Self-supervised visual learning in the low-data regime: a comparative evaluation

Sotirios Konstantakos[a], Despina Ioanna Chalkiadaki[a], Ioannis Mademlis[a], Yuki M. Asano[b], Efstratios Gavves[b], Georgios Th. Papadopoulos[a,*]

[a]*Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece*
[b]*QUVA Lab, University of Amsterdam, Amsterdam, Netherlands*

Abstract

Self-Supervised Learning (SSL) is a valuable and robust training methodology for contemporary Deep Neural Networks (DNNs), enabling unsupervised pretraining on a 'pretext task' that does not require ground-truth labels/annotation. This allows efficient representation learning from massive amounts of unlabeled training data, which in turn leads to increased accuracy in a 'downstream task' by exploiting supervised transfer learning. Despite the relatively straightforward conceptualization and applicability of SSL, it is not always feasible to collect and/or to utilize very large pretraining datasets, especially when it comes to real-world application settings. In particular, in cases of specialized and domain-specific application scenarios, it may not be achievable or practical to assemble a relevant image pretraining dataset in the order of millions of instances or it could be computationally infeasible to pretrain at this scale, e.g., due to unavailability of sufficient computational resources that SSL methods typically require to produce improved visual analysis results. This situation motivates an investigation on the effectiveness of common SSL pretext tasks, when the pretraining dataset is of relatively limited/constrained size. In this context, this work introduces a taxonomy of modern visual SSL methods, accompanied by detailed explanations and insights regarding the main categories of approaches, and, subsequently, conducts a thorough comparative experimental evaluation in the low-data regime, targeting to identify: a) what is learnt via low-data SSL pretraining, and b) how do different SSL categories behave in such training scenarios. Interestingly, for domain-specific downstream tasks, in-domain low-data SSL pretraining outperforms the common approach of large-scale pretraining on general datasets. Grounded on the obtained results, valuable insights are highlighted regarding the performance of each category of SSL methods, which in turn suggest straightforward future research directions in the field.

Keywords: Self-supervised learning, Deep Neural Networks, low-data regime, image analysis, visual representation learning

*Corresponding author

Email addresses: skwnstantakos@hua.gr (Sotirios Konstantakos), it2022112@hua.gr (Despina Ioanna Chalkiadaki), imademlis@hua.gr (Ioannis Mademlis), y.m.asano@uva.nl (Yuki M. Asano), egavves@uva.nl (Efstratios Gavves), G.Th.Papadopoulos@hua.gr (Georgios Th. Papadopoulos)

1. Introduction

The introduction and wide use of large-scale Deep Neural Networks (DNNs) has resulted in revolutionary effects and tremendous advances in the broader field of Artificial Intelligence (AI) over the past decade, by establishing a well-defined methodology where optimal input representations are learnt, with respect to a given task, from massive quantities of training data [1]. In this respect, the computer vision field has significantly benefited from the use of DNNs for tasks such as object detection [2] [3], semantic segmentation [4] [5], image captioning [6], etc. The typical approach is to employ supervised training, where a large number of images tightly coupled with their ground-truth labels are jointly utilized, so that the DNN directly learns the desired downstream task (i.e. a direct mapping from the input images to the desired high-level semantic entities). However, training from scratch, i.e., with randomly initialized DNN parameters, usually requires extremely large annotated datasets and a significant amount of computational power. Both of these resources are practically limited in the majority of real-world cases and for most computer vision tasks. Notably, ground-truth labels can be scarce or expensive to obtain for large datasets: manual annotation may be time-consuming, costly, requiring specialized expertise, and/or contaminating the data with human biases and subjective judgments.

The typical approach for developing a visual analysis module for a real-world application relies on the adoption of the so-called 'transfer learning' paradigm, i.e. pretraining a DNN (from scratch) on a large-scale annotated dataset and then fine-tuning it in the targeted downstream task; both steps follow a supervised learning methodology, i.e. they require the availability of ground-truth annotated datasets. The most common approach by far is to pretrain for whole-image classification on ImageNet-1k [7]. ImageNet-1k is composed of approximately 1.3 million manually annotated images across 1,000 classes. The resulting DNN model encodes knowledge about image semantics and can be exploited as a good initialization for finetuning on any downstream task; this way, comparatively small and limited annotated downstream datasets can be efficiently utilized [8]. ImageNet-pretrained models are publicly available for several well-known DNN architectures, such as variants of the ResNet Convolutional Neural Network (CNN) [9]. However, the more the downstream and the pretraining datasets/tasks differ from each other, the less effective this strategy is [10]. Thus, on the one hand, transfer learning does alleviate the burden of manually annotating a massive number of images for each desired downstream task. On the other hand, the bottleneck of manual labelling remains intact also for the large-scale pretraining dataset, in cases where the publicly available DNN models that have been pretrained on ImageNet-1k (or any other broad/massive dataset) prove to be unsuitable for a given downstream task/domain.

Self-Supervised Learning (SSL) has emerged as an alternative transfer learning strategy, where the DNN is pretrained on a 'pretext task' that does not require ground-truth labels/annotation to be available. Essentially, pseudo-labels are automatically created from the training images themselves [11]. Thus, massive domain-specific datasets can easily be exploited for pretraining, without requiring any human annotation effort; such unlabelled images may be abundant. In fact, SSL has proven beneficial even if ground-truth annotation/labels are available for the massive pretraining dataset, but simply ignored by the SSL pretext task. There are many examples in the literature where SSL pretraining leads to higher downstream accuracy (after task-specific finetuning) than regular supervised pretraining [12]. Table 1 defines the most common relevant terms, while Figure 1 illustrates the conceptualization of the SSL paradigm. Overall, SSL leverages the visual structure and relationships present in the images themselves to provide supervision signals for representation learning, through pretext tasks, such as completing an image

or reconstructing it after applying various transformations (e.g., color alternations, translations, etc.). The typical goal is for the DNN to learn semantically rich image representations without relying on ground-truth labels, although SSL has alternatively been applied to learning features that encode image/style aesthetics [13]. Thorough experimental evaluations have so far showcased that the very large amounts of data allow the DNNs to learn useful image representations during pretraining, even without exploiting ground-truth semantics [14]. To this end, the dominant trend in recent years is to scale SSL regarding both the pretraining dataset size and the DNN model complexity [15], where even ImageNet-1k being considered rather small under contemporary standards.

Although the SSL methodology has proven beneficial in the case of abundance of relevant unlabelled data, it is not always feasible or practical to assemble and/or to utilize very large pretraining datasets in real-world scenarios. In particular, for several application cases (e.g., medical imaging domain) it can be a significant challenge to create such a relevant image pretraining dataset in the order of millions of instances, albeit no manual annotation is required for SSL. Additionally, even in the case that such a sizeable set of relevant images can be collected, it is not always feasible to pretrain DNNs at this scale, e.g., due to unavailability of sufficient computational resources that SSL methods typically require to produce improved visual analysis results.

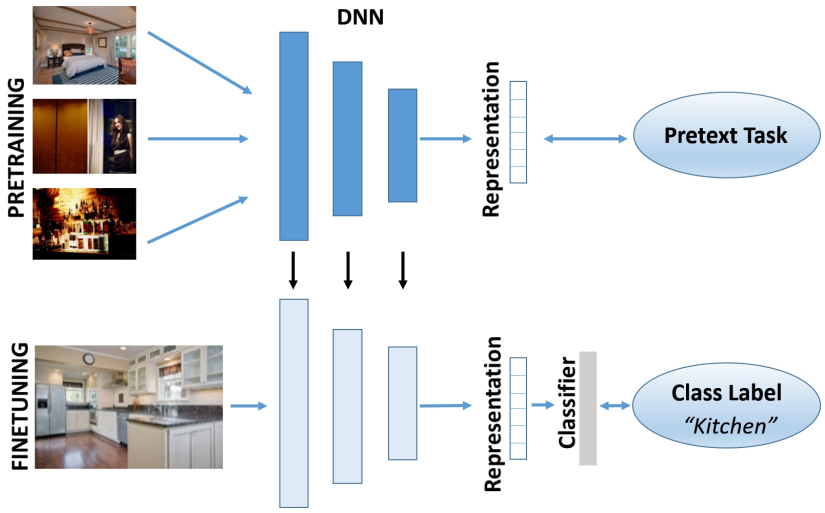


Figure 1: Conceptualization of the SSL paradigm

The challenges and limitations faced when applying the SSL methodology in real-world scenarios provide a strong motivation for investigating the effectiveness of the most common SSL pretext tasks in situations where the pretraining dataset is of relatively limited size. So far, the relevant literature has markedly not explored the behavior of SSL methods when the assumption of abundance of relevant data is not present, since the ability to pretrain on (at least) ImageNet-scale datasets is almost always assumed to be a prerequisite. Yet, a study in the low-data regime would be important, but currently missing, for practitioners who necessarily work with specific image domains (e.g., X-rays), where it is difficult to obtain massive amounts of even unlabeled data.

Table 1: Definition of main terms in self-supervised learning

Term	Definition
Pretext task	An auxiliary task for DNN pretraining, which exploits parts of the data themselves as pseudo-ground-truth. It allows the DNN to learn useful data representations without utilizing actual ground-truth annotation. Different pretext tasks can be alternatively employed for a given pretraining dataset.
Downstream task	The desired application (e.g., whole-image classification, object detection, etc.) to which the pretrained DNN will be adjusted. A single pretrained DNN can be adjusted for multiple different downstream tasks.
Downstream dataset	Each downstream task implies the existence of a relevant downstream dataset, typically accompanied by actual relevant ground-truth annotation. The downstream dataset is usually significantly smaller than the pretraining one.
Pseudo-labels	Pseudo-ground-truth labels automatically generated from the pretraining data themselves, based on the type of the employed pretext task.
Pretrained model	A DNN pretrained on a dataset without exploiting ground-truth labels, using a pretext task.
Finetuning	The process of adjusting the parameters of a pretrained DNN, by partially retraining it on a specific downstream task.
Transfer learning	The act of exploiting data representations learnt via pretraining a DNN, while finetuning it for a downstream task. The most common approach is to simply initialize the DNN architecture with the pretrained model's parameters, before finetuning it.
Linear probing	A common method for evaluating the quality of learnt data representations, by training a linear classifier on top of a frozen pretrained DNN. This is the simplest downstream benchmark, since the limited capabilities of the linear classification head highlight the quality of the learnt representations.

This work attempts to remedy this gap in the literature and to conduct a thorough comparative evaluation of self-supervised visual learning methods in the low-data regime. For the purposes of this study, 'low-data' has been selected to mean a pretraining dataset with a size in the range of 50k - 300k images, which is significantly smaller than the typical dataset sizes used for SSL pretraining (e.g., ImageNet-1k that contains 1.2 million labeled images). The main goals of this study is to identify: a) what is learnt via SSL pretraining in domains where there are no (extremely) large-scale datasets available, and b) how do different SSL categories of methods behave in such training scenarios. To this end, the main contributions of this work are the following ones:

- A taxonomy of modern SSL methods is introduced, accompanied by detailed explanations and insights regarding each main category of approaches. This classification offers clarity regarding the variety of SSL methods currently available, as well as shed light on the advantages and disadvantages of each algorithmic category.
- An in-depth comparative experimental evaluation is conducted under the assumption of performing pretraining in the low-data regime, using one representative pretext task from each of the 4 most common SSL categories. Downstream accuracy metrics are supplemented by visual/qualitative explanations and statistical analyses of the image representations learnt by the various SSL methods. Comparisons are made against traditional baselines, such as supervised pretraining approaches or training from scratch with random parameter initialization, depending on the specific setup.
- Grounded on the obtained results, valuable insights are highlighted regarding the performance of each category of SSL methods (or individual SSL approaches), which in turn suggest straightforward future research directions in the field.

Although SSL pretext tasks can be designed and employed for many different types of data (e.g., timeseries [16], text [17], video [18] [19], audio [20], point clouds [21], or even multimodal data [22] [23]), this article focuses on image analysis for computer vision applications. Moreover, it focuses on generic SSL methods and not ones explicitly designed for specific tasks (e.g., for multi-view clustering [24], product attribute recognition [25], etc.). The remainder of this paper is organized as follows: Section 2 presents a taxonomy of the most common pretext tasks for visual SSL, accompanied by detailed algorithmic explanations for each category. In Section 3, the experimental evaluation methodology and all pretraining or downstream datasets/tasks utilized in this work are detailed. Section 4 illustrates and discusses the obtained results, emphasizing on the extraction of valuable observations and insights. Finally, conclusions about the potential of SSL in the low-data regime are drawn in Section 5.

2. Pretext Tasks

Pretext tasks are auxiliary objectives/assignments that a DNN learns during SSL pretraining, without relying on explicit ground-truth annotation/labels. Each pretext task essentially defines a way to extract pseudo-labels from the training images themselves, so that a DNN being trained for this task gradually learns to capture the visual structure and regularities of the dataset. Simple, common tasks of this nature involve, for example, the prediction of the rotation angle of an image [26] or the relative position of image blocks [27], after applying to the input image a rotation transformation or splitting it into rectangular blocks, respectively. Pretraining the DNN

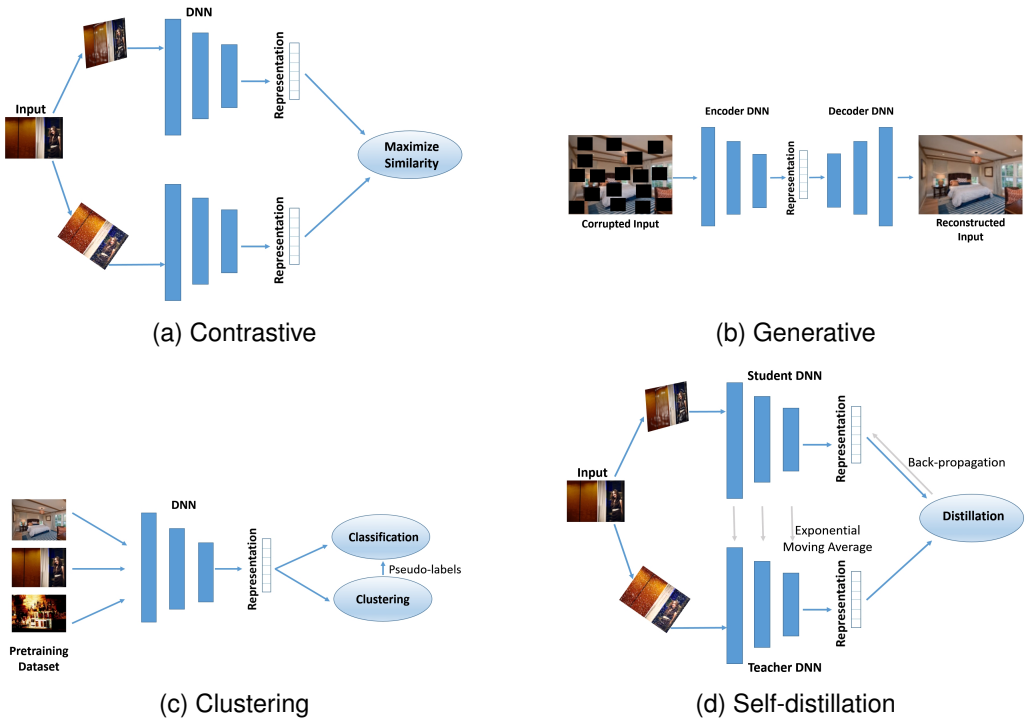


Figure 2: Abstract indicative illustrations of the 4 main categories of SSL pretext tasks

for such a pretext task allows it to learn meaningful representations from unlabeled data without any explicit supervision. This knowledge can then be exploited via regular supervised finetuning of the SSL-pretrained DNN model on a separate annotated downstream dataset, designed for the actually desired task/function. The ability of SSL pretraining to increase accuracy in downstream tasks, compared to direct training from scratch with a randomly initialized DNN model or even to apply transfer learning subsequently to traditional supervised pretraining, depends heavily on the quality and meaningfulness of the considered pretext task.

Taking into account the different types of pretext tasks employed during SSL pretraining, literature methods can be divided into four main categories, namely contrastive, generative, clustering and self-distillation. Figure 2 abstractly summarizes how the aforementioned main categories of SSL pretext tasks operate, based on key representative approaches per case. It needs to be highlighted that in all cases the actual goal of SSL pretraining is for the DNN to learn semantically meaningful visual representations from the input images, instead of simply aiming at solving the pretext task itself. The main SSL pretext task categories are as follows:

- Contrastive: The aim is for the DNN to learn to differentiate between similar (positive) and dissimilar (negative) images. Since no ground-truth semantics/class labels are exploited, the positive pairs are constructed by simply applying data augmentation transformations to an anchor image, which modify its appearance but not its semantic content. In contrast, a negative pair is constructed by simply selecting two different training images. The DNN is trained to generate internal representations that are close to/away from each other in the

latent space for positive/negative image pairs, respectively, via contrastive loss functions such as InfoNCE [28].

- Generative: The DNN is trained to output information reflecting the input image. Typically, the latter has been modified so that it is partially occluded, missing or transformed. Then, the target output is either the original, unmodified image, or a vector encoding the transformation that has been applied to the input. Architecturally, variants of Denoising Autoencoders [29] are the most common form of DNN to be utilized; these consist of an Encoder and a consecutive Decoder subnetwork, with the Encoder being typically the one which is retained for downstream finetuning. Other generative neural architectures have also been employed, such as Variational Autoencoders (VAEs) [30] or Generative Adversarial Networks (GANs) [31]. When the input is partially missing/occluded and the desired target output is the full original image, generative pretext tasks are essentially image reconstruction tasks.

- Clustering: Clustering pretext tasks typically employ an image clustering step, using any clustering algorithm such as K-Means [32], which groups the set of training images into a number of clusters. Then, during SSL pretext pretraining, these cluster assignments are utilized as pseudo-labels for supervised classification. Alternating phases of clustering and SSL, as well as on-line clustering approaches have also been attempted.

- Self-distillation: In neural distillation, a 'student' DNN is trained by utilizing the predictions of another 'teacher' DNN as target pseudo-ground-truth for a given input [33]. In SSL self-distillation pretext tasks, the teacher and the student are usually different variants of the single DNN model being pretrained; for instance, during different epochs or stages of training. The target of the self-distillation loss may be the latent image representation itself, while a pair derived from a single training image (e.g., the original and one transformed/augmented variant) may be utilized as respective input for the two DNNs.

The most commonly employed loss functions for SSL DNN pretraining in computer vision are summarized in Table 2. In Cross-Entropy (CE), which is used for classification, C is the total number of classes, $\mathbf{y}, \mathbf{p} \in \mathbb{R}^C$ denote the target and predicted output vectors, respectively, while y_c/p_c is the ground-truth/predicted probability corresponding to the c-th class. In the cosine similarity loss function, $\mathbf{v}_1$ and $\mathbf{v}_2$ are two feature vectors whose similarity is being assessed. In the InfoNCE loss, $sim(\cdot)$ typically corresponds to cosine similarity, $\mathbf{a} \in \mathbb{R}^L$ and $\mathbf{p} \in \mathbb{R}^L$ are the representation vectors of an anchor and a 'positive' image belonging to the same class, respectively, while $\mathbf{R} \in \mathbb{R}^{L\times(B+1)}$ contains in its columns $\mathbf{r}_k$, $1 \leq k \leq B+1$, the vector $\mathbf{p}$ and the representation vectors of B 'negative' images. For both Mean Squared Error (MSE) and Mean Absolute Error (MAE), $\mathbf{v} \in \mathbb{R}^L$ and $\hat{\mathbf{v}} \in \mathbb{R}^L$ are the pseudo-ground-truth and the predicted/reconstructed vectors, respectively, while L is typically the dimensionality of an image representation.

2.1. Generative pretext

During the early days of visual SSL for DNNs, (relatively) simple generative methods comprised the dominant approach. Research in this algorithmic category is still active, although alternative approaches have also attracted similar or increased attention. Key representative generative SSL methods are described in the followings.

Table 2: Summary of most commonly employed loss functions in visual SSL

Loss function	Details
Cross-Entropy (CE)	*Expression:* $\mathcal{L}_{\text{CE}}(\mathbf{y}, \mathbf{p}) = -\sum_{c=1}^{C} y_c \log(p_c)$ *Applications:* Clustering, self-distillation. *Properties:* Highly sensitive to class imbalance; Target outputs can be self-assigned through clustering or matched by self-distillation.
Cosine similarity	*Expression:* $\mathcal{L}_{\text{sim}}(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1^T \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}$ *Applications:* Contrastive, self-distillation. *Properties:* Robust to feature scaling; Commonly used to measure similarity between representation vectors.
InfoNCE (Noise Contrastive Estimation) [34]	*Expression:* $\mathcal{L}_{\text{NCE}}(\mathbf{a}, \mathbf{p}, \mathbf{R}) = -\log\left(\frac{\exp(\text{sim}(\mathbf{a},\mathbf{p}))}{\sum_{k=1}^{B+1}\exp(\text{sim}(\mathbf{a},\mathbf{r}_k))}\right)$ *Applications:* Contrastive training in visual SSL. *Properties:* Demands large dataset size for meaningful results; Typically used in combination with stochastic image augmentation functions for generating 'positive' images.
Mean Squared Error (MSE)	*Expression:* $\mathcal{L}_{\text{MSE}}(\mathbf{v}, \hat{\mathbf{v}}) = \frac{1}{L}\sum_{i=1}^{L}(v_i - \hat{v}_i)^2$ *Applications:* Self-distillation, image reconstruction generative SSL *Properties:* Sensitive to outliers; Used in feature alignment and for shaping targets in image reconstruction generative pretext tasks.
Mean Absolute Error (MAE)	*Expression:* $\mathcal{L}_{\text{MAE}}(\mathbf{v}, \hat{\mathbf{v}}) = \frac{1}{L}\sum_{i=1}^{L}\|v_i - \hat{v}_i\|$ *Applications:* Image reconstruction generative SSL. *Properties:* Lower sensitivity to outliers;Used for shaping targets in image reconstruction generative pretext tasks.

2.1.1. Rotation Prediction

Rotation prediction [26] is a simple SSL pretext task where the DNN must predict the rotation that has been artificially applied to a given input image. Therefore, the degree of rotation is utilized as a pseudo-ground-truth label. Typically, each training image is rotated to one of the K predefined angles (e.g., 0°, 90°, 180°, 270°) and the DNN is trained to classify the image into one of these rotation classes. It does not correct/undo the pre-applied rotation, but simply identifies it. From a representation learning standpoint, pretraining on this task forces the DNN model to learn basic image semantics [35], despite its simplicity.

Categorical Cross-Entropy is most commonly employed as the classification loss function for rotation prediction, using the actually pre-applied rotation as pseudo-ground-truth label:

$$\mathcal{L}_{Rot} = \mathcal{L}_{CE}(\mathbf{y}, \mathbf{p}), \tag{1}$$

where $\mathbf{y} \in 0, 1^K$ is the one-hot-encoded label vector corresponding to rotated input image $\mathbf{X} \in \mathbb{R}^{W\times H\times 3}$ and $\mathbf{p} \in \mathbb{R}^K$ is the output vector of predicted probabilities given $\mathbf{X}$.

After pretraining, the classification neural head can be discarded and the pretrained backbone DNN may be employed for downstream finetuning.

2.1.2. Colorization

Colorization [36] involves converting grayscale images back into their original colored versions, which are utilized as pseudo-ground-truth. By pretraining on this task, the DNN learns about the natural color distributions and correlations in images. Thus, in order to colorize them

correctly, the DNN learns to encode their semantic context (e.g., skies are blue, foliage is green, etc.).

The loss function employed for colorization typically is the mean squared error (MSE) between the DNN's output $\mathbf{P} \in \mathbb{R}^{W \times H \times 3}$ and the original colored image $\mathbf{Y} \in \mathbb{R}^{W \times H \times 3}$:

$$\mathcal{L}_{Col} = \mathcal{L}_{MSE}(\mathbf{P}, \mathbf{Y}). \tag{2}$$

In this task, $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$ is the respective, grayscale input image. After pretraining, the colored image prediction head can be discarded and the pretrained feature extractor finetuned for downstream tasks. The head is typically an entire Decoder subnetwork.

2.1.3. Super-resolution

Super-resolution [37] is an SSL pretext task where the DNN learns to map low-resolution images into their high-resolution original versions, which are exploited as pseudo-ground-truth. This is not simple upsampling, since the DNN must learn to infer realistic fine details; to achieve this, it must gradually learn to encode image semantics that are necessary for a plausible reconstruction of the original high-resolution image.

The typically employed loss function is the Mean Squared Error (MSE) between the predicted high-resolution image $\mathbf{P}$ and the original high-resolution image $\mathbf{Y}$:

$$\mathcal{L}_{SR} = \mathcal{L}_{MSE}(\mathbf{P}, \mathbf{Y}). \tag{3}$$

In this task, $\mathbf{X} \in \mathbb{R}^{w \times h \times 3}$, where $w < W$ and $h < H$, is the respective, low-resolution input image. After pretraining, the high-resolution image prediction head can be discarded and the pretrained feature extractor finetuned for downstream tasks. The head is typically an entire Decoder subnetwork.

2.1.4. BEiT

While the majority of SSL pretext tasks are DNN architecture-agnostic, 'Bidirectional Encoder representation from Image Transformers' (BEiT, [38]) is specifically designed for Vision Transformers [39]. Inspired by the well-known BERT architecture for Natural Language Processing (NLP) tasks [40], it introduces the so-called 'Masked Image Modeling' (MIM) generative pretext task.

This task considers two independent and alternative views of each image: a) a sequence of M non-overlapping image blocks in raw pixel space, which are called 'patches', and b) a sequence of M image region representations in the low-dimensional latent space of a pretrained discrete Variational Autoencoder (VAE) [41], which are called 'visual tokens'. This space has been pre-discretized into a vocabulary of $V = 8,192$ visual words, leading to the assignment of an integer ID $v \in \mathbb{N}, 1 \leq v \leq V$ to each image region, based on its initial latent representation, via a nearest neighbour look-up [42]. Thus, each visual token ends up being an integer index to the vocabulary. The DNN input for a given image $\mathbf{Y}$ is the latter one's corresponding sequence of patches, after approximately 40% of them have been masked and replaced by a special mask embedding. The respective pseudo-ground-truth target is the sequence of visual tokens for the masked patches of $\mathbf{Y}$. Thus, during MIM pretraining, the DNN must learn to generate the sequence of visual tokens corresponding to the hidden parts of each $\mathbf{Y}$, given as input both its masked and its visible patches.

Obviously, the first subnetwork of the pretrained VAE, which is called 'tokenizer', must be available a priori for BEiT pretraining to proceed, since it is utilized to produce the pseudo-ground-truth targets for all training images. Reliance of the image reconstruction on the discrete visual tokens space, instead of the input raw pixel space, was proposed by BEiT as more efficient because it eliminates high-frequency visual details/noise of low or no semantic content. Otherwise, pretraining for MIM may focus on undesired, very fine-grained visual learning, due to the typically high spatial redundancy of images.

The loss function employed for BEiT pretraining is:

$$\mathcal{L}_{BEiT} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}_{CE}(\mathbf{y}_i, \mathbf{p}_i), \tag{4}$$

where $K = 0.4M$ is the number of masked patches, $\mathbf{y}_i \in \mathbb{R}^V$ is the pseudo-ground-truth discrete distribution over visual tokens for the i-th masked patch of image $\mathbf{Y}$, while $\mathbf{p}_i \in \mathbb{R}^V$ is the corresponding predicted softmax distribution. The BEiT pseudocode is shown in 1.

Algorithm 1 BEiT pretraining pseudocode example for a single training image

Require: $\mathbf{Y}$ - Input image, $P(\cdot)$ - Patch extraction function, $E(\cdot)$ - Embedding function, $T(\cdot)$ - ViT.

1: Divide $\mathbf{Y}$ into M image patches $\mathbf{b}_i$ using P and generate corresponding visual tokens $\mathbf{y}_i$.
2: Mask approximately 40% of the patches in a blockwise manner, resulting in a masked image $\hat{\mathbf{Y}}$.
3: Replace the masked patches in $\hat{\mathbf{Y}}$ with learnable mask embeddings using E, creating a corrupted image $\tilde{\mathbf{Y}}$.
4: Pass $\tilde{\mathbf{Y}}$ through T and obtain M encoded patch representations $\mathbf{H} \in \mathbb{R}^{L \times M}$, where $\mathbf{h}_i \in \mathbb{R}^L$ is the i-th patch representation.
5: For each masked patch representation, obtain $\mathbf{p}_i = softmax(\mathbf{W}_c \mathbf{h}_i + \mathbf{b}_c)$, where $\mathbf{W}_c \in \mathbb{R}^{V \times L}$ and $\mathbf{b}_c \in \mathbb{R}^V$ are trainable parameters.
6: Compute $\mathcal{L}_{BEiT}$.
7: Back-propagate from $\mathcal{L}_{BEiT}$ to update parameters of $T(\cdot)$, $\mathbf{W}_c$ and $\mathbf{b}_c$.

After pretraining, a task-specific neural head may be appended and the overall DNN can be finetuned for downstream tasks.

2.1.5. MAE

MAE [43] (Masked AutoEncoders) is an alternative image reconstruction generative SSL pretext task, also designed mainly for Vision Transformers. In contrast to BEiT, it operates only in the raw pixel space and, thus, does not require any auxiliary neural networks. This is achieved by masking a significantly larger-than-typical proportion of the input image $\mathbf{Y}$ (75%) and employing an entire Decoder subnetwork as a neural head for reconstructing the original image directly in pixel space. Importantly, the Encoder and the Decoder are asymmetric in model complexity.

In MAE pretraining, the Encoder only receives as input the non-masked image blocks and encodes them into a set of respective feature vectors. The lightweight Decoder receives the sequence of these feature vectors along with the appropriately placed mask embeddings. The overall DNN is trained so that the Decoder learns to output the full/uncorrupted original image $\mathbf{Y}$, which is utilized as pseudo-ground-truth target. Overall, the very high masking ratio,

the very lightweight Decoder and the postponement of mask embedding processing until the decoding stage function synergistically so that the Encoder gradually learns semantically rich representations, instead of unimportant high-frequency visual details/noise, at comparatively low computational/memory demands. Essentially, masking 75% of the input pixels makes pixel-level reconstruction challenging, despite the inherently high spatial redundancy of typical images, and, thus, the DNN is compelled to learn rich representations during pretraining.

The loss function employed for MAE is typically the MSE between the DNN's output $\mathbf{P} \in \mathbb{R}^{W \times H \times 3}$ and the original image $\mathbf{Y} \in \mathbb{R}^{W \times H \times 3}$:

$$\mathcal{L}_{Mas} = \mathcal{L}_{MSE}(\mathbf{P}, \mathbf{Y}). \tag{5}$$

However, in contrast to typical applications of $\mathcal{L}_{MSE}$, only the pixels corresponding to masked image regions are evaluated within the loss function; the unmasked/visible input patches do not contribute to loss calculations. The MAE pseudocode is shown in 2.

Algorithm 2 MAE pretext task pseudocode example for a single training image

Require: $\mathbf{Y}$ - Input image, $E(\cdot)$ - Encoder function mapping a corrupted version of $\mathbf{Y}$ to a sequence of latent space vectors $\mathbf{z}_i$, $D(\cdot)$ - Decoder function mapping the sequence of all $\mathbf{z}_i$ plus their positional embeddings back to $\mathbf{Y}$.

1: Corrupt $\mathbf{Y}$ by masking 75% of its pixels, to generate $\hat{\mathbf{Y}}$.
2: Compute all $\mathbf{z}_i$ and their positional embeddings from the non-masked regions $\hat{\mathbf{Y}}$: $\{\mathbf{z}_1, \ldots, \mathbf{z}_i, \ldots\} \leftarrow E(\hat{\mathbf{Y}})$.
3: Compute the reconstructed image: $\mathbf{P} \leftarrow D(\{\mathbf{z}_1, \ldots, \mathbf{z}_i, \ldots\})$.
4: Compute loss $\mathcal{L}_{Mas}(\mathbf{P}, \mathbf{Y})$.
5: Back-propagate from $\mathcal{L}_{Mas}$ to update the parameters of $\mathbf{E}$ and $\mathbf{D}$.

After pretraining, the Decoder can be discarded and the pretrained Encoder DNN may be exploited for downstream finetuning.

2.1.6. I-JEPA

I-JEPA ('Image-based Joint-Embedding Predictive Architecture') [44] is a generative SSL DNN pretraining approach that relocates image reconstruction from the pixel space to the latent/embedding space of abstract image representations. The motivation is to render the learnt features more semantically-oriented and less dependent on the pixel-level image details. The method is designed by default for ViT neural architectures and can be considered a modification of MAE where, for each training image, the single input, the M predicted outputs and the M respective pseudo-ground-truth targets are representation vectors in the embedding space. The targets are representations of M potentially overlapping, randomly sampled image blocks of random aspect ratios and scales (each one occupying 15%–20% of the full image). They are acquired by appropriately cropping the complete image embedding M times, according to the corresponding masks. Similarly, the input is the representation vector of a randomly sampled image block, initially corresponding to a significant portion of the image (at least 85%), but subsequently trimmed so that all of its regions spatially overlapping with the selected targets have been removed. Any original image portions *not* included in the final input image block have been essentially masked out.

The main DNN is composed of two ViTs arranged in an Encoder-Decoder scheme. The so-called 'Context Encoder' receives the single raw input image block and generates its repre-

sentation in the latent space. The Decoder[1] receives this embedding, along with tokens encoding the position in pixel space of the M target blocks, and predicts the embeddings corresponding to these target blocks. During pretraining, the M pseudo-ground-truth target representations are derived by a separate copy of the Encoder ViT called 'Target Encoder', which receives the complete raw input image, generates the respective embedding and spatially crops it M times according to the selected target image blocks. The Context Encoder and the Decoder are trained in the typical manner, via error back-propagation and a Stochastic Gradient Descent variant, but the Target Encoder is simply updated over the training iterations with an exponential moving average of the Context Encoder's parameters[2].

The main advantage of I-JEPA is that, since it operates in the relatively low-dimensional latent space of representations, it is inherently more computationally efficient than MAE that functions in the raw pixel space. Additionally, since I-JEPA combines a variant of the image reconstruction task with abstract image representations, the embeddings learnt by the pretrained DNN strike by design a balance between capturing high-level semantics and encoding low-level image information. This suggests that I-JEPA is an SSL pretraining method particularly suitable for a wide range of downstream tasks, from high-level whole-image classification to low-level dense image prediction (e.g., monocular depth estimation).

Assuming M target blocks per training image[3], the loss function employed for I-JEPA is the average MSE between the M Decoder predictions in the latent space and the respective pseudo-ground-truth targets:

$$\mathcal{L}_{IJEPA} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{MSE}(D(\mathbf{z}_i), \mathbf{t}_i), \tag{6}$$

where $D(\mathbf{z}_i)$ is the Decoder's prediction for the i^{th} block and $\mathbf{t}_i$ is the corresponding pseudo-ground-truth target for that block.

After pretraining, the Decoder and the Target Encoder can be discarded, while the pretrained Context Encoder DNN may be exploited for downstream finetuning.

2.2. Contrastive pretext

Contrastive pretext tasks force the DNN to learn to differentiate between similar (positive) and dissimilar (negative) inputs. Typically, each pair of similar inputs is constructed by randomly augmenting/transforming a training image; then, the original ('anchor') and the transformed version comprise a positive pair. In contrast, a negative pair is composed of two different training images. The goal of contrastive pretext pretraining is for the DNN to learn an image representation that is invariant to appearance and/or geometric transformations (e.g., rotation, translation, scaling, cropping of multiple differently-sized image regions, etc.); therefore focusing on the semantic content [46]. The semantically inconsequential nature of such transformations, at least with respect to whole-image classification decisions, can be considered a form of prior knowledge.

The pretraining procedure relies on a contrastive loss function, such as InfoNCE [28] or NT-Xent [47], that maximizes the similarity of DNN-encoded latent representations between the

[1] Called 'Predictor' in [44].

[2] This design choice seems to have been borrowed from earlier contrastive SSL methods, such as [45]

[3] M is a fixed hyperparameter and [44] suggests $M = 5$.

anchor and the positive input, while simultaneously minimizing the respective representational similarity between the anchor and the negative input. Notably, this contrastive SSL learning strategy may require the discrimination between each anchor and a multitude of visually similar though negative samples, a fact that renders the learning task challenging. After pretraining, the DNN has learnt to encode semantically similar/dissimilar inputs close to/away from each other, respectively, in its internal latent space. Key representative contrastive SSL methods are described in the followings.

2.2.1. Early methods

Contrastive visual representation learning has a long history [48], but the concept of essentially treating each image as a unique semantic class appeared in [49]. This approach poses a significant shift in granularity; from actual classes to individual instances. With the exponential growth in dataset size, maintaining instance-level representations becomes challenging. This led to the use of a 'memory bank' [50] for storing each instance representation vector; the result is a streamlining of the training process and more efficient negative sampling. Subsequent algorithms extended the concept and the use of memory banks [36] [51] [52] [53].

However, typical contrastive methods with memory banks depend on eventually inserting the representations of all the training images into the cache, as computed by the DNN at their last sampling during the pretraining process. This gives rise to scaling and consistency issues. Thus, an alternative approach eventually emerged: in-batch negative sampling. Instead of using a cache structure, these methods exploit as negative samples the other images present in the same training mini-batch as the anchor [54] [55] [56], at the cost of requiring very large mini-batch sizes for proper pretraining.

2.2.2. MoCo

MoCo ('Momentum Contrast') [45] is a contrastive SSL pretraining algorithm that employs a large, fixed-size queue, as a slowly refurbished memory bank of negative samples, and two neural structures: the so-called 'Momentum Encoder' (ME) and 'Query Encoder' (QE). The ME is a copy of the QE that is updated over the training iterations with a slow moving average of the QE's parameters. The QE is the main DNN which is being pretrained and it is regularly updated by common training; during inference, it computes the representation of the input image.

During training with a positive pair, the anchor/transformed image is passed through the QE/ME, respectively, to generate the corresponding representations. Optimizing with the contrastive loss gradually leads to: a) maximizing the similarity between the anchor and the transformed image representations, and b) minimizing the similarity between the embedding of the anchor and those of the negative samples extracted from the memory bank. The latter one is constantly being replenished with newer image representations derived by the ME. The slowly progressing update of the ME parameters allows the representations it generates to be relatively consistent across training iterations, while the fixed queue size allows scaling to very large datasets. The employed loss function is the InfoNCE:

$$\mathcal{L}_{\text{MoCo}} = \mathcal{L}_{\text{NCE}}(\mathbf{a}, \mathbf{p}, \mathbf{R}), \tag{7}$$

where $\mathbf{a} \in \mathbb{R}^L$ is the anchor image representation (generated by QE), $\mathbf{p} \in \mathbb{R}^L$ is the corresponding augmented/transformed image representation (generated by ME) and $\mathbf{R} \in \mathbb{R}^{L\times(K+1)}$ is a set of K negative samples randomly selected from the current memory bank along with $\mathbf{p}$.

After pretraining the ME and the memory bank can be discarded, while the trained QE can be utilized for downstream finetuning.

2.2.3. PIRL

PIRL ('Pretext-Invariant Representation Learning') [53] is a contrastive SSL pretraining algorithm that utilizes a regular memory bank for negative sampling and two trainable linear projection heads that can be alternatively appended to the employed DNN. The first/second projection head maps the high-dimensional representation of the anchor image and its augmented/transformed counterpart, respectively, to a low-dimensional space before computing the contrastive loss. Thus, during training, both the anchor image and its transformed variant are passed through the single DNN in its current state, but are then projected by different heads. Additionally, PIRL slowly updates the cached representation of each dataset image as training progresses, as an exponential moving average of the stored representations (from prior epochs) of that image. This is in contrast to MoCo, which applies a slow moving average update on the separate Momentum Encoder's parameters, instead of applying it on the cached image representations, but the rationale in PIRL is similar: to ensure consistency of the representations for the negative samples during training.

The employed loss function is InfoNCE and operates similarly to the MoCo case. After pretraining, the projection heads and the memory bank can be discarded, while the trained DNN can be utilized for downstream finetuning.

2.2.4. SimCLR

SimCLR ('Simple Contrastive Learning of Representations') [47] is a contrastive SSL pretraining algorithm that generates two differently (but randomly) augmented/transformed images from each anchor image and seeks as a pretext task to maximize representational similarity between them. Therefore, each positive pair consists of two differently transformed variants of a single training image; not of such a transformed version and the original anchor. A single DNN is employed for generating all high-dimensional image representations and a single, lightweight, trainable neural projection head, i.e., a shallow, non-linear MultiLayer Perceptron (MLP), is employed for mapping them to a low-dimensional space before computing the contrastive loss. Negative sampling is in-batch (therefore, there is no memory bank) and all negative pairs are also randomly augmented/transformed before passing through the DNN. A large mini-batch size and the composition of multiple strong augmentation transformations have proven instrumental for SimCLR's performance.

The loss function employed in SimCLR is the Normalized Temperature-scaled Cross Entropy (NT-Xent), which is a variant of InfoNCE:

$$\mathcal{L}_{NT-Xent}(\mathcal{B}) = -\log \frac{\exp(\text{sim}(\mathbf{b}_i, \mathbf{b}_j)/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp(\text{sim}(\mathbf{b}_i, \mathbf{b}_k)/\tau)}, \tag{8}$$

where $\text{sim}(\cdot)$ is the cosine similarity, τ is a scalar temperature hyperparameter and $\mathcal{B}$ is the set of B images in the current training mini-batch. $\mathbf{b}_i$ and $\mathbf{b}_j$ are the latent representations of two differently augmented transformations of the in-batch image currently utilized as an anchor. Each image in the mini-batch is similarly augmented twice, resulting in $2B$ representations for negative sampling. The indicator function $\mathbb{1}_{[k\neq i]}$ is 1 when $k \neq i$ and 0 otherwise. The SimCLR pseudocode is shown in 3.

After pretraining, the projection head can be discarded and the trained DNN can be utilized for downstream finetuning.

Algorithm 3 SimCLR pseudocode for a single mini-batch

Require: Mini-batch $\mathcal{B} = \{\mathbf{B}^1, \mathbf{B}^2, ..., \mathbf{B}^B\}$, Encoder $g(\cdot)$, Projection head $f(\cdot)$, Temperature τ.

for each $\mathbf{B}^i \in \mathcal{B}$ **do**
  Generate two augmented instances $\mathbf{B}^i_1, \mathbf{B}^i_2$ from $\mathbf{B}^i$.
  Encode augmented instances: $\mathbf{h}^i_1 \leftarrow g(\mathbf{B}^i_1)$, $\mathbf{h}^i_2 \leftarrow g(\mathbf{B}^i_2)$).
  Project to lower dimension: $\mathbf{b}^i_1 \leftarrow f(\mathbf{h}^i_1)$, $\mathbf{b}^i_2 \leftarrow f(\mathbf{h}^i_2)$.
end for
for each $i = 1$ to B **do**
  Compute $\mathcal{L}_{NT-Xent}(\mathcal{B})$.
  Back-propagate from $\mathcal{L}_{NT-Xent}(\mathcal{B})$ to update the parameters of $g(\cdot)$ and $f(\cdot)$.
end for7

2.2.5. Barlow Twins

'Barlow Twins' [57] is an SSL pretraining algorithm that relies on a heavily modified version of the common contrastive pretext tasks. It takes its name after the neuroscientist Horace Barlow, who proposed the efficient coding hypothesis [58]. Similarly to SimCLR, Barlow Twins generates two differently (but randomly) augmented/transformed images ('views') from each training image, which are passed through the single DNN being pretrained. For each pair of views derived from one training image, the DNN is pushed to generate respective latent representations that are similar to each other *and* do not contain redundant vector components. To achieve this, the two image view embeddings are treated as random vectors. Thus, their empirical cross-correlation matrix is evaluated during each pretraining iteration, using the generated representations of multiple different images across the entire current mini-batch as different observations. The loss function penalizes the difference of this matrix from the identity one.

The goal is to obtain an image representation that retains as much information about the input as possible, while capturing as little as possible about the particular distortions introduced by augmentation. No negative samples are considered and, therefore, neither a memory bank nor a very large mini-batch size are required. The removal of negative sampling from regular contrastive SSL pretext tasks would obviously create the danger of training converging to a trivial and useless solution, i.e., a DNN that constantly outputs a common representation vector for any input image. In Barlow Twins this 'representation collapse' case is inherently avoided by the inclusion of the redundancy avoidance goal in the loss function, since any collapsed representations within a mini-batch would be heavily cross-correlated and the matrix in question would be far from diagonal. Finally, in contrast to regular contrastive methods, the Barlow Twins objective has been shown to actually require a high-dimensional latent space, in order for the pretrained DNN to perform well in downstream tasks; therefore, no compressive projection head is required. This property potentially stems from the fact that fully decorrelated image representations actually may need a large number of independent components in order to capture the necessary information.

Due to the absence of negative sampling, Barlow Twins does not employ an actual contrastive loss function. Its loss function is the following one:

$$\mathcal{L}_{BT}(\mathcal{B}) = \sum_{i=1}^{L}(1 - C_{ii})^2 + \lambda \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} C_{ij}^2, \tag{9}$$

where $\mathcal{B}$ contains the latent representations of the $2B$ views derived from the current mini-batch

of size B. Thus, $\mathcal{B}$ holds the L-dimensional neural embeddings of two differently augmented views of each one among B randomly sampled training images. $\mathbf{C} \in \mathbb{R}^{L \times L}$ is the empirical cross-correlation matrix of the two views, as computed across the mini-batch. The first term penalizes the deviation of the diagonal entries of $\mathbf{C}$ from 1. The second term, weighted by a hyperparameter λ, penalizes high absolute correlations between different features of the latent representations of the two views (non-zero off-diagonal elements of $\mathbf{C}$).

After pretraining the trained DNN can be directly utilized for downstream finetuning.

2.3. Self-distillation pretext

Neural distillation in general involves two DNNs, a 'teacher' and a 'student', where the latter one is being trained using the predictions of the former for each input as the corresponding pseudo-ground-truth target. In the case of SSL pretraining, the so-called 'self-distillation' paradigm constitutes a category of pretext tasks that have recently emerged from the contrastive SSL methods, but do not require any negative sampling.

In typical formulations, the teacher and the student are of (almost) identical architecture and initialization. During SSL pretraining, their parameters are being updated concurrently, but at a different pace and/or with a different update mechanism, while the student learns to reproduce the latent image representation generated by the teacher for different transformed/augmented versions of the same training image. Eventually, the goal is again to learn an image representation that captures semantic information and is invariant to appearance/geometric perturbations. Key representative self-distillation SSL methods are described in the followings.

2.3.1. BYOL

BYOL ('Bootstrap Your Own Latent') [59] is a self-distillation SSL pretraining method that employs two DNNs: a 'target network' and an 'on-line network'[4]. These are architecturally identical, with the exception of an appended final prediction head in the latter one. The on-line network is actively trained in the typical manner, while the target network's parameters are being slowly updated across the training iterations as an exponential moving average of the corresponding on-line network's parameters. At each iteration, each of the two DNNs receives as input a differently augmented/transformed version of a common training image. The primary pretraining objective is to maximize the representational similarity of these two 'image views' in a low-dimensional projection of the latent space of the two DNNs, across the entire training dataset. Batch normalization is modified so that batch statistics between the two views remain unshared, ensuring independent normalization for each view.

The on-line network learns by taking cues from the target network: as its parameters are being updated with each training iteration (via back-propagation and a Stochastic Gradient Descent variant), the target network receives corresponding updates at a slower pace (via the exponential moving average formulation). This dynamic, combined with the training objective, sets the stage for the on-line network (serving as the student) to 'chase' the target network (serving as the teacher), effectively learning from a slightly older version of itself which is being dynamically constructed. Similarly to SimCLR, a shallow, non-linear MLP is utilized as a projector that maps latent image representations to a low-dimensional space before computing the loss; however, different sets of trainable parameters are attached to the on-line and to the target network's projector.

[4]The terms seem to have been inspired by Deep Q-Learning [60]

BYOL focuses solely on bringing positive pairs close together in the DNNs' latent space, negating the need for negative sampling. This simplifies learning and trims down the computational overhead. This absence of negative samples within the training objective and the very close relationship between the teacher and the student DNN jointly create the possibility of convergence to a trivial solution that perfectly minimizes the training loss in an undesired manner, i.e., a DNN that always outputs identical representation vectors for any and all input images. However, the use of the exponential moving average update mechanism for the target network and of the final prediction head for the on-line network help to avoid this collapse scenario in practice. Finally, BYOL has been shown to be much more robust than SimCLR to the selected mini-batch size and to the choice of augmentation transformations.

Assuming that $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^L$ are the latent representations of two differently augmented views of a single training image, i.e., one encoded by the target network and one by the on-line network, the loss function employed for BYOL pretraining is their Mean Squared Error (MSE):

$$\mathcal{L}_{BYOL} = \mathcal{L}_{MSE}(\mathbf{b}_1, \mathbf{b}_2). \tag{10}$$

After pretraining the teacher/target DNN is discarded, along with the projector and the prediction head of the trained student/on-line DNN. The remaining student backbone can be directly utilized for downstream finetuning.

2.3.2. SimSiam

SimSiam [61] is a self-distillation SSL pretraining method that follows closely on the footsteps of BYOL, but utilizes a Siamese architecture: a single, common DNN is exploited for processing both augmented views of each training image, similarly to SimCLR. This is done by dispensing entirely with the exponential moving average update mechanism and employing identical parameters for the student/on-line and the teacher/target network, except of course for the appended prediction head at the end of the on-line network only. The collapse to a trivial solution is avoided by simply *not* updating the DNN parameters (via regular back-propagation and a Stochastic Gradient Descent variant) based on the loss's gradient with regard to the target network's output. Only the loss gradient back-propagating through the on-line network contributes to DNN parameter update during pretraining. Despite conjectures that have been proposed regarding why this design choice helps to avoid collapsing [62], the reason is not yet fully clear. Nevertheless, the addition of a BYOL-like exponential moving average update mechanism for the target network, which thus becomes fully distinct from the on-line DNN parameter-wise, helps to further increase the achievable downstream accuracy. Baseline SimSiam holds an accuracy advantage over BYOL only when the number of pretraining iterations is limited.

The loss function employed for SimSiam pretraining is the negative cosine similarity of the latent representations of two differently augmented views of a single training image, i.e., $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^L$:

$$\mathcal{L}_{\text{SimSiam}} = -\mathcal{L}_{sim}(\mathbf{b}_1, \mathbf{b}_2). \tag{11}$$

After pretraining, the projector and the prediction head of the single common DNN can be discarded, while the remaining backbone can be directly utilized for downstream finetuning.

2.3.3. DINO

DINO ('DIstillation with NO labels') [63] is a self-distillation SSL method that has been designed with Vision Transformer (ViT) DNNs in mind, unlike the majority of architecture-agnostic SSL algorithms, although it is equally applicable to CNNs. It can be considered a

modified version of BYOL, where a softmax final layer is employed in both the student and the teacher DNN, resulting in the output image representations being probability distributions. Additionally, the final prediction head has been removed from the student/on-line network and, therefore, the student and the teacher are architecturally identical (albeit distinct parameter-wise). Since simple removal of the predictor from the on-line network would make convergence to a trivial/collapsed solution significantly more probable, this undesired scenario is avoided by centering and sharpening the teacher/target network's output before computing the loss. Along with the exponential moving average update mechanism for the target network, this is enough to prevent collapse.

A potential reason is that the composition of centering and sharpening explicitly mitigates *both* potential forms of collapse: a) one where the DNN always predicts a high-entropy uniform representation for any input image, and b) one where the DNN always predicts a low-entropy impulse representation for any input image. Then: a) sharpening is implemented by proper tuning of the softmax temperature hyperparameter, while b) centering is equivalent to adding a dynamically updated bias term to the teacher's output; this bias is computed by considering all teacher outputs for the current mini-batch.

An important property observed with DINO is that the teacher almost constantly outperforms the student during pretraining, on the grounds of the former one being essentially an ensemble learning model. Potentially this is a main contributor to the impressive performance of DINO-pretrained DNNs, especially with ViT neural architectures. In fact, ViTs pretrained with DINO on ImageNet without any supervision have been shown to have learnt representations readily suited for semantic image segmentation. Moreover, they lead to competitive downstream classification accuracy without *any* finetuning, by simply utilizing their representations for building a rudimentary k-Nearest Neighbours (kNN) classifier. Such zero-shot learning abilities are comparable to what SSL-pretrained Large Language Models (LLMs), in the field of NLP, or weakly supervised vision-language DNNs like CLIP [64] are capable of, a fact that has led to the recent emergence of the umbrella term 'foundation models' for all similar cases.

Let us assume that $\mathbf{B}_1^g, \mathbf{B}_2^g$ are two differently augmented 'global' crops of a single training image, i.e., covering a significant portion of the original image's area, while k additional, differently augmented 'local' crops, i.e., covering a substantially smaller portion of the original image's area, are independently and randomly derived. $\mathcal{V}$ is the set of all $k+2$ views obtained from a single training image, which includes $\mathbf{B}_1^g$ and $\mathbf{B}_2^g$ among its elements. All $k+2$ views are passed through the student network $S(\cdot)$, but only the two global views are passed through the teacher network $T(\cdot)$. Then the loss function employed for DINO pretraining is the following one:

$$\mathcal{L}_{DINO} = \sum_{\mathbf{B}_i \in \{\mathbf{B}_1^g, \mathbf{B}_2^g\}} \sum_{\substack{\mathbf{B}_j \in \mathcal{V} \\ \mathbf{B}_j \neq \mathbf{B}_i}} \mathcal{L}_{CE}(T(\mathbf{B}_i), S(\mathbf{B}_j)), \tag{12}$$

where the outputs of $S(\cdot)$ and $T(\cdot)$ are softmax distributions.

After pretraining, the student and the projection head of the teacher DNN can be discarded. The remaining teacher backbone can be utilized for downstream finetuning, or directly for extracting image representations. The DINO pseudocode is shown in Algorithm 4.

DINO has been enhanced with the later version DINOv2, which incorporates various improvements borrowed from competing or earlier methods [65]. The authors have coupled DINOv2 with a very large/complex ViT neural architecture and an extremely large-scale pretraining dataset of 142M images, in order to derive a highly competent foundation model for computer

Algorithm 4 DINO pretext task pseudocode for a single training image

Require: $\mathbf{Y}$ - Input image, $S(\cdot)$ - Student, $T(\cdot)$ - Teacher, $\mathcal{V}$ - Set of $k+2$ augmented views derived from $\mathbf{Y}$, α - Momentum hyperparameter, $\mathbf{c}$ - Centering parameter, $m > 0$ centering update rate hyperparameter, B - Mini-batch size

1: Generate $\mathcal{V}$ from $\mathbf{Y}$, containing k local views/augmented crops and 2 global views/augmented crops.
2: Compute each $S(\mathbf{B}_i)$, where $\mathbf{B}_i$, $1 \leq i \leq (k+2)$ is the i-th element of $\mathcal{V}$.
3: Compute $T(\mathbf{B}_1^g)$ and $T(\mathbf{B}_2^g)$, where $\mathbf{B}_1^g, \mathbf{B}_2^g$ are the two global views within $\mathcal{V}$.
4: Apply centering to $T(\mathbf{B}_1^g)$ and $T(\mathbf{B}_2^g)$: $T(\mathbf{B}_i) \leftarrow T(\mathbf{B}_i) + \mathbf{c}$.
5: Compute $\mathcal{L}_{DINO}$.
6: Back-propagate from $\mathcal{L}_{DINO}$ to update the parameters of $S(\cdot)$.
7: Update the parameters $\mathbf{w}_T$ of $T(\cdot)$ with an exponential moving average of the parameters $\mathbf{w}_S$ of $S(\cdot)$: $\mathbf{w}_T \leftarrow \alpha \mathbf{w}_T + (1-\alpha)\mathbf{w}_S$.
8: Update $\mathbf{c}$: $\mathbf{c} \leftarrow m\mathbf{c} + (1-m)\frac{1}{B}\sum_{j=1}^{B} T(\mathbf{B}_j)$, where $\mathbf{B}_j$ is the j-th image in the current training mini-batch.

vision with very impressive zero-shot learning abilities. The most important algorithmic improvements over DINO are the following ones:

- A complementary image reconstruction training objective is added that is borrowed from iBOT ('image BERT pretraining with On-line Tokenizer') [66], an SSL method that essentially merges BEiT with self-distillation. In iBOT, the teacher serves as an on-line (instead of pretrained) visual tokenizer. Thus, in DINOv2, the image view passing through the student has randomly masked regions, while this is not the case for the image view fed to the teacher. Then, the additional objective is prescribed by a CE loss between the generated features of both networks on the masked regions.

- DINO's centering of teacher outputs is replaced in DINOv2 by an alternative Sinkhorn–Knopp (SK) normalization process [67]. This approach for avoiding solution collapse to a common, low-entropy impulse representation has been borrowed from [68], i.e., a recently proposed pretraining-time ensemble learning method for DINO that computes data-dependant importance weights for ensemble members.

- The KoLeo regularizer [69] has been added to the training objectives. This is a differential entropy regularizer that is employed for spreading the output representations of different images within a mini-batch.

2.4. Clustering pretext

SSL via clustering involves, in general, grouping images into clusters, based on their similarity in the input space, and pretraining a DNN to predict the cluster assignments. Since clustering is an unsupervised task, no actual ground-truth labels are utilized. The cluster assignments are used as pseudo-ground-truth targets. In most cases, any clustering algorithm can be used (e.g., K-Means), but certain SSL methods incorporate a specific on-line clustering approach. The quality of the learnt representation depends on the suitability of the considered clustering algorithm and the similarity metric used to group the images [70]. Key representative clustering SSL methods are described in the followings.

2.4.1. DeepCluster

DeepCluster [71] attempts to structure the high-dimensional latent space of DNNs via an iterative approach of alternating phases/steps. It assumes a single, randomly initialized DNN to which a final classification neural head has been attached for SSL pretraining. Starting with inferring image representations via this DNN for the entire training set, said generated features are then clustered using K-Means. The resultant cluster assignments are utilized as pseudo-ground-truth labels that guide self-supervised DNN training for regular classification in the subsequent epoch. These two phases alternate. As training progresses, the latent space gets refined and this, in turn, provides a more nuanced structure during the succeeding clustering phase. Thus, during each classification step the pseudo-ground-truth labels are kept fixed, but they are updated once for the entire training set during the next clustering step. This cycle continues, allowing the two alternating phases to continually aid each other.

In order to prevent trivial or collapsed solutions to the optimization problems of both phases, DeepCluster employs common tricks for class-imbalanced classification and for avoiding empty clusters. Data augmentation pre-applied to the training set imbibes a degree of representational invariance to non-semantic transformations. Notably, ImageNet-1k training using DeepCluster leads to clusters that often align with human-understandable semantic classes, but the approach is not sufficiently robust to suboptimal clusterings or to the choice of data augmentation transformations.

The loss function employed for the SSL DNN pretraining phases on the classification pretext task is regular Cross-Entropy (CE):

$$\mathcal{L}_{DeepCluster} = \mathcal{L}_{CE}(\mathbf{y}_i, \mathbf{p}_i), \tag{13}$$

where $\mathbf{y}_i/\mathbf{p}_i$ is the pseudo-ground-truth/predicted distribution over K pseudo-classes/clusters, respectively, for the i-th training image.

After pretraining, all metadata generated by the clustering phases and the classification head can be discarded, while the pretrained backbone DNN may be utilized for downstream finetuning. The DeepCluster pseudocode is shown in 5.

Algorithm 5 DeepCluster pretext task pseudocode

Require: $\mathcal{X}$ - Training dataset, $f(\cdot)$ - DNN feature extractor function, $C(\cdot)$ - Clustering function

Initialize parameters of $f(\cdot)$ (e.g., randomly).
while not converged **do**
  Compute set of representations $\mathcal{H}$ containing all $\mathbf{h}_i \leftarrow \mathbf{f}(\mathbf{X}_i)$, $\mathbf{X}_i \in \mathcal{X}$.
  Compute cluster assignments $C(\mathcal{H})$ to obtain set of pseudo-ground-truth-labels $\mathcal{Y}$.
  for each mini-batch $\mathcal{B}$ sampled from $\mathcal{X}$ **do**
    Compute $\mathcal{L}_{DeepCluster}$ for each image in $\mathcal{B}$.
    Back-propagate from $\mathcal{L}_{DeepCluster}$ to update parameters of $f(\cdot)$.
  end for
end while

DeepCluster has been enhanced in DeepClusterV2 [72], which incorporates a number of various improvements. The most important ones are the following:

- Pretraining has been accelerated by removing precautions for avoiding trivial solutions, since this scenario is not commonly observed in practice, and by re-using image representations from the previous epoch at the start of each clustering phase, instead of inferring

them anew using the current DNN. This can be considered a form of memory bank caching without momentum.

- As in [73], multiple clustering problems are simultaneously solved independently and in a multitask fashion. Thus, the classification training phases proceed by the sum of the respective losses, assuming a common, shared set of DNN parameters.
- Strong data augmentation and the MLP projection head are borrowed from SimCLR [47].
- Spherical K-Means is utilized for the clustering phases, instead of vanilla K-Means.

2.4.2. SeLa

SeLa ('Self Labelling') [73] is an algorithm for SSL DNN pretraining via clustering. It can be considered as a modification of DeepCluster, since it also depends on consecutive iterations of two alternating steps. The classification pretext pretraining phase is identical to that of DeepCluster. However, in the clustering phase, instead of simply performing K-Means on all inferred image representations, a replacement method has been devised. Finding the cluster assignment for each image under a constraint of equally sized clusters, for avoiding undesired trivial solutions where all images are placed in a single dominant cluster, is formulated as an optimal transport problem and solved efficiently via the Sinkhorn-Knopp algorithm [67]. The overall two-step iterative method is formally derived as a way to optimize a single objective with regard to both DNN parameters and cluster assignments, a fact that mathematically guarantees convergence to a local optimum. Finally, SeLa pre-applies typical data augmentation schemes to the training set and adopts the multi-clustering approach that can also be found in DeepClusterV2. Multi-clustering is implemented in SeLa by using parallel classification heads (one per clustering task), with a shared backbone DNN.

As in DeepCluster, the loss function actually employed for the classification pretext task is Cross-Entropy (CE). However, the general optimization objective with regard to both DNN parameters and cluster assignments, from which the overall algorithm is derived, is the following one:

$$\min_{p,q} : -\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{K} q(y|\mathbf{X}_i)\log p(y|\mathbf{X}_i) \tag{14}$$

s.t.:

$$q(y|\mathbf{X}_i) \in 0, 1, \forall y \in 1, 2, \ldots, K, \forall i \in 1, 2, \ldots, N, \tag{15}$$

$$\sum_{y=1}^{K} q(y|\mathbf{X}_i) = 1, \forall i \in 1, 2, \ldots, N, \tag{16}$$

$$\sum_{i=1}^{N} q(y|\mathbf{X}_i) = \frac{N}{K}, \forall y \in 1, 2, \ldots, K, \tag{17}$$

where K is the number of pseudo-classes/clusters, N is the number of training images, $p(y|\mathbf{X}_i)$ is the predicted, softmax-derived probability distribution of class labels over the K classes, given the i-th training image $\mathbf{X}_i$, while $q(y|\mathbf{X}_i)$ is the corresponding pseudo-ground-truth distribution over the K classes; q can be considered to be a fully deterministic impulse distribution.

After pretraining, the classification heads can be discarded and the pretrained backbone DNN may be utilized for downstream finetuning.

2.4.3. SwAV

SwAV ('Swapping Assignments between multiple Views of the same image') [72] is an SSL method the combines characteristics from both contrastive and clustering SSL approaches. It does not utilize negative samples and, therefore, does not rely on a memory bank, a momentum encoder and/or a large mini-batch size. More importantly, the vectors that are being pushed close together by its training objective are *not* directly the image representations of differently augmented/transformed views of a single training instance/image, but their cluster assignment vectors from an on-line image clustering process that occurs in parallel. Thus, the goal of DNN pretraining is to arrive at a set of DNN parameters that are optimal in the following sense: they induce such image representations that the cluster assignment vector of one view can be predicted from the latent representation of the other view, and vice versa. Essentially, representations of semantically identical images are gradually forced to become approximately similar to each other so as to generate appropriate, simplified attention weights vectors [74], using each $\mathcal{L}_2$-regularized representation as a query and the K $\mathcal{L}_2$-regularized, learnt cluster centroids within the latent space as a set of keys[5]. These simplified attention weights vectors are forced by the training process to eventually approximate the learnt cluster assignment vectors that, as training proceeds, become consistent with the semantic content and invariant to appearance/geometric transformations. The on-line clustering aspect, where processing takes place in-batch instead of partitioning the entire training set in dedicated clustering phases, allows the method to scale well for very large pretraining datasets.

Within each training iteration, the training objective is optimized in two steps. First, the cluster assignments are learnt for the current mini-batch, assuming constant/fixed cluster centroids and DNN parameters. This is done by solving an optimal transport problem with an equipartition constraint, for avoiding collapse to a trivial solution of one dominant cluster, using the iterative Sinkhorn-Knopp algorithm. The approach is similar to that of [73], but modified here to operate within a mini-batch due to the on-line clustering setting. Second, assuming fixed cluster assignment vectors, the centroids and the neural network parameters are updated by minimizing the loss function in the typical manner (error back-propagation and a Stochastic Gradient Descent variant). This loss function makes use of the Cross-Entropy (CE) between cluster assignment vectors and the simplified attention weights vectors previously defined:

$$\mathcal{L}_{\text{SwAV}} = \mathcal{L}_{CE}(\mathbf{q}_i^{v1}, \mathbf{p}_i^{v2}) - \mathcal{L}_{CE}(\mathbf{q}_i^{v2}, \mathbf{p}_i^{v1}), \tag{18}$$

$$\mathbf{p}_i^{v*} = softmax(\mathbf{C}\mathbf{h}_i^{v*}), \tag{19}$$

where $\mathbf{C} \in \mathbb{R}^{K \times L}$ contains the K $\mathcal{L}_2$-regularized, current cluster centroids and $\mathbf{h}_i^{v1}/\mathbf{h}_i^{v2}$ is the $\mathcal{L}_2$-regularized, L-dimensional latent representation of the first/second augmented view of the i-th training image, respectively. $\mathbf{q}_i^{v1}/\mathbf{q}_i^{v2}$ is the K-dimensional current cluster assignment vector for the first/second augmented view of the i-th training image, respectively.

After pretraining the DNN can be directly utilized for downstream finetuning, while all clustering metadata can be discarded.

2.5. Discussion on the behavior of the SSL pretraining categories

The main categories of SSL methods for image analysis tasks, as presented in this section, are characterized by various trade-offs in different aspects. Two important points of comparison

[5] K is a fixed hyperparameter, as in K-Means.

comprise the following ones: a) the reliance on handcrafted design choices (e.g., augmentation transformations, generative pseudo-ground-truth targets, etc.) as prior knowledge, b) the susceptibility of training to solution/representation collapse modes.

Table 3: A succinct comparison of SSL algorithmic families

Attribute	**Generative**	**Contrastive**	**Self-distillation**	**Clustering**
Prior knowledge (augmentations)	Low	High	High	High
Susceptibility to solution collapse	No	No	Possible	Possible
Negative sampling	No	Yes	No	No
Mini-batch size	Flexible	Large	Moderate	Moderate
Representations robustness	Not evaluated	Moderate	High	High

With respect to the first point, the majority of SSL methods relies on prior knowledge and exhibits various degrees of robustness to such choices. Essentially, the worst offenders are early generative SSL methods (e.g., rotation prediction, colorization, etc.), which can be nowadays considered outdated. Additionally, contrastive, self-distillation and clustering SSL approaches are heavily dependent on data augmentation transformations, which is a form of prior knowledge about the visual medium and/or the specific desired downstream task. The latter observation stems from the fact that not all downstream tasks are equally invariant to the specific image transformations applied during pretraining; e.g., multi-cropping and rotation may be semantically irrelevant for whole-image classification, but not as much in the case of monocular depth estimation. As a result, pretraining with these common augmentation transformations in order to obtain high-level semantic image representations that are invariant to them may be inappropriate for low-level downstream vision tasks. Among the common SSL algorithms, the only ones that do not rely at all on similar forms of prior knowledge are the image reconstruction generative tasks.

The image reconstruction generative SSL approaches also inherently do not suffer from the danger of trivial solutions via representation collapse, which is theoretically an additional advantage. However, in practice, recent methods in categories that are the most susceptible to this scenario, i.e., self-distillation and SSL via clustering, have converged to clever methodological and/or implementation nuisances for essentially bypassing this issue and avoiding collapse altogether. Mainstream contrastive SSL methods, which inherently do not suffer from collapse issues due to negative sampling, typically come with the cost of requiring very large mini-batch sizes or auxiliary structures (e.g., a large memory bank). A well-known exception is Barlow Twins, since it does not use the contrastive loss and does not employ negative sampling, but in fact DNNs pretrained with it struggle to reach good downstream accuracy without the use of very high-dimensional latent representations and a relatively large mini-batch size.

It has to be noted that it is a self-distillation SSL approach, i.e., DINO, that has exhibited strong zero-shot learning capabilities when combined with a complex ViT, with DINOv2 pretrained on an extremely large-scale dataset reaching the zero-shot performance of competing weakly supervised methods. Essentially, very high model complexity and pretraining dataset size contribute significantly to the performance of DINOv2, but nevertheless self-distillation seems to currently dominate the SSL landscape in computer vision. Even so, research on image reconstruction generative approaches and on SSL via on-line clustering is still on-going.

Regarding robustness analysis, several common SSL-learnt representations have been empirically evaluated against downstream distribution shifts and corruptions in [75]. DINO, a self-

distillation method, and SwAV, an approach of SSL via on-line clustering, are found to be the most robust. However, no image reconstruction generative pretext task is considered in that comparative evaluation.

Table 3 succinctly compares the four SSL algorithmic categories presented in this section, excluding the now-outdated early generative methods. The table expresses the prevailing tendencies within the state-of-the-art of each family, with individual methods potentially deviating.

3. Comparative Evaluation Framework in the Low-Data Regime

The goal of this study is to investigate in depth how the different SSL methodologies behave and what eventual downstream recognition performance is achieved, when pretraining occurs in the low-data regime. This stands in contrast to the currently dominant trend of ever-increasing scale, regarding both the pretraining dataset size and the DNN model complexity in SSL research. As argued in Section 1, low-data pretraining remains highly relevant in cases of domain-specific downstream tasks, where generic ImageNet-1k pretraining-derived or zero-shot representations cannot naturally generalize well. Therefore, a thorough empirical comparative evaluation is designed that incorporates all SSL algorithmic categories outlined in Section 2, in order to identify what can be learnt via SSL in low-data computer vision scenarios. As mentioned in Section 1, 'low-data' implies a pretraining dataset with a size in the range of 50k - 300k images, which is significantly smaller than the typical dataset sizes used for SSL pretraining in the literature (e.g., ImageNet-1k).

In the defined evaluation framework, 4 widely used SSL methods from different algorithmic categories are compared under different setups on image classification. The considered approaches are MAE, SimCLR, DINO and DeepClusterV2 that belong to the generative, contrastive, self-distillation and clustering SSL category, respectively. It needs to be highlighted that the 4 aforementioned methods do not necessarily correspond to the 4 best-performing in the literature; however, they are relatively recent, representative of their respective algorithmic category and exhibit performance comparable to the state-of-the-art. The employed neural architecture for each considered SSL method has been selected based on the design choices described in the original paper of each approach; thus, ResNet-50 [9] is used for the cases of SimCLR and DeepClusterV2, while ViT-L/16 [39] is utilized for MAE and DINO. This strategy was preferred because not all SSL methods seem to work equally well with Vision Transformers and each SSL approach has an informally declared 'default' architecture. For the baselines pretrained in a supervised manner, both ResNet-50 and ViT-L/16 variants have been employed. As an additional baseline, an off-the-shelf pretrained ViT-L/14 DINOv2 model has also been evaluated. In all cases, linear probing has been employed for downstream finetuning, instead of end-to-end finetuning, due to its widespread usage in the SSL literature and its ability to demonstrate the quality of the representations learnt during pretraining. To this end, each pretrained model is kept frozen and its inferred features are utilized to train a simple downstream linear classifier. In the followings, this linear probing setup is implied wherever finetuning is mentioned.

In order to better structure the wide set of performed experiments and to facilitate the derivation of key/detailed insights, the experiments in the defined experimental framework are divided into the following main classes: a) Main experiments, which are related to whole-image classification tasks; these are further broken down into i) object- and scene-centric (based on the recognition target of the downstream task) and ii) transfer-learning and single-dataset setup (taking into account whether different datasets are used for the pretraining and downstream tasks or not), b) Robustness experiments, which investigate robustness aspects of the various SSL pretraining

Table 4: Configurations of main experiments

Granularity	Setup	Pretraining	Downstream	Notation
Object-centric	Transfer-learning	ImageNet-100	COCO	M_T^O(IN-100↓COCO)
	Transfer-learning	ImageNet-100	STL-10	M_T^O(IN-100↓STL-10)
	Single-dataset	ImageNet-100	ImageNet-100	M_S^O(IN-100↓IN-100)
Scene-centric	Transfer-learning	Places-100	ADE20K	M_T^S(P-100↓ADE)
	Transfer-learning	Places-100	SUN Database	M_T^S(P-100↓SUN)
	Single-dataset	Places-100	Places-100	M_S^S(P-100↓P-100)

tasks, and c) Domain-specific experiments, which investigate SSL performance in specialized visual domains that differ significantly from typical/conventional natural RBG images.

3.1. Main experiments

The main experiments concern whole-image classification tasks and are divided into: a) object- and scene-centric, taking into account the semantic granularity of the recognition target of the downstream task, and b) transfer-learning and single-dataset, considering the dataset setup (i.e., when the pretraining dataset and the downstream dataset for supervised finetuning differ or not, respectively). ImageNet-100[6] and Places-100[7] are selected as the pretraining dataset for the object- and scene-centric case, accordingly. Additionally, the considered downstream datasets for the transfer-learning setup are COCO [79] and STL-10 [80] for the object-centric case, and ADE20K [81] and SUN Database [82] for the scene-centric one. For the single-dataset setup, where both the SSL pretraining and the subsequent supervised downstream finetuning tasks are conducted on a single dataset, ImageNet-100 and Places-100 are used for the object- and the scene-centric cases, respectively. Table 4 summarizes the configurations of the considered main experiments, while notation is also included for clarity purposes.

Moreover, the employed SSL pretraining methods are quantitatively compared with respect to downstream classification accuracy against baseline benchmarks, so as to better demonstrate the recognition capabilities of the various SSL configurations. To this end, in the transfer-learning setup, two sets of DNN variants pretrained according to the regular supervised approach are considered, namely one pretrained on ImageNet-1k (object-centric) or Places-205 (scene-centric) and one on ImageNet-100 (object-centric) or Places-100 (scene-centric). Both of these exploit supervised pretraining, but the first one, i.e., pretraining on ImageNet-1k or Places-205, is not low-data. Additionally, an off-the-shelf DINOv2 model, which has been distilled from a DNN pretrained on a dataset of 142 million images (again, non-low-data), is also employed as a competing baseline. On the other hand, the single-dataset experiments are also repeated under the semi-supervised downstream finetuning protocol described in [83][84]. This includes two different variants of downstream finetuning, namely one using 5% and one using 10% of the ground-truth class labels.

3.2. Robustness experiments

A complementary set of experiments is conducted in order to evaluate robustness aspects of the considered SSL methods. For that purpose, three typical robustness experimental bench-

[6]This is a low-data variant of the ImageNet-1k dataset [7], transformed according to [76]

[7]This is a low-data variant of the Places-365 dataset [77], transformed to Places-100 according to [78]

marks/protocols of the literature are considered, namely a) Noisy-ImageNet-100, b) Imbalanced-ImageNet-100, and c) VTAB, as detailed below.

Regarding the Noisy-ImageNet-100 benchmark, it involves the evaluation of robustness, using three datasets that have been derived by applying various forms of noise, perturbations and corruptions to a subset of ImageNet-1k or by selecting naturally adversarial images for its classes; the considered datasets are ImageNet-A [85], ImageNet-P and ImageNet-C [86]. For each considered SSL method, the corresponding ImageNet-100-pretrained variant from the main experiments (i.e., after downstream finetuning/linear probing on regular ImageNet-100) is evaluated on each of the three Noisy-ImageNet-100 datasets, which have been derived by removing from ImageNet-A/P/C any classes not included in ImageNet-100.

The Imbalanced-ImageNet-100 dataset concerns the evaluation of resilience against uneven class distribution, using an imbalanced variant of ImageNet called ImageNet-Long-Tail [87]. Thus, for each SSL method the corresponding ImageNet-100-pretrained variant from the main experiments undergoes supervised downstream finetuning on Imbalanced-ImageNet-100, which has been derived by adapting for ImageNet-100 the process applied in [87] to ImageNet-1k.

In order to evaluate the robustness of the learnt visual representations against cross-domain shifts the Visual Task Adaptation Benchmark (VTAB) [88] has been adopted. In particular, for each considered SSL method, the corresponding ImageNet-100-pretrained variant from the main object-centric experiments (Section 3.1) has been finetuned and tested across the 19 VTAB downstream datasets, using the off-the-self VTAB protocol[8]. VTAB evaluates downstream accuracy metrics for three high-level group tasks (each computed based on multiple datasets), namely the 'natural', the 'specialized' and the 'structured' one.

Apart from evaluating the robustness of the considered SSL methods, additional quantitative comparisons include the two baselines used for the main experiments (as detailed in Section 3.1), namely supervised pretraining on ImageNet-1k and on ImageNet-100. As per relevant protocols, downstream finetuning is subsequently conducted for Imbalanced-ImageNet-100, before evaluation, but not for Noisy-ImageNet-100. Additionally, for the case of the off-the-shelf (non-low-data) pretrained DINOv2 baseline, a linear classification head finetuned on ImageNet-100/Imbalanced-Imagenet-100 is utilized for inference in the Noisy-ImageNet-100/Imbalanced-ImageNet-100 experimental benchmark, respectively.

3.3. Domain-specific experiments

Specialized image domains differing significantly from conventional/natural RBG images constitute a particularly compelling evaluation scenario for SSL, due to the relative scarcity of (sufficiently) large domain-specific datasets and/or ground-truth annotations. In order to evaluate whether under such evaluation settings domain-specific pretraining in the low-data regime outperforms generic low-data pretraining on ImageNet-100 and/or large-scale pretraining on ImageNet-1k, in terms of downstream accuracy, the selected SSL methods are evaluated under the transfer-learning setup in three different (and significantly diverse) visual domains: a) remote sensing (high altitude aerial photographs), b) medical imaging (i.e., chest X-rays), and c) security imaging (i.e., airport luggage X-rays). The linear probing finetuning protocol for whole-image classification is employed here as well. In all cases, in-domain SSL pretraining is compared against supervised pretraining on ImageNet-100 and on ImageNet-1k, as well as against an off-the-shelf non-low-data-pretrained DINOv2 baseline.

[8] `https://github.com/google-research/task_adaptation`

(a) (b)

Figure 3: Example images from the MLRSNet remote sensing dataset [89] belonging to classes a) 'Airport' and b) 'Beach'

All domain-specific datasets, utilized either for pretraining or downstream finetuning/evaluation, are low-data in the sense defined in this study. For the remote sensing domain, MLRSNet [89] and AID [91] have been used as the pretraining and downstream datasets, respectively, while indicative exemplary images are shown in Figure 3. Regarding the medical imaging domain, the MedPix [90] and ChestX-Det [92] datasets have been used for pretraining and downstream finetuning, respectively, while example images are shown in Figure 4. Moreover, for the security imaging domain, two different subsets of the SIXRay [93] dataset have been used as the pretraining and downstream subset, namely SIXRay-100 and SIXRay-10, respectively, while indicative examples are shown in Figure 5.

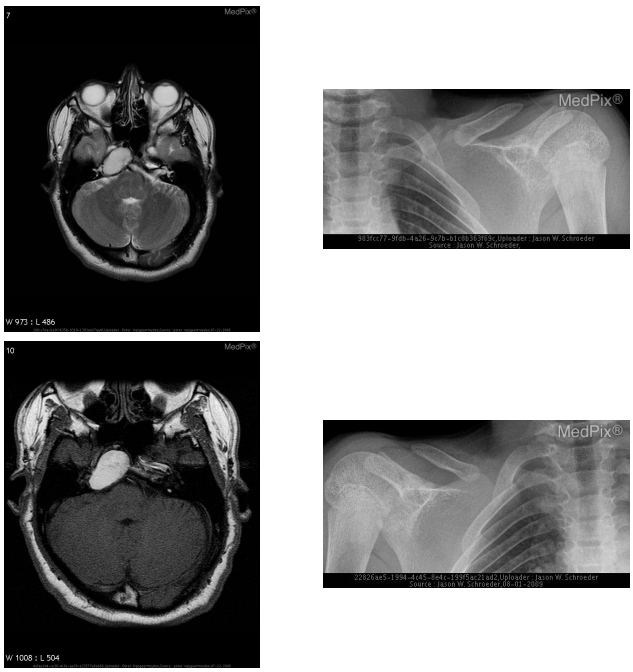
3.4. Employed datasets

A summary of the datasets employed in the conducted experimental study can be found in Table 5, excluding the 19 VTAB ones used for the robustness experiments (Section 3.2); details about them are available in [88]. In certain cases, adjustments needed to be made for the data to be incorporated in the experimental evaluation, mainly targeting to fit to the defined 'low-data regime'. In particular, Places-100 was constructed from Places-365 by randomly sampling 500 images per class, instead of 400 as in [78], for a total of 50,000 images (instead of 40,000) that were then segmented into a training/validation/test set following a 70-15-15 split ratio[9]. For the case of ADE20K, whose official on-line repository contains only a training set and a test set,

[9]In the sequel, this modified variant is implied wherever Places-100 is mentioned

Table 5: Image datasets employed in the evaluation study

Dataset	Description
ImageNet-1k [7]	It contains 1.3 million natural RGB images, uniformly distributed to 1,000 classes. Each image is assigned only one class label, which is typically object-centric. It is split into training/validation/test sets of 1,281,167/50,000/100,000 images, respectively.
ImageNet-100 [76]	It is a low-data variant of ImageNet-1k, retaining 100 classes with 1,300 training and 300 test images per class. It is split to training/validation/test sets of 100,000 (1,000 images per class), 30,000 and 30,000 (300 images per class) images, respectively.
Places-365 [77]	It contains 10 million images distributed across 434 scene-centric classes, having 5,000 to 30,000 training images per class. It is split to training/validation/test sets of 8 million/36,000/328,000 images, respectively.
Places-100 [78]	It is a low-data variant of Places-365, derived by randomly selecting 100 classes and 400 images per class from Places-365. It is split to training/test sets of 30,000/10,000 images, respectively.
STL-10 [80]	It consists of 500 labeled training images, 800 labeled test images and 100,000 unlabeled images, covering in total 10 classes (namely 'Airplane', 'Bird', 'Cat', 'Deer', 'Dog', 'Horse', 'Monkey', 'Ship', 'Truck' and 'Car').
COCO [79]	It contains 164,000 images with 80 classes of objects. Its version utilized in this paper is split to training/validation/test sets of 82,000/41,000/41,000 images, respectively.
ADE20K [81]	It includes 365 scene classesand contains 25,574/2,000 images for training/test, respectively.
SUN Database [82]	The utilized version includes 108,754 scene-centric images falling under 397 classes, with at least 100 images per class. It is split to training/validation/test sets of 76,128/10,875/21,750 images, respectively.
ImageNet-A [85]	It contains 30,000 images falling under 200 classes of ImageNet-1k, which have been selected based on their ability to fool ResNet-50 classifiers trained on ImageNet-1k. The dataset is typically employed only for inference.
ImageNet-P [86]	Modification of the ImageNet-1k validation set, so that each image has been slightly perturbed by the application of various transformations (owing to noise, blur, weather conditions and digital distortions). It contains a total of 200,000 images and is typically employed only for inference.
ImageNet-C [86]	It is a dataset similar to the ImageNet-P one, but with more intense corruptions. It contains a total of 200,000 images and is also typically employed only for inference.
ImageNet-Long-Tail [87]	It has been obtained from ImageNet-1k by sampling using a long tail distribution, so that the resulting classes are imbalanced. It contains 115,846 images, split to training/validation/test sets of 81,092/17,377/17,377 images, respectively.
MLRSNet [89]	It is designed for multi-label remote sensing image classification. It contains 100,000 images, each annotated with multiple labels from a set of 60 classes. It is split to training/validation/test sets of 80,000/10,000/10,000 images, respectively.
AID [91]	It is a collection of high-resolution aerial images for various land-use and land-cover classification tasks. It is split to training/validation/test sets of 8,000/1,000/1,000 images, respectively.
MedPix [90]	It contains 44,000 medical images with annotations for 10 different types of organs and tissues. It includes 32,000 training images, 4,000 validation images, and 8,000 test images.
ChestX-Det [92]	ChestX-Det is an expanded version of the ChestX-Det10 dataset [94], comprising 3578 images sourced from NIH ChestX-14. The dataset covers 13 common categories of chest-related diseases and abnormalities. It is divided to 2,077 training, 756 validation and 756 test images.
SIXRay [93]	It contains 1,059,23 X-ray scan images of airport luggage, with a subset of them depicting prohibited items. Different overlapping subsets have been defined, each with a varying ratio of negative-to-positive images. In this study, two such subsets are utilized: SIXRay-100, containing 742,104/131,769 training/test images, respectively, and SIXRay-10, containing 67,464/11,979 training/test images, respectively.



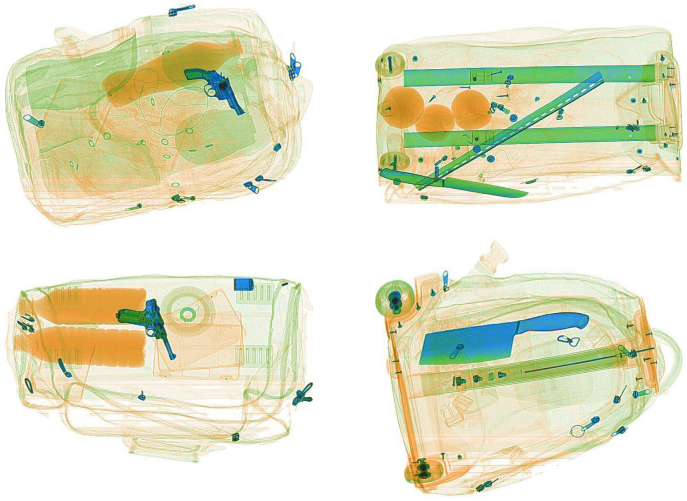
(a) (b)

Figure 4: Example images from the MedPix medical imaging dataset [90] belonging to classes a) 'Cholesterol granuloma' and b) 'Cleidocranial Dysostosis, Dysplasia'

the former one was randomly split into a training and a validation subset using a 80:20 ratio. Similarly, the labeled images of STL-10 were grouped into 10 predefined folds, each comprising 50 images from the total of 500 available annotated ones; then, one fold was utilized as the validation set and the remaining 9 formed the training one. Moreover, the SIXRay-100 dataset was resized to better fit the low-data regime, by constructing a subset of 300,000 images in the following way: all 8,929 positive images were retained and the negative ones were randomly sampled from the original negative images of the dataset; the formulated dataset was segmented following a 70-15-15 split ratio, resulting in a training/validation/test set of 210,000/45,000/45,000 images, respectively[10]. Overall, SIXRay-10 and the subsampled SIXRay-100 overlap in content, but the test images of SIXRay-10 are deliberately not included in the training or the validation set of SIXRay-100.

Unless otherwise specified, the following considerations hold in the conducted experiments: a) the training set of the pretraining dataset is employed for both SSL pretraining and supervised pretraining variants, b) in single-dataset experiments, the training set of the pretraining dataset is subsequently also utilized for supervised downstream finetuning, taking into account ground-truth annotation labels, c) each training session exploits an early stopping criterion based on validation loss, to automatically adjust the number of epochs, d) in all experiments except those falling under a single-dataset setup, the training set of each downstream dataset is utilized for

[10]In the sequel, this modified low-data variant is implied wherever SIXRay-100 is mentioned



(a) (b)

Figure 5: Example images from the SIXRay security imaging dataset [93] belonging to classes a) 'Gun' and b) 'Knife'

supervised downstream finetuning, taking into account ground-truth annotation labels, and e) the annotated test set of each downstream dataset is used for evaluation purposes, by inferring predictions using the trained DNN.

4. Experimental Results and Insights

Table 6: Top-5 accuracy in downstream classification for single-dataset SSL setup

Method	M_S^O(IN-100↓IN-100)	M_S^S(P-100↓P-100)
SimCLR (ResNet)	**73.5**%	69.8%
DINO (ViT)	72.1%	68.5%
MAE (ViT)	72.7%	69.1%
DeepClusterV2 (ResNet)	70.3%	66.8%
None/Random (ResNet)	70.6%	70.3%
None/Random (ViT)	71.2%	**70.9**%

This section presents the evaluation results for each of the three experimental sets described in Section 3, along with visual/qualitative and statistical analyses of the image representations learnt by the various SSL methods. Top-5 classification accuracy is the employed evaluation metric, excluding specific downstream tasks with a particularly low number of classes; top-1 accuracy is reported in such cases.

4.1. Main experimental results

This section presents in detail the comparative evaluation results obtained from the application of the experimental setting described in Section 3.1. In particular, Table 6 presents the

Table 7: Top-5 accuracy in downstream classification for transfer-learning SSL setup (top-1 accuracy reported for STL-10 dataset). The 'Sup-Large' and DINOv2 baselines have been pretrained on large-scale datasets.

Method	M_T^O(IN-100↓COCO)	M_T^O(IN-100↓STL-10)	M_T^S(P-100↓ADE)	M_T^S(P-100↓SUN DB)
SimCLR (ResNet)	70.8%	64.2%	49.4%	51.7%
DINO (ViT)	75.7%	67.1%	53.1%	55.2%
MAE (ViT)	74.4%	65.7%	51.1%	53.3%
DeepClusterV2 (ResNet)	67.0%	59.3%	46.2%	48.1%
Sup. (ResNet)	73.2%	67.8%	54.3%	53.5%
Sup. (ViT)	74.9%	69.7%	56.2%	55.8%
Sup.-Large (ResNet)	79.1%	73.6%	57.4%	59.4%
Sup.-Large (ViT)	**80.4**%	75.4%	58.9%	**61.2**%
DINOv2 (ViT)	79.7%	**76.1**%	**59.3**%	60.5%

Table 8: Top-5 accuracy in downstream classification for semi-supervised single-dataset setup

Method	M_S^O(IN-100↓IN-100) 5%	M_S^O(IN-100↓IN-100) 10%	M_S^S(P-100↓P-100) 5%	M_S^S(P-100↓P-100) 10%
SimCLR (ResNet)	59.7%	66.4%	55.2%	58.3%
DINO (ViT)	63.2%	**70.6**%	59.3%	62.6%
MAE (ViT)	61.3%	67.8%	57.0%	60.4%
DeepClusterV2 (ResNet)	58.1%	64.5%	51.5%	54.6%
None/Random (ResNet)	64.0%	69.3%	60.2%	63.5%
None/Random (ViT)	**64.7**%	69.8%	**61.3**%	**64.4**%

obtained results for the single-dataset setup for both object- and scene-centric cases, i.e. experiments M_S^O(IN-100↓IN-100) and M_S^S(P-100↓P-100), while also including as baselines conventional DNNs trained in a typical supervised setting with random initialization. Similarly, Table 7 illustrates the results for the transfer-learning setup, i.e., experiments M_T^O(IN-100↓COCO), M_T^O(IN-100↓STL-10), M_T^S(P-100↓ADE) and M_T^S(P-100↓SUN). Regarding the baselines, it must be noted that the 'Sup-Large' and DINOv2 models have been pretrained in large-scale datasets (ImageNet-1k/Places-205 for object-centric/scene-centric 'Sup-Large' baselines, respectively, and LVD-142M for DINOv2). Moreover, Table 8 provides the results of the single-dataset experiments, using the semi-supervised downstream finetuning protocol.

Single-dataset setup: From the results presented in Table 6, **SimCLR is shown to be the best performing SSL method**, despite being evaluated using a ResNet-50 architecture. In particular, in the object-centric setting (M_S^O(IN-100↓IN-100)) it achieves the overall best accuracy, while in the scene-centric case (M_S^S(P-100↓P-100)) is slightly outperformed by the baselines (trained from scratch with random parameter initialization). This result is potentially a side-effect of the fact that the pretraining dataset is more than double in size for the object-centric case (ImageNet-100), compared to that of the scene-centric one (Places-100). This finding suggests that there may exist a pretraining dataset size limit (within the range of 35k-100k images), below which SSL is indeed ineffective for representation learning. To verify this insight, an additional experiment has been performed in the M_S^O(IN-100↓IN-100) setting: ResNet-50 was separately trained via the best method (SimCLR pretraining + downstream finetuning) and via the random initialization baseline (direct training) on 3 randomly sampled subsets of ImageNet-100, containing 80%, 65%

Table 9: Top-5 accuracy in downstream classification for Noisy-ImageNet-100 robustness protocol evaluation

Method	Datasets		
	IN-A-100	IN-P-100	IN-C-100
SimCLR (ResNet)	70.3%	75.1%	68.3%
DINO (ViT)	73.0%	78.2%	72.6%
MAE (ViT)	74.6%	76.5%	70.4%
DeepClusterV2 (ResNet)	66.4%	68.7%	64.2%
Sup. (ResNet)	77.2%	78.0%	73.5%
Sup.-Large (ResNet)	81.8%	83.0%	76.5%
Sup. (ViT)	78.3%	78.5%	74.0%
Sup.-Large (ViT)	**82.5**%	84.2%	77.6%
DINOv2 (ViT)	81.8 %	**84.7** %	**78.3**%

and 50% of its training images, respectively (all classes are retained in all cases). The resulting SimCLR downstream accuracy falls from 73.5% (for the full ImageNet-100) to 70.2%, 67.7% and 63.9%, respectively. Correspondingly, the baseline's accuracy falls from 70.6% to 69.4%, 67.4% and 64.8%, respectively. This indeed validates the proposed hypothesis and localizes the hypothesized size limit between 50k and 65k training images.

An additional crucial finding is the observed surprising **dominance of SimCLR/ResNet-50 over DINO/ViT-L/16**. Given that in the examined single-dataset setup the absolute priority is not for computing representations that generalize well to different datasets, this may potentially be a side-effect of the more data-hungry nature of Vision Transformers compared to CNNs [95], which potentially limits their applicability in the low-data regime. Since DINO's powerful representation learning abilities are mostly evident in combination with ViT architectures, the aforementioned observation indicates that its value is limited in the low-data single-dataset setup.

Transfer-learning setup: Examining the results illustrated in Table 7, it can be seen that in the transfer-learning setup **DINO proves to be the best-performing SSL method**. Another critical observation though is that **baseline approaches (supervised pretraining and large-scale-pretrained DINOv2) outpass all SSL methods**. In fact, with the exception of the COCO dataset, even low-data supervised pretraining achieves higher accuracy than DINO. Nevertheless, the obtained results indicate that DINO models more general image representations compared to competing SSL approaches in the low-data regime.

Semi-supervised single-dataset setup: From the results presented in Table 8, it can be observed that **baseline approaches (trained from scratch with random parameter initialization) in general outperform SSL methods in the low-data semi-supervised single-dataset setting**, with the exception being DINO that performs the best in the object-centric scenario and when using 10% of the ground-truth class labels. This finding mostly confirms the results of the transfer-learning setting. In the semi-supervised evaluation setup, the ground-truth labels utilized for downstream finetuning contain a significant amount of noise and, thus, convey a more limited amount of useful knowledge about the task. In such a scenario, where there is no transfer but the inherently discriminant capability of learnt representations is important, **DINO outperforms all SSL competitors**. This reinforces the conclusion that, up to a degree, DINO retains its ability to learn more generally applicable embeddings, even under low-data pretraining conditions.

Table 10: Accuracy in downstream classification for Imbalanced-ImageNet-100 (top-5) and VTAB (top-1) robustness protocols evaluation

Method	Datasets			
	Imbalanced-IN-100	VTAB-Natural	VTAB-Specialized	VTAB-Structured
SimCLR (ResNet)	63.2%	62.5%	58.3%	60.1%
DINO (ViT)	79.1%	65.2%	61.8%	63.0%
MAE (ViT)	69.3%	66.7%	62.9%	64.5%
DeepClusterV2 (ResNet)	59.1%	60.3%	56.7%	58.2%
Sup. (ResNet)	78.2%	63.2%	62.8%	62.5%
Sup.-Large (ResNet)	85.3%	66.9%	63.4%	65.8%
Sup. (ViT)	80.0%	67.1%	**64.7**%	66.0%
Sup.-Large (ViT)	86.2%	68.2%	64.2%	66.5%
DINOv2 (ViT)	**86.8**%	**68.7**%	64.5%	**66.8**%

4.2. Robustness experimental results

This section discusses the outcomes obtained by the execution of the robustness evaluation protocols detailed in Section 3.2. In particular, Tables 9 and 10 illustrate downstream evaluation results for the Noisy-ImageNet-100, and the Imbalanced-ImageNet-100 and VTAB benchmarks, respectively. It must be highlighted that no downstream finetuning on Noisy-ImageNet-100 is performed, according to the relevant protocols. To this end, all SSL-pretrained models have been finetuned on regular ImageNet-100 and then evaluated on the three Noisy-ImageNet-100 datasets (in Table 9). On the other hand, the non-SSL baselines were trained from scratch on ImageNet-100 ('Sup.') or ImageNet-1k ('Sup.-Large') and then evaluated on Noisy-ImageNet-100. The DINOv2 baseline was pretrained on LVD-142M and then evaluated on Noisy-ImageNet-100, after appending to it a linear classifier finetuned on ImageNet-100. For the Imbalanced-ImageNet-100 benchmark (Table 10), the SSL methods and the basic supervised baselines were pretrained on ImageNet-100 and then finetuned on Imbalanced-ImageNet-100, before evaluation on the latter one's test set. The 'Sup.-Large' baselines were pretrained on ImageNet-1k, while the off-the-shelf DINOv2 baseline was pretrained on LVD-142M. Moreover, the robustness evaluation protocol for VTAB (Table 10) is similar to the one for Imbalanced-ImageNet-100.

From the results presented in Tables 9 and 10, it can be observed that **MAE and DINO (both on ViT architectures) are proven to be the most robust low-data SSL approaches**. However, **no low-data SSL method is able to surpass in performance DINOv2 or the supervised pretraining/training from scratch baselines**; not even the low-data baselines. The latter observation suggests that SSL pretraining in the low-data regime seems to be at a disadvantage with respect to robustness, but the reason is likely different among the three sets of experiments. In particular, the low-data supervised baselines are significantly less robust than their large-scale counterparts in the case of Noisy-ImageNet-100 and Imbalanced-ImageNet-100, but this is not true for VTAB. The particular characteristic of VTAB is that it uses a transfer learning setup with very few downstream finetuning examples, for evaluating how generally applicable different learnt representations are. This differs from both the transfer-learning setup, where there is a reasonably-sized downstream finetuning dataset available, and from the semi-supervised one, where the finetuning dataset simply has noisy ground-truth annotation and no transfer learning takes place. In this challenging scenario, MAE exhibits an advantage over DINO, under low-data pretraining conditions. Given that both methods are evaluated on ViT architectures, this finding

Table 11: Top-1 accuracy in downstream classification for domain-specific experiments

Method	Datasets		
	AID	ChestX-Det	SIXRay-10
SimCLR (ResNet)	58.1%	68.2%	73.3%
DINO (ViT)	65.7%	**76.4**%	**74.1**%
MAE (ViT)	63.5%	72.1%	69.7%
DeepClusterV2 (ResNet)	57.4%	70.6%	66.3%
Sup. (IN-100, ResNet)	60.2%	70.3%	68.4%
Sup.-Large (IN-1k, ResNet)	66.2%	72.5%	70.2%
Sup. (IN-100, ViT)	65.1%	73.6%	72.5%
Sup.-Large (IN-1k, ViT)	68.7%	74.5%	73.9%
DINOv2 (LVD-142M, ViT)	**69.4** %	74.8%	73.5%

suggests that MAE simply needs less pretraining data than DINO, in order to learn generally applicable features that are inherently semantically discriminant. It is likely that this behaviour is concealed in the transfer/semi-supervised-learning setup due to the larger finetuning dataset and the lack of transfer, respectively. Still, the low-data regime prevents even MAE from surpassing the supervised pretraining baselines.

4.3. Domain-specific experimental results

This section presents in detail comparative evaluation results in domain-specific application cases, as detailed in Section 3.3. In particular, Table 11 illustrates downstream classification results for the AID (remote sensing), ChestX-Det (medical imaging) and SIXRay-10 (security imaging) datasets. AID is composed of aerial-view RGB images, while ChestX-Det and SIXRay-10 comprise X-ray scans. It needs to be highlighted that, although the remote sensing and the medical imaging experiments correspond to typical transfer-learning scenarios, the security imaging setting exhibits particular key characteristics: a) the pretraining and downstream datasets are overlapping in content, and b) there is extreme imbalance in the size of the supported classes (e.g., the 'Negative' class dominates the dataset, while the 'Hammer' one contains only 60 images). To this end, the security imaging experiment practically lies in-between the single-dataset and transfer-learning scenarios, while the downstream task is highly challenging, due to the increased class imbalance. Regarding experimental details, the MLRSNet, MedPix and SIXRay-100 datasets were used for SSL pretraining in the corresponding domains. In all three domain-specific experiments, SSL-pretrained models are compared against baselines pretrained in a supervised manner on ImageNet-100/ImageNet-1k, with the exception of DINOv2 that was pretrained using the LVD-142M dataset.

From the results presented in Table 3.3, it can be observed that **in the remote sensing domain (AID dataset) the large-scale pretraining baselines in general perform better than the low-data SSL approaches**: pretrained DINOv2 and supervised ImageNet-1k pretraining lead to the highest performance. This suggests that, even though the remote sensing domain deviates from the one of natural images, the fact that it still remains in the RGB space leads the supervised pretraining methods to surpass the low-data SSL methods (of similar backbone architecture). However, **in-domain low-data DINO pretraining does outperform low-data supervised pretraining on ImageNet-100**. On the contrary, **in the medical and security imaging**

Table 12: Median silhouette coefficients for object-centric downstream scenarios

Method	M_S^O(IN-100↓IN-100)	M_T^O(IN-100↓COCO)	M_T^O(IN-100↓STL-10)
SimCLR (ResNet)	0.53	0.40	0.42
DINO (ViT)	0.52	0.45	0.49
MAE (ViT)	0.49	0.38	0.36
DeepClusterV2 (ResNet)	0.40	0.34	0.31
Sup. (ResNet)	0.51	0.48	0.46
Sup. (ViT)	0.54	0.50	0.50
Sup.-Large (ResNet)	0.51	0.50	0.48
Sup.-Large (ViT)	0.54	0.52	0.53
DINOv2 (ViT)	**0.57**	**0.54**	**0.55**

domains in-domain low-data DINO pretraining outperforms all other approaches, including the large-scale pretraining ones. This is considered to stem from the significant irrelevance of (conventional) RGB supervised pretraining under a shift to the target X-ray domain. Moreover, the aforementioned difference in performance supports the intuition that **in-domain low-data SSL pretraining is advantageous in highly domain-specific scenarios, compared to large-scale supervised pretraining schemes in conventional RGB settings**. It is notable to mention that, given the extreme class imbalance of SIXRay, SSL methods are able to learn generic features from all images, including those of the dominant 'Negative' class, even under a low-data pretraining setting. In contrast, large-scale supervised pretraining on ImageNet-1k does not lead to modeling better representations (most likely due to feature space discrepancy) and the potential of the subsequent supervised finetuning is limited by the class imbalance issue.

4.4. Explanation of learnt representations

In order to provide detailed insights and to shed more light on the actual knowledge patterns modeled by each SSL method, qualitative and statistical explanations are estimated for the SSL-learnt representations in the low-data regime. In particular, the silhouette coefficient [96], one of the most simple, robust and well-performing cluster validity indices [97], and the RELAX approach [98] ('REpresentation LeArning eXplainability'), a recently proposed method that generates attribution-based explanations of representations and models their uncertainty, are used.

With respect to the silhouette coefficient, when clustering N data points into K clusters, coefficient $S_i \in \mathbb{R}$, $S_i \in [-1, 1]$, of the i-th data point indicates the compactness of its assigned cluster and its separation from other clusters. The optimal clustering is considered the one where coefficient S_i approaches 1 for each data point i, while a negative silhouette coefficient implies problematic assignment of the i-th data point. The average over all data points ($S = \frac{1}{N}\sum_{i=1}^{N} S_i$) characterizes the quality of the overall dataset clustering. In order to estimate clustering results more robust to the presence of outliers, one can calculate the median S_m value instead. In order to compute S_m when evaluating inferred image representations on downstream image classification tasks, K can be set to the number of supported classes and the representation vectors of all test images belonging to the same ground-truth class can be considered as virtually having an identical cluster assignment. A high S_m value implies globular class representations that are both well separated and significantly compact in the latent space.

Table 13: Median silhouette coefficients for scene-centric downstream scenarios

Method	M_S^S(P-100↓P-100)	M_T^S(P-100↓ADE)	M_T^S(P-100↓SUN)
SimCLR (ResNet)	0.47	0.39	0.41
DINO (ViT)	0.46	0.45	0.42
MAE (ViT)	0.43	0.38	0.36
DeepClusterV2 (ResNet)	0.40	0.34	0.31
Sup. (ResNet)	0.48	0.42	0.41
Sup. (ViT)	**0.56**	0.49	0.47
Sup.-Large (ResNet)	0.48	0.43	0.44
Sup.-Large (ViT)	**0.56**	**0.59**	0.48
DINOv2 (ViT)	0.54	0.56	**0.49**

Table 14: Median silhouette coefficients for domain-specific downstream scenarios

Method	Datasets		
	AID	ChestX-Det	SIXRay-10
SimCLR (ResNet)	0.44	0.47	0.53
DINO (ViT)	0.58	**0.61**	0.54
MAE (ViT)	0.51	0.48	0.60
DeepClusterV2 (ResNet)	0.45	0.50	0.48
Sup. (ResNet)	0.50	0.54	0.55
Sup. (ViT)	0.53	0.58	0.59
Sup.-Large (ResNet)	0.52	0.57	0.56
Sup.-Large (ViT)	0.57	0.60	0.61
DINOv2 (ViT)	**0.59**	0.56	**0.63**

Regarding the RELAX-based explanation estimation, it measures similarities in the representation space between an input image and multiple/different masked out versions of it, in order to identify the image regions that are the most important for the feature extractor. The method generates visual/qualitative explanations of the learnt representations in the pixel space in the form of attribution maps, by considering only the output of the backbone DNN that extracts the image embeddings for the test images of any downstream dataset, i.e., without utilizing the classification head and its predictions. By definition, the importance score of each pixel can vary for different masked/semi-occluded versions of the same image; the variance in these scores for a given pixel is a proxy for the uncertainty of the obtained explanation at that pixel. Thus, if image regions that RELAX deems to be highly important for trained feature extractor F also tend to have low uncertainty, then F likely represents robustly their semantic content.

Tables 12, 13 and 14 illustrate the estimated median silhouette coefficient S_m for the considered object-centric, scene-centric and domain-specific downstream scenarios, respectively, from Sections 4.1 and 4.3. It must be noted that for the single-dataset setup in Tables 12 and 13, 'Sup.' and 'Sup.-Large' both imply a single setting, i.e., no pretraining/random initialization. In all cases, the median S_m for each model was calculated by obtaining its inferred representations for the test images of each downstream dataset, after finetuning. As can be seen, the obtained quantitative explanations are in general closely correlated with the respective downstream classification accuracy scores (reported in Sections 4.1 and 4.3), indicating that indeed better performing SSL

methods owe their discrimination capability to their ability to form more distinct, compact and well-separated class representations. This observation also holds for the supervised pretraining baselines and for the DINOv2 baseline, which is pretrained on the giant LVD-142M dataset and whose high test accuracy in the majority of transfer-learning setup cases is linked to its ability to generate very accurate/robust class representations with low intra-class variance and high inter-class separation in a zero-shot way; this is also in line with the foundation model status of off-the-shelf DINOv2 models. Additionally, it needs to be highlighted that in-domain low-data SSL pretraining in general leads to almost equally good or even better downstream image representations in domain-specific experimental setting, as also observed in the respective downstream classification tasks.

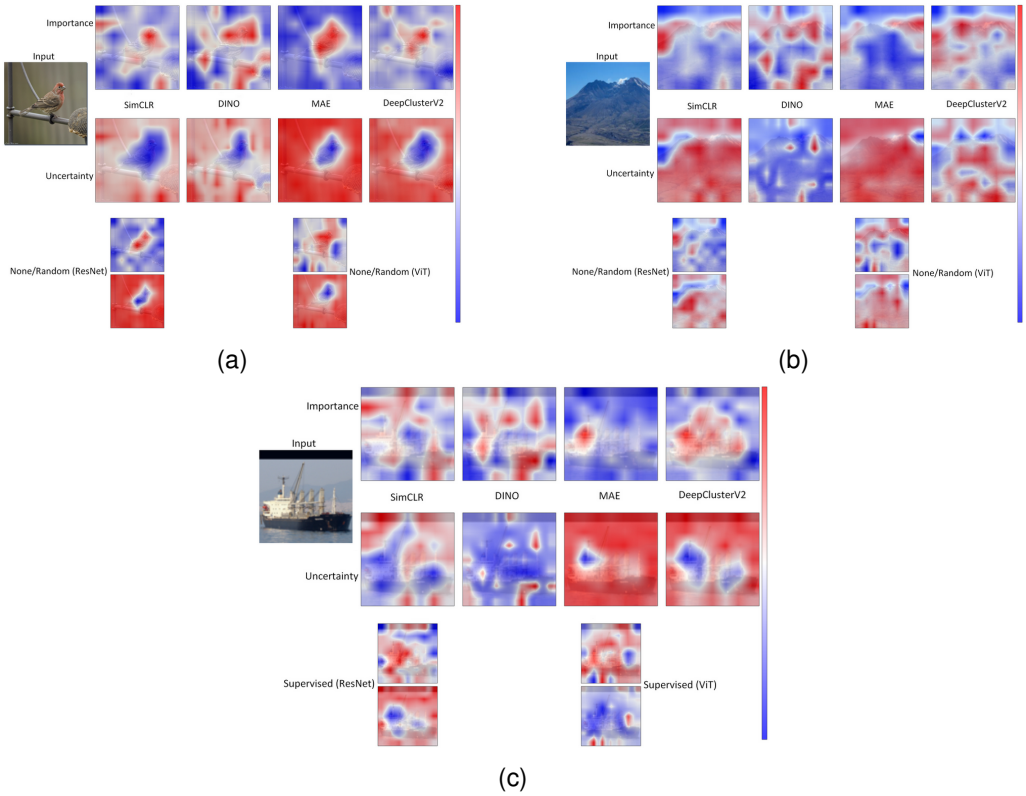


Figure 6: RELAX-based explanations for indicative test images from: (a) ImageNet-100 (single-dataset setup), (b) Places-100 (single-dataset setup), and (c) STL-10 (transfer-learning setup). Red/blue color indicates high/low value, respectively.

Figure 6 illustrates indicative RELAX-based visual explanations regarding the representations learnt by the various SSL pretraining methods, using exemplary test images from the ImageNet-100, Places-100 and STL-10 downstream datasets. Additionally, the outcome of low-data SSL pretraining is compared against a low-data supervised pretraining baseline (for the transfer-learning setup) or with downstream training from scratch with random initialization (for the single-dataset setup), both making use of ViT and ResNet architectures. In all cases, an importance and an uncertainty heatmap are depicted, where red/blue color indicates high/low value,

respectively. The provided indicative examples were selected after extensive visual inspection of hundreds of generated explanations. From the obtained results, it can be seen that SimCLR and DINO tend to diffuse their attention to multiple image regions, while MAE and DeepClusterV2 are keen to emphasize only on a few contiguous key areas. In the case of MAE, these areas are in general highly consistent with the visible foreground object, a fact that might underlie the method's remarkable VTAB performance under low-data pretraining. On the other hand, DINO tends in general to generate features with lower uncertainty across the image in the majority of cases, which may explain its superiority in most benchmarks.

5. Conclusions

In this paper, the issue of Self-Supervised Learning (SSL) in real-world application settings, i.e., cases where it is not always feasible to collect and/or to utilize very large (in the order of millions of instances) pretraining datasets, was examined. In this context, this work introduced a taxonomy of modern SSL methods for image analysis, accompanied by detailed explanations and insights regarding the main categories of approaches. Subsequently, a comprehensive comparative experimental study was performed in the so-called 'low-data regime', i.e., assuming a pretraining dataset with 50k - 300k images, aiming at shedding light on how the various SSL methods behave in such training scenarios. The performed comparative evaluation showcased that a pretraining dataset size limit exists, in the range 50k - 65k images, below which SSL is ineffective for representation learning. In the majority of cases, supervised pretraining and large-scale DINOv2 pretraining surpassed low-data SSL pretraining in terms of both downstream accuracy and robustness, with DINO retaining its ability to learn more generally applicable embeddings in the low-data regime. Very interestingly, in domain-specific experiments, in-domain low-data SSL pretraining outperformed all competitors and even large-scale pretraining on general datasets. This finding suggests that when extreme amounts of data are not available and/or the target downstream dataset is significantly different in nature compared to the large-scale datasets commonly used for pretraining (e.g., medical and security imaging against conventional natural RGB images), in-domain low-data SSL pretraining offers recognition advantages. At a finer level of detail, when comparing the performance of SSL methods only, MAE seems to need less pretraining data than DINO, in order to learn generally applicable features that are inherently semantically discriminant and consistent with the visible objects. On the other hand, given a reasonably-sized downstream dataset for finetuning, DINO typically surpasses MAE in the low-data setting, since it achieves higher downstream classification accuracy and generates the most well-separated and compact class representations. Moreover, the superiority of DINO persisted in domain-specific scenarios, when moving beyond the RGB space, where it outperformed all SSL competitors and even large-scale supervised pretraining. This indicates that low-data SSL pretraining not only works, but is also highly relevant in cases of domain-specific downstream tasks where generic (e.g., ImageNet-1k pretraining-derived) representations cannot generalize well.

Furthermore, suggestions for future research can be extrapolated from the above-mentioned insights. For example, given the finding that low-data SSL is indeed useful in domain-specific settings, general SSL methodologies should be adapted for exploiting domain-specific image properties (e.g., particularities of X-ray scans, such as pixel-level visual overlaps among classes), to obtain SSL pretraining schemes with higher data efficiency (i.e., overcoming the identified training dataset size limit) and better domain-specific features. Additionally, despite the superiority of self-distillation in current SSL literature (explained in this study by the lower RELAX

uncertainty of DINO), the generative MAE method dominates in low-data with respect to robustness. This is also elucidated in this study via RELAX visualizations, by observing that MAE features tend to emphasize the visible foreground object. New methods combining the best of both worlds in an adjustable manner (e.g., by setting a robustness-performance trade-off in pretraining) should become a priority for interested practitioners working with small datasets.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101073876 (Ceasefire). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.
[3] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks 28 (2015).
[4] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.
[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2017) 834–848.
[6] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156–3164.
[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database (2009).
[8] R. Wang, Y. Hao, L. Hu, J. Chen, M. Chen, D. Wu, Self-supervised learning with data-efficient supervised fine-tuning for crowd counting, IEEE Transactions on Multimedia 25 (2023) 1538–1546.
[9] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[10] V. Risojević, V. Stojnić, Do we still need Imagenet pre-training in remote sensing scene classification?, arXiv preprint arXiv:2111.03690 (2021).
[11] S. Chen, J.-H. Xue, J. Chang, J. Zhang, J. Yang, Q. Tian, SSL++: Improving self-supervised learning by mitigating the proxy task-specificity problem, IEEE Transactions on Image Processing 31 (2021) 1134–1148.
[12] S. Azizi, L. Culp, J. Freyberg, B. Mustafa, S. Baur, S. Kornblith, T. Chen, N. Tomasev, J. Mitrović, P. Strachan, et al., Robust and data-efficient generalization of self-supervised machine learning for diagnostic imaging, Nature Biomedical Engineering (2023) 1–24.
[13] H. Zhang, Y. Luo, L. Zhang, Y. Wu, M. Wang, Z. Shen, Considering three elements of aesthetics: Multi-task self-supervised feature learning for image style classification, Neurocomputing 520 (2023) 262–273.
[14] T. Konkle, G. A. Alvarez, A self-supervised domain-general learning framework for human ventral stream representation, Nature Communications 13 (1) (2022) 491.
[15] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin, et al., Self-supervised pretraining of visual features in the wild, arXiv preprint arXiv:2103.01988 (2021).
[16] Z. Ying, D. Cheng, C. Chen, X. Li, P. Zhu, Y. Luo, Y. Liang, Predicting stock market trends with self-supervised learning, Neurocomputing 568 (2024) 127033.
[17] J. D. M.-W. C. Kenton, L. K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019.
[18] M. C. Schiappa, Y. S. Rawat, M. Shah, Self-supervised learning for videos: A survey, ACM Computing Surveys 55 (13s) (2023) 1–37.
[19] Z. Chen, H. Wang, C. W. Chen, Self-supervised video representation learning by serial restoration with elastic complexity, IEEE Transactions on Multimedia (2023) 1–14.

[20] S. Liu, A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu, B. W. Schuller, Audio self-supervised learning: A survey, Patterns 3 (12) (2022).
[21] Y. Wu, J. Liu, M. Gong, P. Gong, X. Fan, A. K. Qin, Q. Miao, W. Ma, Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding, IEEE Transactions on Multimedia (2023) 1–13.
[22] Z. Tao, X. Liu, Y. Xia, X. Wang, L. Yang, X. Huang, T.-S. Chua, Self-supervised learning for multimedia recommendation, IEEE Transactions on Multimedia 25 (2023) 5107–5116.
[23] F. Luo, S. Chen, J. Chen, Z. Wu, Y.-G. Jiang, Self-supervised learning for semi-supervised temporal language grounding, IEEE Transactions on Multimedia (2022) 1–11.
[24] L. Huang, X. Fan, T. Xia, Y. Li, Y. Ding, SC2-Net: Self-supervised learning for multi-view complementarity representation and consistency fusion network, Neurocomputing 556 (2023) 126695.
[25] Y. Dai, Y. Li, B. Sun, Object and attribute recognition for product image with self-supervised learning, Neurocomputing 558 (2023) 126763.
[26] S. Gidaris, N. Komodakis, Unsupervised representation learning by predicting image rotations, arXiv preprint arXiv:1803.07728 (2018).
[27] C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.
[28] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2018, pp. 4724–4734.
[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the International Conference on Machine Learning (ICML), ACM, 2008, pp. 1096–1103.
[30] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).
[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2672–2680.
[32] S. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.
[33] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).
[34] M. U. Gutmann, A. Hyvärinen, Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics, Journal of Machine Learning Research 13 (Feb) (2012) 307–361.
[35] W. Deng, S. Gould, L. Zheng, What does rotation prediction tell us about classifier accuracy under varying testing environments?, in: Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2021.
[36] C. Zhuang, A. Zhai, D. Yamins, Local aggregation for unsupervised learning of visual embeddings, Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 6002–6012.
[37] R. Timofte, V. De Smet, L. Van Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1920–1927.
[38] H. Bao, L. Dong, S. Piao, F. Wei, BEiT: BERT pre-training of image transformers, arXiv preprint arXiv:2106.08254 (2021).
[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional Transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019.
[41] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, in: Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2021, pp. 8821–8831.
[42] A. Van Den Oord, O. Vinyals, et al., Neural discrete representation learning, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vol. 30, 2017.
[43] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16000–16009.
[44] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, N. Ballas, Self-supervised learning from images with a joint-embedding predictive architecture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 15619–15629.
[45] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9729–9738.
[46] Q. Men, E. S.L. Ho, H. P.H. Shum, H. Leung, Focalized contrastive view-invariant learning for self-supervised skeleton-based action recognition, Neurocomputing 537 (2023) 198–209.

[47] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the International Conference on Machine Learning (ICML), 2020, pp. 1597–1607.
[48] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2 (2006) 1735–1742.
[49] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, Proceedings of the Advances in Neural Information Processing Systems (NIPS) 27 (2014).
[50] Z. Wu, Y. Xiong, S. X. Yu, D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3733–3742.
[51] Y. Tian, D. Krishnan, P. Isola, Contrastive multiview coding, in: Proceedings of the European Conference on Computer Vision (ECCV), 2019, pp. 552–569.
[52] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9729–9738.
[53] I. Misra, L. van der Maaten, Self-supervised learning of pretext-invariant representations, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019) 6707–6717.
[54] C. Doersch, A. Zisserman, Multi-task self-supervised visual learning, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2051–2060.
[55] M. Ye, X. Zhang, P. C. Yuen, S.-F. Chang, Unsupervised embedding learning via invariant and spreading instance feature, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6210–6219.
[56] X. Ji, J. F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9865–9874.
[57] J. Zbontar, L. Jing, I. Misra, Y. LeCun, et al., Barlow Twins: Self-supervised learning via redundancy reduction, arXiv preprint arXiv:2103.03230 (2021).
[58] H. B. Barlow, et al., Possible principles underlying the transformation of sensory messages, Sensory Communication 1 (01) (1961) 217–233.
[59] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap Your Own Latent: a new approach to self-supervised learning, Proceedings of the Advances in Neural Information Processing Systems (NIPS) 33 (2020) 21271–21284.
[60] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[61] X. Chen, K. He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15750–15758.
[62] C. Zhang, K. Zhang, C.-D. Yoo, I.-S. Kweon, How does SimSiam avoid collapse without negative samples? Towards a unified understanding of progress in SSL, in: Proceedings of the International Conference on Learning Representations (ICLR), 2022.
[63] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised Vision Transformers, arXiv preprint arXiv:2104.14294 (2021).
[64] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2021, pp. 8748–8763.
[65] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., DINOv2: Learning robust visual features without supervision, arXiv preprint arXiv:2304.07193 (2023).
[66] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, T. Kong, iBOT: Image bert pre-training with online tokenizer, arXiv preprint arXiv:2111.07832 (2021).
[67] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vol. 26, 2013.
[68] Y. Ruan, S. Singh, W. Morningstar, A. A. Alemi, S. Ioffe, I. Fischer, J. V. Dillon, Weighted ensemble self-supervised learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2023.
[69] A. Sablayrolles, M. Douze, C. Schmid, H. Jégou, Spreading vectors for similarity search, in: Proceedings of the International Conference on Learning Representations (ICLR), 2018.
[70] X. Zhou, N. L. Zhang, Deep clustering with features from self-supervised pretraining, arXiv preprint arXiv:2207.13364 (2022).
[71] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.

[72] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2020.
[73] Y. M. Asano, C. Rupprecht, A. Vedaldi, Self-labelling via simultaneous clustering and representation learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2020.
[74] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by jointly learning to align and translate, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015.
[75] P. C. Chhipa, J. R. Holmgren, K. De, R. Saini, M. Liwicki, Can self-supervised representation learning methodswithstand distribution shifts and corruptions?, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 4467–4476.
[76] B. Zhao, X. Xiao, G. Gan, B. Zhang, S.-T. Xia, Maintaining discrimination and fairness in class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13208–13217.
[77] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).
[78] X. He, Y. Zhang, Quantifying and mitigating privacy risks of contrastive learning, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 845–863.
[79] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2014, pp. 740–755.
[80] A. Coates, A. Ng, H. Lee, An analysis of single layer networks in unsupervised feature learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, 2011.
[81] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ADE20k dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 633–641.
[82] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, SUN Database: Large-scale scene recognition from abbey to zoo, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.
[83] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for Deep Neural Networks, in: Proceedings of the International Conference on Machine Learning Workshops (ICMLW), Vol. 3, 2013, p. 896.
[84] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4L: Self-supervised semi-supervised learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1476–1485.
[85] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, D. Song, Natural adversarial examples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
[86] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: Proceedings of the International Conference on Learning Representations (ICLR), 2019.
[87] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S. X. Yu, Large-scale long-tailed recognition in an open world, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[88] X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, et al., The Visual Task Adaptation Benchmark (2019).
[89] X. Qi, P. Zhu, Y. Wang, L. Zhang, J. Peng, M. Wu, J. Chen, X. Zhao, N. Zang, P. T. Mathiopoulos, MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding, ISPRS Journal of Photogrammetry and Remote Sensing 169 (2020) 337–350.
[90] N. L. of Medicine, Medpix medical image database.
URL `https://medpix.nlm.nih.gov/home`
[91] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, X. Lu, AID: A benchmark data set for performance evaluation of aerial scene classification, IEEE Transactions on Geoscience and Remote Sensing 55 (7) (2017) 3965–3981.
[92] J. Lian, J. Liu, S. Zhang, K. Gao, X. Liu, D. Zhang, Y. Yu, A structure-aware relation network for thoracic diseases detection and segmentation, IEEE Transactions on Medical Imaging 40 (8) (2021) 2042–2052.
[93] C. Miao, L. Xie, F. Wan, c. Su, H. Liu, j. Jiao, Q. Ye, SIXray: A large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[94] J. Liu, J. Lian, Y. Yu, ChestX-Det10: chest X-ray dataset on detection of thoracic abnormalities, arXiv preprint arXiv:2006.10550 (2020).
[95] Y. Xu, Q. Zhang, J. Zhang, D. Tao, ViTAE: Vision Transformer advanced by exploring intrinsic inductive bias 34 (2021) 28522–28535.
[96] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[97] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognition 46 (1) (2013) 243–256.
[98] K. K. Wickstrøm, D. J. Trosten, S. Løkse, A. Boubekki, K. ø. Mikalsen, M. C. Kampffmeyer, R. Jenssen, RELAX: Representation learning explainability, International Journal of Computer Vision 131 (6) (2023) 1584–1610.