].

The sequence of tokens is then fed to a stack of Transformer layers [49]. A Transformer layer is composed of a self-attention [49] and a fully-connected layer with skip connections [29]. Self-attention uses an attention mechanism [4] applied to the entire sequence of elements to update the representation. The output representation associated to the `[CLS]` token is used as the output of the encoder.

(a) No Mask (b) Random Mask (c) Focal Mask

**Fig. 3. Masking strategies.** When applying a Random Mask, we randomly drop patches across a global view of the image. When applying a Focal Mask, we randomly select a local continuous block of an image, and mask everything around it. We typically leverage both Random and Focal Masking strategies when pre-training with MSNs.

## 3 Masked Siamese Networks

We now describe the proposed Masked Siamese Network (MSN) training procedure, which combines invariance-based pre-training with mask denoising; see Fig. 1 for a schematic. MSNs first use random data augmentations to generate two views of an image, referred to as the anchor view and the target view. Subsequently, a random mask is applied to the anchor view, while the target view is left unchanged. Similar to clustering-based SSL approaches [1,10,11], learning occurs by computing a soft-distribution over a set of prototypes for both the anchor and target views. The objective is then to assign the representation of the masked anchor view to the same prototypes as the representation of the unmasked target view. We use a standard cross-entropy loss to optimize this criterion.

In contrast to previous work on masked image modelling, the mask-denoising process in MSN is discriminative, rather than generative [5,27,52,55,61]. MSN architectures do not directly predict pixel values (or tokens) for the masked patches. Instead, the loss is applied directly to the output corresponding to the `[CLS]` token of the encoder.

**Input Views.** In each iteration of pre-training, we sample a mini-batch of $B \geq 1$ images. For an index $i \in [B]$, let $\mathbf{x}_i$ denote the $i^{\text{th}}$ image in the mini-batch. For each image $\mathbf{x}_i$, we first apply a random set of data augmentations to generate a target view, denoted $\mathbf{x}_i^+$, and $M \geq 1$ anchor views, denoted $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,M}$.

**Patchify and Mask.** Next, we "patchify" each view by converting it into a sequence of non-overlapping $N \times N$ patches. After patchifying the anchor view $\mathbf{x}_{i,m}$, we also apply the additional step of masking by randomly dropping some of the patches. We denote by $\hat{\mathbf{x}}_{i,m}$ the sequence of masked anchor patches, and by $\hat{\mathbf{x}}_i^+$ the sequence of unmasked target patches. Because of masking, the anchor sequence $\hat{\mathbf{x}}_{i,m}$ can have a different length than the patchified target sequence $\hat{\mathbf{x}}_i^+$, even if both image views originally have the same resolution.

We investigate two strategies for masking the anchor views, Random Masking and Focal Masking, which are depicted in Fig. 3. When applying Random Masking, we randomly drop potentially non-contiguous patches across the sequence. Conversely, when applying Focal Masking, we randomly select a local continuous block of the anchor view and drop all the patches around it.

**Encoder.** Given a parameterized anchor encoder, denoted $f_\theta(\cdot)$, let $z_{i,m} \in \mathbb{R}^d$ denote the representation computed from the patchified (and masked) anchor view $\hat{\mathbf{x}}_{i,m}$. Similarly, given a parameterized target encoder $f_{\bar{\theta}}(\cdot)$, with a potentially different set of parameters $\bar{\theta}$, let $z_i^+ \in \mathbb{R}^d$ denote the representation computed from the patchified target view $\hat{\mathbf{x}}_i^+$. In MSNs, the parameters $\bar{\theta}$ of the target encoder are updated via an exponential moving average of the anchor encoder parameters [25]. Both encoders correspond to the trunk of a ViT [21]. We take the output of the network to be the representation corresponding to the `[CLS]` token.

**Similarity Metric and Predictions.** Let $\mathbf{q} \in \mathbb{R}^{K \times d}$ denote $K > 1$ learnable prototypes, each of dimension $d$. To train the encoder, we compute a distribution based on the similarity between these prototypes and each anchor and target view pair, and we penalize the encoder for differences between these distributions. More precisely, for an anchor representation $z_{i,m}$, we compute a "prediction" $p_{i,m} \in \Delta_K$ in the $K$-dimensional simplex by measuring the cosine similarity to the prototypes matrix $\mathbf{q}$. For $L_2$-normalized representations and prototypes, the predictions $p_{i,m}$ can be concisely written as

$$p_{i,m} := \text{softmax}\left(\frac{z_{i,m} \cdot \mathbf{q}}{\tau}\right),$$

where $\tau \in (0, 1)$ is a temperature. Similarly, for each target representation $z_i^+$, we generate a prediction $p_i^+ \in \Delta_K$ by measuring the cosine similarity to the same prototypes matrix $\mathbf{q}$. When computing the target predictions, we also use a temperature parameter $\tau^+ \in (0, 1)$. Note, we always choose $\tau^+ < \tau$ to encourage sharper target predictions, which implicitly guides the model to produce confident low entropy anchor predictions. As we show in Appendix D, target sharpening coupled with mean-entropy maximization is provably sufficient to eliminate collapsing solutions in the MSN framework.

**Training Objective.** As previously mentioned, to train the encoder, we penalize when the anchor prediction $p_{i,m}$ is different from the target prediction $p_i^+$. We enforce this criterion using a standard cross-entropy loss $H(p_{i,m}, p_i^+)$.

We also incorporate the mean entropy maximization (ME-MAX) regularizer, also used in [1,33], to encourage the model to utilize the full set of prototypes. Denote the average prediction across all the anchor views by

$$\overline{p} := \frac{1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} p_{i,m}.$$

The ME-MAX regularizer simply seeks to maximize the entropy of $\overline{p}$, denoted $H(\overline{p})$, or equivalently, minimize the negative entropy of $\overline{p}$. Thus, the overall objective to be minimized when training the encoder parameters $\theta$ and prototypes $\mathbf{q}$ is

$$\frac{1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} H(p_{i,m}, p_i^+) - \lambda H(\overline{p}), \tag{1}$$

where $\lambda > 0$ controls the weight of the ME-MAX regularization. Note that when training, we only compute gradients with respect to the anchor predictions $p_{i,m}$, not the target predictions $p_i^+$.

## 4 Related Work

Unsupervised pre-training for vision has seen rapid progress with the development of view-invariant representation learning and joint embedding architectures [6,11,15,25,28,53]. Most similar to our approach is DINO [11] which leverages a Siamese Network with a cross-entropy loss and a momentum encoder. DINO also uses multi-crop training, which is a form of focal masking, but it requires an unmasked anchor view during training. MSN can be seen as a generalization of DINO, leveraging both random and focal masking without requiring any unmasked anchor views. Since the cross-entropy loss in Eq. (1) is only differentiated with respect to the anchor predictions, not the target, MSN only backpropagates through the anchor network and only needs to store the activation associated with the masked view. MSN therefore reduces the computational and memory requirements. MSN also differs from DINO in its mechanism for preventing representation collapse (entropy maximization as opposed to centering and sharpening). Our empirical results show that MSN compares favourably to DINO across various degrees of supervision for the downstream task.

A prominent line of work in SSL is to remove a portion of the input and learn to reconstruct the removed content [18]. For example, in the field of image recognition, some works have proposed to predict augmented image channels [60], which can be regarded as a form of image colorization [34,35,59]. Other approaches propose to remove and learn to regress entire image regions: the seminal Context Encoders of Pathak et al. [43] train a network to generate missing image patches based on their surroundings. Recent works revisit this idea and

investigate the pre-training of ViTs with masked auto-encoders [5,12,27,52,55]. These approaches corrupt images with mask-noise and predict missing input values at the pixel level [21,27,54] or using a tokenizer [5,52]. Our approach does not predict the missing value at the input level, but instead performs the denoising step implicitly by ensuring that the global representation of the noisy input matches that of the uncorrupted input.

Some recent approaches have started to explore the combination of joint-embedding architectures and denoising pre-training tasks [3,23,61]. Those approaches mask an image by replacing the masked patches with a learnable mask token, and output a single vector for each masked patch. The objective is then to directly match each computed patch vector to the equivalent patch token extracted from a target encoder. Different from these approaches, we only match the view representations globally and do not consider a patch level loss. Consequently, we can completely ignore the masked patches, significantly reducing the computational and memory requirements. For example, when training our largest model, a ViT-L/7, we mask over 70% of the input patches, and reduce memory and computational overhead by half.

**Table 1. Extreme low-shot.** We evaluate the label-efficiency of self-supervised models pretrained on the ImageNet-1K dataset. For evaluation, we use an extremely small number of the ImageNet-1K labels and report the mean top-1 accuracy and standard deviation across 3 random splits of the data.

| | | | Images per Class | | |
|---|---|---|---|---|---|
| **Method** | **Architecture** | **Epochs** | 1 | 2 | 5 |
| iBOT [61] | ViT-S/16 | 800 | 40.4 ± 0.5 | 50.8 ± 0.8 | 59.9 ± 0.2 |
| | ViT-B/16 | 400 | 46.1 ± 0.3 | 56.2 ± 0.7 | 64.7 ± 0.3 |
| DINO [11] | ViT-S/16 | 800 | 38.9 ± 0.4 | 48.9 ± 0.3 | 58.5 ± 0.1 |
| | ViT-B/16 | 400 | 41.8 ± 0.3 | 51.9 ± 0.6 | 61.4 ± 0.2 |
| | ViT-S/8 | 800 | 45.5 ± 0.4 | 56.0 ± 0.7 | 64.7 ± 0.4 |
| | ViT-B/8 | 300 | 45.8 ± 0.5 | 55.9 ± 0.6 | 64.6 ± 0.2 |
| MAE [27] | ViT-B/16 | 1600 | 8.2 ± 0.3 | 25.0 ± 0.3 | 40.5 ± 0.2 |
| | ViT-L/16 | 1600 | 12.3 ± 0.2 | 19.3 ± 1.8 | 42.3 ± 0.3 |
| | ViT-H/14 | 1600 | 11.6 ± 0.4 | 18.6 ± 0.2 | 32.8 ± 0.2 |
| MSN (Ours) | ViT-S/16 | 800 | 47.1 ± 0.1 | 55.8 ± 0.6 | 62.8 ± 0.3 |
| | ViT-B/16 | 600 | 49.8 ± 0.2 | 58.9 ± 0.4 | 65.5 ± 0.3 |
| | ViT-B/8 | 600 | 55.1 ± 0.1 | 64.9 ± 0.7 | 71.6 ± 0.3 |
| | ViT-L/7 | 200 | **57.1 ± 0.6** | **66.4 ± 0.6** | **72.1 ± 0.2** |

## 5 Results

We evaluate MSN representations learned on the ImageNet-1K dataset [44]. We first consider low-shot evaluation on ImageNet-1K using as few as 1–5 images per class. We also compare with the state-of-the-art in settings where more

supervision is available and investigate transfer-learning performance. Finally, we conduct ablation experiments with MSN. By default, we pre-train with a batch-size of 1024 images, generating several anchor views from each image: 1 view with a random mask, and 10 views with focal masks. We find that the optimal masking ratio is model-dependent, with larger models benefiting from more aggressive patch dropping. We describe MSN implementation details in Appendix C.

### 5.1 Label-Efficient Learning

The premise of SSL is to learn representations on unlabeled data that can be effectively applied to prediction tasks with few labels [14]. In this section we explore the performance of self-supervised approaches when very few labeled examples are available.

**Table 2.** Low-shot evaluation on ImageNet-1K using 1% of the labels (approximately 13 images per class). †Indicates evaluations we computed using publicly available models.

| Method | Architecture | Params. | Top 1 |
|---|---|---|---|
| **Comparing similar architectures** | | | |
| Barlow-Tw. [57] | RN50 | 24M | 55.0 |
| SimCLRv2 [14] | RN50 | 24M | 57.9 |
| PAWS [1] | RN50 | 24M | 66.5 |
| DINO [11] | ViT-S/16 | 22M | 64.5 |
| iBOT [61] | ViT-S/16 | 22M | 65.9 |
| MSN | ViT-S/16 | 22M | **67.2** |
| **Comparing larger architectures** | | | |
| BYOL [25] | RN200 (2×) | 250M | 71.2 |
| SimCLRv2 [14] | RN151+SK (3×) | 795M | 74.9 |
| iBOT [61]† | ViT-B/16 | 86M | 69.7 |
| DINO [11]† | ViT-B/8 | 86M | 70.0 |
| MSN | ViT-B/4 | 86M | **75.7** |

**Extreme Low-Shot.** We first evaluate the classification performance of unsupervised models that have been pre-trained on ImageNet-1K, by using 1, 2, and 5 labeled images per class for supervised evaluation. We compare MSN to the joint-embedding approach, DINO [14], the auto-encoding approach, MAE [27],

and the hybrid approach, iBOT [61], which combines a joint-embedding architecture with a token-based patch-level loss. We download the official released models of each related approach for evaluation.

To adapt the joint-embeddings models to the supervised task, we freeze the weights of the pre-trained model and train a linear classifier on top using 1, 2 or 5 labeled samples (see Appendix C). For MAE, we rely on partial fine-tuning [27], except for the 1 image per class setting, and all results with the ViT-H/14 architecture, which use a linear classifier. Partial fine-tuning corresponds to fine-tuning the last block of the pre-trained model along with a linear head. MAE benefits from partial fine-tuning, but for sufficiently large models, such as the ViT-H/14, this leads to significant overfitting in the low-shot regime. We compare both protocols in more detail in Appendix E.

Table 1 reports the extreme low-shot evaluation results. MSN outperforms the other representation learning approaches across all levels of supervision. Moreover, the improvement offered by MSN increases as the amount of available labeled data is decreased. The performance of MSN also benefits from increased model size—settings with less labeled data appear to benefit more from increased model depth and smaller patch sizes.

**Table 3.** Linear evaluation on ImageNet-1K using 100% of the labels.

| Method | Architecture | Params. | Epochs | Top 1 |
|---|---|---|---|---|
| **Comparing similar architectures** | | | | |
| SimCLRv2 [14] | RN50 | 24M | 800 | 71.7 |
| BYOL [25] | RN50 | 24M | 1000 | 74.4 |
| DINO [11] | ViT-S/16 | 22M | 800 | 77.0 |
| iBOT [61] | ViT-S/16 | 22M | 800 | **77.9** |
| MSN | ViT-S/16 | 22M | 600 | 76.9 |
| **Comparing larger architectures** | | | | |
| MAE [27] | ViT-H/14 | 632M | 1600 | 76.6 |
| BYOL [25] | RN200 (2×) | 250M | 800 | 79.6 |
| SimCLRv2 [14] | RN151+SK (3×) | 795M | 800 | 79.8 |
| iBOT [61] | ViT-B/16 | 86M | 400 | 79.4 |
| DINO [11] | ViT-B/8 | 86M | 300 | 80.1 |
| MoCov3 [16] | ViT-BN-L/7 | 304M | 300 | **81.0** |
| MSN | ViT-L/7 | 304M | 200 | **80.7** |

We also observe that joint-embedding approaches appear to be more robust to the limited availability of downstream supervision than reconstruction-based auto-encoding approaches. To explain this observation, we refer to the Masked

Auto-Encoders paper [27] which conjectures that using a pixel reconstruction loss results in encoder representations of a lower semantic level than other methods. Conversely, the inductive bias introduced by invariance-based pre-training appears to be helpful in the low-shot regime.

**1% ImageNet-1K.** Table 2 reports a comparison on the 1% ImageNet-1K task, which is a standard benchmark for low-shot evaluation of self-supervised models [13]. For reference, the best reported result in the literature on 1% labeled data is 76.6%, achieved with a multi-stage semi-supervised pipeline, i.e., self-distilling from a fine-tuned ResNet-152 with 3× wider channels and selective kernels [14]. Here we focus on comparing to other models trained in a self-supervised setting. Our best MSN model using a ViT-L/7 achieves 75.1% top 1 accuracy, surpassing the previous 800M parameter state-of-the-art convolutional network [14] while using significantly fewer parameters and no fine-tuning. When focusing the comparison on similar architectures (models with similar FLOP counts), MSN also consistently improves upon previous approaches.

### 5.2 Linear Evaluation and Fine-Tuning

In this section we compare with the state-of-the-art on standard evaluation benchmarks where more supervised samples are available to adapt the representation. We use the full ImageNet-1K training images with 1.28M labels.

**Table 4.** End-to-end fine-tuning of a ViT-B/16 encoder on ImageNet-1K using 100% of the labels. MSN obtains competitive performance with both joint-embedding approaches and auto-encoding approaches.

| Initialization | Pretrain Epochs | Top 1 |
|---|---|---|
| DINO [11] | 800 | 83.6 |
| BEiT [5] | 800 | 83.2 |
| iBOT [27] | 800 | 83.8 |
| MAE [27] | 1600 | 83.6 |
| SimMIM [55] | - | 83.8 |
| MaskFeat [52] | - | 84.0 |
| Data2Vec [3] | 800 | **84.2** |
| MSN | 600 | 83.4 |

**Linear Evaluation.** We evaluate self-supervised pretrained models by freezing their weights and training a linear classifier. Table 3 reports the linear evaluation results on ImageNet-1K. We observe that MSN performs competitively with the state-of-the-art. The best MSN model achieves 80.7% top-1 accuracy.

**Fine-Tuning.** In this evaluation setting, we finetune all the weights of the self-supervised model using all the labels from the ImageNet-1K training set. We focus on the ViT-B/16 architecture. We adopt the same fine-tuning protocol as [5], and provide the details in Appendix C. Table 4 reports the comparison with fine-tuning evaluation using 100% labels on ImageNet-1K. MSN is competitive with joint-embedding approaches, such as DINO, and generative auto-encoding approaches, such as MAE.

### 5.3 Transfer Learning

We also report transfer learning experiments on the CIFAR10, CIFAR100 and iNaturalist datasets in Table 5 when using a self-supervised ViT-B/16 pre-trained on ImageNet-1K. Across all tasks, various levels of supervision, and evaluation methods, MSN either outperforms or achieves similar results to DINO pre-training. Recall that MSN pre-training is also less computationally expensive than DINO pre-training due to the anchor masking.

**Table 5.** Transfer Learning with a ViT-Base/16 pre-trained on ImageNet-1K. Across all tasks, various levels of supervision, and evaluation methods, MSN either outperforms or achieves similar results to DINO pre-training. The MSN model is trained with a masking ratio of 0.3; i.e., dropping 30% of patches, and thus reduces the computational cost of pre-training relative to DINO.

| **Evaluation** | **Method** | **Top 1** | | | | |
|---|---|---|---|---|---|---|
| | | CIFAR10 | | CIFAR100 | iNat18 | iNat19 |
| | | 4000 labels | 50000 labels | | | |
| Fine-Tuning | DINO | – | 99.0 | 90.5 | 72.0 | 78.2 |
| | MSN | – | 99.0 | 90.5 | 72.1 | 78.1 |
| Linear Eval. | DINO | 93.2 | 95.3 | 82.9 | – | – |
| | MSN | 93.8 | 95.7 | 82.8 | – | – |

### 5.4 Ablations

We now conduct a series of experiments to gain insights into the important design decisions used in MSN such as the masking strategy and the data augmentation strategy. We measure the accuracy of the models by training a logistic regression classifier on the frozen trunk using 1% of ImageNet-1K labels (∼13 imgs/class).

**Combining Random and Focal Masking.** In MSN we apply both random and focal masking to the anchor views. Focal masking corresponds to selecting a small crop from the anchor view. Random masking corresponds to randomly dropping potentially non-contiguous patches from the anchor view.

**Table 6. Masking strategy.** Impact of masking strategy on low-shot accuracy (1% of ImageNet-1K labels) of a ViT-B/16. We only generate one anchor view of each image, except in the last row, where we generate two views, one with a Random Mask and one with a Focal Mask. A random masking ratio of 0.5 is used. Applying a random mask to the anchor view is better than applying no mask. By combining both random and focal masking strategies, we obtain the strongest performance.

| Anchor View | Top 1 |
|---|---|
| No Mask | 49.3 |
| Focal Mask | 39.3 |
| Random Mask | 52.3 |
| Random Mask + Focal Mask | **59.8** |

Table 6 reports the effect on low-shot evaluation when using a) No Masking, b) Focal Masking, c) Random Masking, or d) Random and Focal Masking. Applying a random mask to the anchor view is always better than applying no mask. By contrast, applying only a focal mask degrades the performance, which highlights the importance of maintaining a global view during pre-training. By combining both random and focal masking strategies, we obtain the strongest performance.

**Random Masking Ratio.** Here we explore the relationship between the optimal masking ratio and the model size. Table 7 reports the low-shot learning performance for various random masking ratios as we increase the model size.[1]

**Table 7. Masking ratio.** Impact of pre-training random masking ratio (fraction of randomly dropped patches in each random mask) on ImageNet 1% accuracy. Accuracy of larger models improves when leveraging aggressive masking during pre-training.

| | Top 1 | | | |
|---|---|---|---|---|
| | Random Masking Ratio | | | |
| **Architecture** | 0.15 | 0.3 | 0.5 | 0.7 |
| ViT-S/16 | **66.3** | 66.0 | 64.8 | – |
| ViT-B/16 | 68.8 | **69.6** | – | – |
| ViT-L/16 | NaN | NaN | **70.1** | 69.4 |

When increasing the model size, we find that increasing the masking ratio (dropping more patches) is helpful for improving low-shot performance. We also

[1] Note that the performance of the ViT-S/16 can be improved by removing the Sinkhorn normalization, as we do in Table 2, however for consistency of evaluation with other models, we keep it in for this ablation.

find that the ViT-L/16 runs with weak masking are unstable, while the runs with more aggressive masking are quite stable. However, we do not have sufficient evidence to claim that increasing the masking ratio always improves the stability of large ViT pre-training.

**Augmentation Invariance and Low-Shot Learning.** We explore the importance of data-augmentation invariance for low-shot learning. We pretrain a ViT-B/16 with MSN, where the teacher and anchor networks either share the input image view or use different input views; in both cases, the anchor view is always masked. The views are constructed by applying random ColorJitter, Crop, Horizontal Flips, and GaussianBlur to the input image.

Table 8 reports top-1 accuracy when evaluating with 1% of ImageNet-1K labels. Sharing the view leads to a top-1 accuracy of 7%; MSN finds a shortcut solution relying on color statistics. Using different colors in the input views resolves this pathological behaviour and achieves a top-1 of 48.3%. Further applying the geometric data-augmentations independently to the two views (as opposed to sharing views) further improves the performance to 52.3%, showing the importance of learning view-invariant representations in the low-shot setting.

**Random Masking Compute and Memory.** We look at the effect of the random masking ratio, i.e., the fraction of dropped patches from the global anchor view, on the computational requirements of large model pre-training. In each iteration we also generate 10 focal views (small crops) of each input image; the random masking ratio has no impact on these views.

**Table 8.** Impact of view-sharing during pre-training when evaluating on ImageNet 1%. The target view is constructed by applying random ColorJitter, Crop, Horizontal Flips, and GaussianBlur to the input image. When using the same image view, MSN finds a shortcut solution. Using color jitter prevents this pathological behaviour. Randomly applying additional geometric data transformations to the anchor further improves performance, demonstrating the importance of view invariance in the low-shot setting.

| **Anchor View Generation** | **Top 1** |
|---|---|
| Target View | 7.0 |
| Target View + ColorJitter | 48.7 |
| Target View + ColorJitter + Crop + Flip + GaussianBlur | **52.3** |

**Table 9.** Impact of random masking ratio on GPU memory usage and runtime when pre-training a ViT-L/7. Measurements are conducted on a single AWS `p4d-24xlarge` machine, containing 8 A100 GPUs, using a batch-size of 2 images per GPU. In each iteration we also generate 10 focal views (small crops) of each input image; the random masking ratio has no impact on these views. Using more aggressive masking of the global view progressively reduces device memory utilization and speeds up training.

| Masking Ratio | Mem./GPU | Throughput |
|---|---|---|
| 0.0 | 26G | 415 imgs/s |
| 0.3 | 21G | 480 imgs/s |
| 0.5 | 18G | 525 imgs/s |
| 0.7 | 17G | 600 imgs/s |

Table 9 reports the memory consumption and throughput (imgs/s) of a ViT-L/7 model on a single AWS `p4d-24xlarge` machine using a batch-size of 2 images per GPU. As expected, using more aggressive masking of the global view progressively reduces device memory utilization and speeds up training. For example, by randomly masking 70% of the patches, we can use MSN to pre-train a full-precision ViT-Large with a patch-size of $7 \times 7$ on as few as 18 AWS `p4d-24xlarge` machines. Without masking, the same job requires over 42 machines when using the default batch-size of 1024 images.

## 6 Conclusion

We propose Masked Siamese Networks (MSNs), a self-supervised learning framework that leverages the idea of mask-denoising while avoiding pixel and token-level reconstruction. We demonstrate empirically that MSNs learn strong off-the-shelf representations that excel at label-efficient learning, while simultaneously improving the scalability of joint-embedding architectures. By relying on view-invariant representation learning, MSN does require the specification of data transformations, and it may be that the optimal transformations and invariances are dataset and task dependant. In future work, we plan to explore more flexible mechanisms to learn those transformations and also explore the use of equivariant representations.

## References

1. Assran, M., et al.: Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In: ICCV (2021)
2. Atito, S., Awais, M., Kittler, J.: SiT: self-supervised vision transformer. arXiv preprint arXiv:2104.03602 (2021)
3. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: Data2vec: a general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555 (2022)

4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
5. Bao, H., Dong, L., Wei, F.: BEiT: BERT pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
6. Bardes, A., Ponce, J., LeCun, Y.: VICReg: variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021)
7. Becker, S., Hinton, G.E.: Self-organizing neural network that discovers surfaces in random-dot stereograms. Nature **355**(6356), 161–163 (1992)
8. Bordes, F., Balestriero, R., Vincent, P.: High fidelity visualization of what your self-supervised representation knows about. arXiv preprint arXiv:2112.09164 (2021)
9. Bromley, J., et al.: Signature verification using a "Siamese" time delay neural network. Int. J. Pattern Recognit Artif Intell. **7**(04), 669–688 (1993)
10. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
11. Caron, M., et al.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
12. Chen, M., et al.: Generative pretraining from pixels. In: International Conference on Machine Learning, pp. 1691–1703. PMLR (2020)
13. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. preprint arXiv:2002.05709 (2020)
14. 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)
15. Chen, X., He, K.: Exploring simple Siamese representation learning. arXiv preprint arXiv:2011.10566 (2020)
16. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
17. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: learning augmentation strategies from data. In: CVPR (2019)
18. 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)
19. 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)
20. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
21. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
22. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M.A., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. CoRR (2014)
23. 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)
24. Goyal, P., Mahajan, D., Gupta, A., Misra, I.: Scaling and benchmarking self-supervised visual representation learning. In: ICCV (2019)
25. Grill, J.B., et al.: Bootstrap your own latent: a new approach to self-supervised learning. In: NeurIPS (2020)
26. Gupta, K., Somepalli, G., Anubhav, A., Magalle Hewa, V.Y.J., Zwicker, M., Shrivastava, A.: PatchGame: learning to signal mid-level patches in referential games. In: Advances in Neural Information Processing Systems, vol. 34, pp. 26015–26027 (2021)

27. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
28. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019)
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
30. Hendrycks, D., et al.: The many faces of robustness: a critical analysis of out-of-distribution generalization. In: ICCV (2021)
31. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: Proceedings of the International Conference on Learning Representations (2019)
32. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: CVPR (2021)
33. Joulin, A., Bach, F.: A convex relaxation for weakly supervised classifiers. arXiv preprint arXiv:1206.6413 (2012)
34. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: ECCV (2016)
35. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: CVPR (2017)
36. Li, C., et al.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021)
37. Li, Z., et al.: MST: masked self-supervised transformer for visual representation. Adv. Neural. Inf. Process. Syst. **34**, 13165–13176 (2021)
38. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
39. Lucas, T., Weinzaepfel, P., Rogez, G.: Barely-supervised learning: semi-supervised learning with very few labeled images. preprint arXiv:2112.12004 (2021)
40. Mairal, J.: Cyanure: an open-source toolbox for empirical risk minimization for Python, C++, and soon more. arXiv preprint arXiv:1912.08165 (2019)
41. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. In: CVPR (2020)
42. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by Solving Jigsaw Puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
43. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: CVPR (2016)
44. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
45. Sohn, K., et al.: FixMatch: simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)
46. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 776–794. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58621-8_45
47. 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)
48. Trinh, T.H., Luong, M.T., Le, Q.V.: Selfie: self-supervised pretraining for image embedding. arXiv preprint arXiv:1906.02940 (2019)
49. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)

50. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., Bottou, L.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(12) (2010)
51. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: Advances in Neural Information Processing Systems, pp. 10506–10518 (2019)
52. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. arXiv preprint arXiv:2112.09133 (2021)
53. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
54. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation. arXiv preprint arXiv:1904.12848 (2019)
55. Xie, Z., et al.: SimMIM: a simple framework for masked image modeling. arXiv preprint arXiv:2111.09886 (2021)
56. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: regularization strategy to train strong classifiers with localizable features. In: ICCV (2019)
57. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021)
58. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
59. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
60. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: unsupervised learning by cross-channel prediction. In: CVPR (2017)
61. Zhou, J., et al.: iBOT: image BERT pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021)
62. Zhu, R., Zhao, B., Liu, J., Sun, Z., Chen, C.W.: Improving contrastive learning by visualizing feature transformation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10306–10315 (2021)