# Global-Local Self-Distillation for Visual Representation Learning

Tim Lebailly
KU Leuven
tim.lebailly@esat.kuleuven.be

Tinne Tuytelaars
KU Leuven
tinne.tuytelaars@esat.kuleuven.be

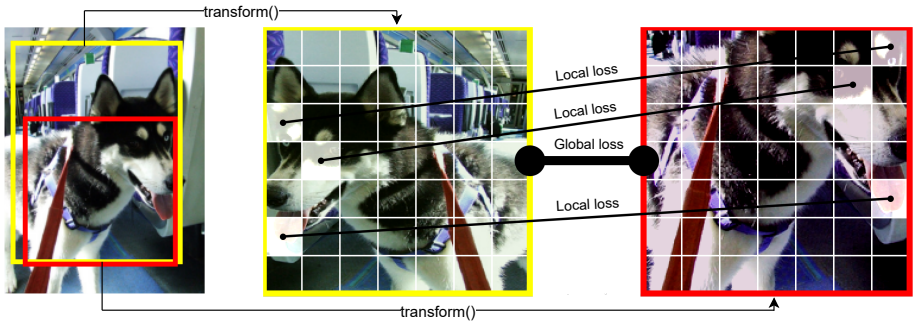


Figure 1. **Visual illustration of our global-local self-distillation framework with a loss for each component.** The global loss maximizes the similarity of both global-representations while the local losses maximize the similarity between pairs of local-representations.

## Abstract

*The downstream accuracy of self-supervised methods is tightly linked to the proxy task solved during training and the quality of the gradients extracted from it. Richer and more meaningful gradients updates are key to allow self-supervised methods to learn better and in a more efficient manner. In a typical self-distillation framework, the representation of two augmented images are enforced to be coherent at the global level. Nonetheless, incorporating local cues in the proxy task can be beneficial and improve the model accuracy on downstream tasks. This leads to a dual objective in which, on the one hand, coherence between global-representations is enforced and on the other, coherence between local-representations is enforced. Unfortunately, an exact correspondence mapping between two sets of local-representations does not exist making the task of matching local-representations from one augmentation to another non-trivial. We propose to leverage the spatial information in the input images to obtain geometric matchings and compare this geometric approach against previous methods based on similarity matchings. Our study shows that not only 1) geometric matchings perform better than similarity based matchings in low-data regimes but also 2) that similarity based matchings are highly hurtful in low-data regimes compared to the vanilla baseline without local self-distillation. The code is available at https://github.com/tileb1/global-local-self-distillation.*

## 1. Introduction

The last few years have seen a lot a progress in self-supervised learning due to its ability to make use of large unlabeled datasets. The trend has been to train ever larger networks on ever larger datasets. However, this is very costly both in terms of compute resources and environmental impact. This also impedes research on this topic to all but a few large labs with the required infrastructure. Recent works (*e.g.* [6, 8, 22]) train large models on distributed computing clusters using hundreds of GPUs for a single run. The cost for such clusters easily exceeds millions of dollars and power consumption easily surpasses the 10s of kilowatts. It is therefore crucial to make the learning as efficient as possible by leveraging as much self-supervisory signal as possible from the input images. One way to achieve this is to incorporate local cues in the self-supervised training.

Recently, transformer backbones using the self-attention mechanism [42] have been gaining more popularity in the computer vision field. Monolithic vision transformers [17] have induced a wave of work on multi-stage vision transformers [31, 51, 44, 38] which do not process patch tokens at a single resolution (*e.g.* 16x16) but at multiple resolutions via patch merging. These architectures encode an input image into a representation which is coarser-grained than the pixel-level, yet preserves the spatial structure of the input image. Given the highly complex mapping from input image to output representation, local regularization is even more motivated.

Typical self-distillation frameworks aim to maximize the

similarity between the global representation of two augmented crops coming from the same input image. The idea is to generate augmentations which carry the same semantic meaning (*e.g*. image of dog) but contain different low-level information (*e.g*. different lighting, background, scale *etc*.). The backbone is then trained to output a representation of both augmented images which are coherent with each other. Under this setup, the network should learn to retain semantic content from the input image while discarding redundancy and noise. To incorporate additional self-supervisory signal in the training, one can devise a similar loss which acts on local-representations instead of the global-representation. This leads to a dual objective where two terms are optimized (local and global) as shown in Figure 1. As opposed to the global-representation, a single augmented image does not lead to a single local-representation but to a set of local-representations. This makes the expression for the local loss non-trivial. A question naturally arise: *How should one generate pairs of local-representations* i.e. *which local-representation from one augmented image should be matched to which local-representation from the other augmented image?*

Ideally, we would like to match local-representations which share the same semantic content. In a self-supervised setup, we don't have access to such oracle and should rely purely on data-driven approximations. Li *et al*. [28] propose to use a matching function which is based on the similarity of local-representations. The assumption is that similar local-representations should be semantically close. In practice, this does not always hold, especially when augmented images don't overlap much. On the other hand, we propose a geometric matching function. The assumption is that representations originating from close-by regions of the input image are semantically close. One can also easily threshold the matching distance to avoid the above mentioned problem of little overlap between augmentations. We propose a study comparing both approaches and summarize our contribution as follows:

- To our knowledge, we are the first to introduce local self-distillation for the features of multi-stage vision transformers based on geometry.
- We study what is the best way to incorporate local self-distillation including a similarity based proxy task as in SOTA method [28] and our geometry based self-supervised proxy task:
  - We show that a similarity based self-distillation proxy task can be hurtful in low-data regimes and performs much worse than the vanilla setup without additional local loss. The geometry based self-distillation proxy task is more robust and improves the vanilla setup in all data-regimes.
  - We show comparable performance between both approaches in high-data regimes (*e.g*. ImageNet-1k [14]).
  - Finally, we show that geometry based matchings lead to a processing of the input image which better preserves the spatial structure of images and show empirical evidence of local-representation mode collapse when using a similarity based matching function.

## 2. Related works

**Self-supervised learning** Early self-supervised methods for representation learning make use of pretext tasks. As pretext tasks, Noroozi et al. [35] solve a jigsaw puzzle and Gidaris et al. [18] predict which rotation was applied on an input image. Other approaches include predicting patch context [16, 34], inpainting patches [36], predicting noise [3], *etc*.

More recently, contrastive learning methods have been the most popular. Contrastive learning is a scheme for metric learning which leverages distinctiveness and similarities between inputs. Chen *et al*. [7] propose a simple framework for contrastive learning of visual representations (SimCLR) which has kickstarted a lot of research in this direction [10, 23, 8, 9, 11, 46, 49, 48]. Grill *et al*. propose a framework called BYOL [20] where negative samples in the contrastive loss are not needed to avoid collapse by using a simple mean squared error loss (MSE) between the output representations of two branches. Caron *et al*. [6] (DINO) extend this framework by introducing visual transformers as the backbone and by viewing this learning paradigm as self-distillation. We were inspired by DINO, yet observed that they only use the global representations, leaving valuable cues at the local scale unexploited. Note that self-supervised methods require large quantities of data to get great results. Few works focus explicitly on self-supervised pretraining on small-scale datasets [13, 39, 30].

**Dense contrastive learning** The above-mentioned methods focus on learning a visual representation at the image-level. Some works take a different approach and aim to learn a representation at the pixel-level, which is useful for dense tasks like segmentation. Pinheiro *et al*. [37] generate positive pixel pairs corresponding to the same location from an input image. Xie *et al*. [47] use a similar loss as well as an additional pixel-to-propagation consistency task improving the downstream task accuracy. Notable works along the same lines include [15, 43].

**Dense self-distillation** Li *et al*. (EsViT [28]) focus on classification downstream tasks and propose a self-distillation task leveraging the local features of a multi-stage visual transformer (rather than pixel-representations) based on their similarities. EsViT is thus not fully dense but does share similarities with these approaches. We argue that explicitly using the spatial information from the original input images, as we do, provides stronger feedback than matching local representations purely based on similarity, as they do (especially in low-data regimes).

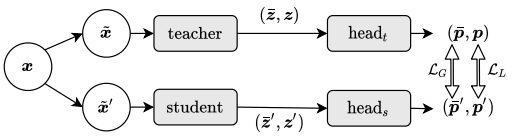


Figure 2. **High-level overview of a global-local self-distillation framework.** The augmentations $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$ are fed respectively through the teacher and student backbone resulting in both global- ($\bar{\boldsymbol{z}}$) and dense-representations ($\boldsymbol{z}$). Then, these representations are respectively fed through a teacher- and student-head to output probability mass functions with which $\mathcal{L}_G$ and $\mathcal{L}_L$ are minimized (more details in Sec. 3.4).

**Vision transformers** Visual transformers (ViT) have been proposed by Dosovitskiy *et al*. [17] as an alternative to the more common CNN backbone (*e.g*. ResNets [25]). Input images are patchified, then each patch is flattened and fed to a linear layer whose output serves as tokens in a traditional NLP transformer backbone [42]. ViTs allow more complex mappings between input and output as the architecture is not translation invariant. In the absence of a supervisory training signal, ViTs are adequate to model complex dependencies and outperform ResNets, as shown by DINO [6]. Some self-supervised works mimic the *masked word prediction* task in NLP with a *masked image modeling* task [2, 21]. More recently, multi-stage architectures have been proposed where patches are not processed at a single resolution but at multiple resolutions via patch merging. Liu *et al*. [31] propose such an architecture where they also process tokens in windows of restricted size to lower the compute requirements. Other notable works along the same lines include [50, 51, 44, 38, 40, 12].

# 3. Methodology

This section starts by reviewing necessary representation terminology (Sec. 3.1) and the augmentation pipeline (Sec. 3.2). Then, the training scheme is discussed (Sec. 3.3). Finally, we review the self-supervised loss needed to incorporate additional local cues. A high-level overview sketch can be found in Figure 2.

## 3.1. Global- versus local-representations

Most previous works (*e.g*. [6, 7, 20, 10, 23]) use a self-supervised loss based only on the *global-representation* of the augmentations. With this term, we refer to either the output of the backbone network after a global average pooling or the `[CLS]` token in the case of vision transformer backbones. In both cases, the global-representation is a vector $\bar{\boldsymbol{z}} \in \mathbb{R}^d$ where $d$ is the size of the latent space. On the other hand, we use the term *dense-representation* to refer to a representation in which spatial structure of the input image is explicitly modeled. Such dense-representations usually take the form of a third-order tensor $\in \mathbb{R}^{H \times W \times d}$ or $\in \mathbb{R}^{HW \times d}$, where $H$ and $W$ are respectively the height and width of the input image or a downscaled version thereof. Examples of such dense-representation include the output feature map of a CNN or an ordered sequence of tokens from a visual transformer (excluding the `[CLS]` token). Finally, we use the term *local-representation* to refer to a 1D slice $\boldsymbol{z}_k \in \mathbb{R}^d$ of a dense-representation associated to a certain local position $k$ (of the $K = HW$ possible locations).

## 3.2. Data augmentation pipeline as a composition of geometric and photometric transforms

There are 3 main components in self-supervised frameworks: 1) a data-augmentation pipeline, 2) a backbone and 3) a self-supervised proxy task. Data augmentation pipelines play a crucial role in self-supervised learning settings since they produce the necessary augmented samples needed to enforce a self-supervised loss. Previous work [41] shows empirical findings on how the downstream task accuracy is linked to parameters of the pipeline. In our work, we assume the data augmentation pipeline as given and use the same one as [6] and [28]. This pipeline is the fruit of empirical testing from many previous works [7, 26, 1, 5, 6].

The data augmentation pipeline is a long composition of multiple transforms, including both geometric transforms and photometric transforms. Geometric transforms include `CROP`, `RESIZE` and `HORIZONTAL_FLIP` while photometric transforms include `COLOR_JITTER`, `SOLARIZE`, `GAUSSIAN_BLUR` and `GRAYSCALE`. We denote the composition of all geometric transforms by $\mathbf{G}$ and the composition of all photometric transforms by $\mathbf{P}$.

Geometric and photometric transforms are respectively parametrized by vectors $\boldsymbol{w}_{geo}$ and $\boldsymbol{w}_{pho}$ with $\boldsymbol{w}_{geo} = [ul_x, ul_y, lr_x, lr_y, h, w, f]$. The first 4 elements represent the location of the crop (in the form of upper left and lower right coordinates) to be taken w.r.t. the original image. $h$ and $w$ refer to the resized shape of the crop while $f$ is a binary variable indicating whether the crop is flipped horizontally or not. The actual form of $\boldsymbol{w}_{pho}$ is not relevant for this analysis. The data augmentation pipeline is characterized by a distribution $\mathcal{D}_{aug}$ from which all parameters are sampled, *i.e*. $\boldsymbol{w} \sim \mathcal{D}_{aug}$ with $\boldsymbol{w} = [\boldsymbol{w}_{geo}, \boldsymbol{w}_{pho}]$.

Given a single input image $\boldsymbol{x}$ and a sampled augmentation parameter vector $\boldsymbol{w}$, we generate an augmentation $\tilde{\boldsymbol{x}}$ as follows

$$\tilde{\boldsymbol{x}} = \mathbf{P}\left(\mathbf{G}(\boldsymbol{x}, \boldsymbol{w}_{geo}), \boldsymbol{w}_{pho}\right) \tag{1}$$

## 3.3. Self-distillation

Before we can dive into the explicit expression of the loss, we first review the self-supervised training scheme which we use in our work. Within self-supervised learning methods, contrastive ones are the most popular and have

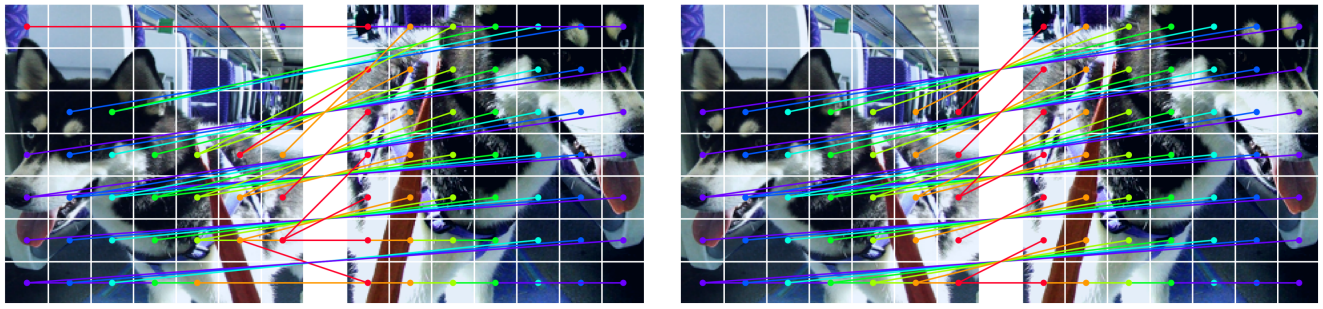

Figure 3. **Visualization of the matchings enforced during training for $\mathcal{L}_L^{sim}$ (left) and $\mathcal{L}_L^{geo}$ (right).** The similarity matchings depend on the state of the backbone during the training phase as they are a function of the local-representations (here shown after 300 epochs of training on ImageNet-1k). The geometric matchings on the other hand are not a function of the local-representations and hence are fixed throughout the training. Colors are used only to better distinguish different matchings.

been used mainly due to their ability to avoid mode collapse in an explicit and simple manner. However, [10, 6, 20] show that negative samples are not needed to learn representations while avoiding collapse by including some of the following tricks: use asymmetric predictors, use stop-gradients in one branch, have one branch reflect a low-passed (*e.g.* with an exponential moving average) version of the other, run more gradient descent steps on one branch, use some kind of normalization on the representation, *etc.* However, [6] are the first to view "contrastive learning without negative pairs" as a form of self-supervised knowledge distillation. Knowledge distillation is a learning paradigm where a student network learns to imitate the output of a teacher network. In a self-supervised setting, both the student $g_s$ and the teacher network $g_t$, parametrized respectively by $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_t$, are initialized to the same random $\boldsymbol{\theta}_{init}$. The student is then optimized such that its output matches the one of the teacher w.r.t. a particular loss function. The teacher network is updated at each epoch to reflect an exponential moving average of the student's weights, *i.e.* $\boldsymbol{\theta}_t \leftarrow \lambda\boldsymbol{\theta}_t + (1-\lambda)\boldsymbol{\theta}_s$.

### 3.4. Self-supervised loss

The global-representation loss used in our study is the same as proposed by DINO [6] and is explained in the following subsection using notations similar to EsViT [28]. Given a backbone $f$ and an augmentation $\tilde{\boldsymbol{x}}$, we obtain both the global- ($\bar{\boldsymbol{z}}$) and the dense-representation ($\boldsymbol{z}$) in a single forward pass, *i.e.* $(\bar{\boldsymbol{z}}, \boldsymbol{z}) = f(\tilde{\boldsymbol{x}})$. By abuse of notations, we will use $\bar{\boldsymbol{z}} = \bar{f}(\tilde{\boldsymbol{x}})$ and $\boldsymbol{z} = f(\tilde{\boldsymbol{x}})$.

Given a student backbone $f_s$ and teacher backbone $f_t$ as well as a set $\mathcal{V} = \{\tilde{\boldsymbol{x}}^1, \tilde{\boldsymbol{x}}^2, \tilde{\boldsymbol{x}}^3, \cdots\}$ containing $N = |\mathcal{V}|$ augmented views of the same input image, a single forward pass of all augmentations in both networks results in:

1. two sets of global-representations $\bar{\mathcal{Z}}_s = \{\bar{f}_s(\tilde{\boldsymbol{x}}) : \tilde{\boldsymbol{x}} \in \mathcal{V}\}$ and $\bar{\mathcal{Z}}_t = \{\bar{f}_t(\tilde{\boldsymbol{x}}) : \tilde{\boldsymbol{x}} \in \mathcal{V}\}$
2. two sets of local-representations $\mathcal{Z}_s = \{f_s(\tilde{\boldsymbol{x}}) : \tilde{\boldsymbol{x}} \in \mathcal{V}\}$ and $\mathcal{Z}_t = \{f_t(\tilde{\boldsymbol{x}}) : \tilde{\boldsymbol{x}} \in \mathcal{V}\}$

#### 3.4.1 Global-representation loss

The global-representations $\bar{\boldsymbol{z}}$ are then mapped to a discrete probability mass function $\bar{\boldsymbol{p}}$ of dimension $I$ using an MLP-head $\bar{h}$, *i.e.* $\bar{\boldsymbol{p}} = \bar{h}(\bar{\boldsymbol{z}})$. For each pair of global-representations coming from the student and the teacher, we use $\bar{h}$ to map them to a probability mass function and minimize their cross-entropy, more explicitly[1]:

$$\mathcal{L}_G = \frac{1}{N(N-1)} \sum_{\bar{\boldsymbol{z}} \in \bar{\mathcal{Z}}_t} \sum_{\substack{\bar{\boldsymbol{z}}' \in \bar{\mathcal{Z}}_s \\ \tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'}} H\left(\bar{h}(\bar{\boldsymbol{z}}), \bar{h}(\bar{\boldsymbol{z}}')\right) \tag{2}$$

with

$$H(p, q) = -\sum_{i \in \mathcal{I}} p(i) \log q(i) \tag{3}$$

where $\mathcal{I}$ is the support of the distributions $p$ and $q$, in our case $\mathcal{I} = [I] = \{1, 2, \cdots, I\}$. The summation constraint $\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'$ of the inner sum refers to the fact that we do not have a term $H\left(\bar{h}(\bar{\boldsymbol{z}}), \bar{h}(\bar{\boldsymbol{z}}')\right)$ where $\bar{\boldsymbol{z}}$ and $\bar{\boldsymbol{z}}'$ are global-representations corresponding to the same augmentation.

#### 3.4.2 Similarity based local-representation loss

Similar to the global-representation loss, each local-representation $\boldsymbol{z}_k, \forall k \in [K]$ is mapped to a probability mass function $\boldsymbol{p}_k$ using another MLP head $h$, *i.e.* $\boldsymbol{p}_k = h(\boldsymbol{z}_k)$. A local-representation $\boldsymbol{z}_k$ from an augmented $\tilde{\boldsymbol{x}}$ is matched to the best corresponding $\boldsymbol{z}'_{k^\star}$ from another augmented image $\tilde{\boldsymbol{x}}'$. Here, the best corresponding local-representation is selected as the local-representation $\boldsymbol{z}'_{k^\star}$ in the other augmentation $\tilde{\boldsymbol{x}}'$ which has the highest similarity with $\boldsymbol{z}_k$ from the first augmentation $\tilde{\boldsymbol{x}}$ as done in EsViT [28]. This is shown on the left of Figure 3. The cross-entropy between the probability outputs of matching local-representations is then minimized for all matchings and all

[1]We choose to leave Eq. (2) in a more readable format avoiding the multi-crop strategy [5] which we do use in practice. The full expression for the loss with the multi-crop strategy can be found in the appendix.

pairs of augmentations $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. Given two dense representations $\boldsymbol{z}$ and $\boldsymbol{z}'$:

$$L_L^{sim}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{K} \sum_{k \in [K]} H\left(h(\boldsymbol{z}_k), h(\boldsymbol{z}'_{k^\star})\right) \tag{4}$$

where $k^\star = \arg\max_j \frac{\boldsymbol{z}_k^\top \boldsymbol{z}'_j}{\|\boldsymbol{z}_k\| \|\boldsymbol{z}'_j\|}$. Averaging over all pairs of dense representations, the total local similarity based self-supervised objective becomes

$$\mathcal{L}_L^{sim} = \frac{1}{N(N-1)} \sum_{\boldsymbol{z} \in \mathcal{Z}_t} \sum_{\substack{\boldsymbol{z}' \in \mathcal{Z}_s \\ \tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'}} L_L^{sim}(\boldsymbol{z}, \boldsymbol{z}') \tag{5}$$

#### 3.4.3 Geometric local-representation loss

Along the set of augmented views $\mathcal{V}$, we also dispose over a set $\mathcal{W}_{geo} = \{\boldsymbol{w}_{geo}^1, \boldsymbol{w}_{geo}^2, \boldsymbol{w}_{geo}^3, \cdots\}$ of vectors $\boldsymbol{w}_{geo}$ which describe the geometric transforms $\boldsymbol{x}$ has undergone to generate $\tilde{\boldsymbol{x}}$. Using this set, we generate another set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \boldsymbol{e}^3, \cdots\}$ where each element $\boldsymbol{e} \in \mathbb{R}^{H \times W \times 2}$ is an object of the same spatial dimension as its associated dense-representation $\boldsymbol{z} \in \mathbb{R}^{H \times W \times d}$. $\boldsymbol{e}_k \in \mathbb{R}^2$ (a slice of $\boldsymbol{e}$) encodes the $(x, y)$ coordinates of the center point of the patch associated to $\boldsymbol{z}_k$ for every $K = HW$ locations in $\boldsymbol{z}$ w.r.t. the original input image grid $\boldsymbol{x}$ (not the augmentation $\tilde{\boldsymbol{x}}$). Note that $\boldsymbol{e}$ is a "positional encoding", though it should not be confused with the positional encoding used in transformers to remove the permutation invariance of the tokens.

Here, the best corresponding $\boldsymbol{z}'_{k^\star}$ from another augmented image $\tilde{\boldsymbol{x}}'$ is selected based on how close they are w.r.t. to the original image grid $\boldsymbol{x}$. The cross-entropy between the probability outputs of matching local-representations is then minimized for most matchings and all pairs of augmentations $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. As opposed to the similarity based local loss, we do not average over all pairs of local-representations. A matching $\boldsymbol{z}_k \leftrightarrow \boldsymbol{z}'_{k^\star}$ obtained via $k^\star = \arg\min_j \left\|\boldsymbol{e}_k - \boldsymbol{e}'_j\right\|^2$ might be very bad (in terms of matching distance $= d(k) = \min_j \left\|\boldsymbol{e}_k - \boldsymbol{e}'_j\right\|$) when there is no overlap between $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. Therefore, we restrict our averaging over the set of matchings $\boldsymbol{z}_k \leftrightarrow \boldsymbol{z}'_{k^\star}$ which have a low matching distance *i.e.* the $\boldsymbol{z}_k$ and $\boldsymbol{z}'_{k^\star}$ lie on the region of overlap between $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. The matching distance threshold $s$ is set to half of the maximum between 1) the length of the diagonal of a local representation $\boldsymbol{z}_k$ corresponding to augmentation $\tilde{\boldsymbol{x}}$ and 2) the length of the diagonal of a local representation $\boldsymbol{z}'_k$ corresponding to augmentation $\tilde{\boldsymbol{x}}'$. By the length of the diagonal of a local representation $\boldsymbol{z}_k$, we refer to the Euclidean distance between $\boldsymbol{e}_k$ and an adjacent diagonal $\boldsymbol{e}_{k^*}$ where $\boldsymbol{e}$ is the positional encoding described in the first paragraph of Sec. 3.4.3. s is set to this value because if $d(k) > s$, either $\boldsymbol{z}_k$ or $\boldsymbol{z}'_{k^\star}$ falls outside the region of overlap (if any) between augmentations $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. Taking the above into consideration and given two dense representations $\boldsymbol{z}$ and $\boldsymbol{z}'$:

$$L_L^{geo}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}_{\{d(k) < s\}} H\left(h(\boldsymbol{z}_k), h(\boldsymbol{z}'_{k^\star})\right) \tag{6}$$

where $k^\star = \arg\min_j \left\|\boldsymbol{e}_k - \boldsymbol{e}'_j\right\|^2$. $\boldsymbol{e}$ and $\boldsymbol{e}'$ are the positional encodings associated respectively to $\boldsymbol{z}$ and $\boldsymbol{z}'$. $d(k) = \min_j \left\|\boldsymbol{e}_k - \boldsymbol{e}'_j\right\|$ is the matching distance of $\boldsymbol{z}_k \leftrightarrow \boldsymbol{z}'_{k^\star}$, $s$ is the dynamically set distance threshold and $\mathbb{1}_{\{\text{condition}\}}$ is the indicator function. Averaging over all pairs of dense representations, the total local self-supervised objective based on geometry becomes

$$\mathcal{L}_L^{geo} = \frac{1}{N(N-1)} \sum_{\boldsymbol{z} \in \mathcal{Z}_t} \sum_{\substack{\boldsymbol{z}' \in \mathcal{Z}_s \\ \tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'}} L_L^{geo}(\boldsymbol{z}, \boldsymbol{z}') \tag{7}$$

Table 1. **Overview of three different settings based on the local loss used.** $\mathcal{L}_G$ and $\mathcal{L}_L^{sim/geo}$ refer respectively to the global- and local-representation loss. The matching type column refers to the matching type in the local-representation loss.

| Setting | Backbone | Global loss | Local loss | Matching type | Proposed in |
|---|---|---|---|---|---|
| `Vanilla` | Swin-T/7 | $\mathcal{L}_G$ | ✗ | ✗ | DINO [6][2] |
| `Similarity` | Swin-T/7 | $\mathcal{L}_G$ | $\mathcal{L}_L^{sim}$ | similarity | EsViT [28] |
| `Geometric` | Swin-T/7 | $\mathcal{L}_G$ | $\mathcal{L}_L^{geo}$ | geometry | New (ours) |

In the following section, we study the effect of the additional local cues by varying the total self-supervised objective in three different settings: `Vanilla`, `Similarity` and `Geometric`. An overview of the three settings can be found in Table 1. The sum of the global- and local-loss is the total objective which is optimized w.r.t. the parameters of the student network. Pseudo code for our the `Similarity` and `Geometric` setting can be found in the appendix.

#### 3.4.4 Computational complexity of the local loss

The local loss leads to limited compute overhead since it only adds an additive term in both the forward and backward pass which is very small compared to the backbone computations. In the `Similarity` setting (with local loss $\mathcal{L}_L^{sim}$), given two dense-representations $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{HW \times d}$, the compute complexity is $O(H^2W^2d)$. This is for $(HW)^2$ inner products, each of cost $O(d)$. The argmax operation is only $O(HW)$. In the `Geometric` setting (with local loss $\mathcal{L}_L^{geo}$), given two positional encodings $\boldsymbol{e}$ and $\boldsymbol{e}' \in \mathbb{R}^{HW \times 2}$, the compute complexity is $O(H^2W^2)$. This is for $(HW)^2$ L2-norms, each of cost $O(1)$. The argmin operation is only $O(HW)$. With a Swin-T backbone and 224x224 input images, $H = W = 7$ and $d = 192$ this results in a negligible cost compared to the computations in the backbone.

[2]Note that we replace the ViT backbone with a Swin transformer so that the only difference between the three settings is the local loss.

# 4. Results

## 4.1. Rationale of the experiment design

To evaluate the merits of the additional geometric local self-distillation, we compare the downstream performance of this method with the `Vanilla` and `Similarity` settings in which the local loss is removed or the geometric local loss is replaced by a similarity local loss (SOTA method). We compare the 3 representation learning methods on ImageNet-1k using the linear and k-NN benchmarks which are industry standard evaluations (Sec. 4.4). To get a grasp of the robustness of the methods depending on the dataset size, we run these benchmarks on randomly sampled subsets of ImageNet-1k. We observe an improvement of our method in all data regimes as well as a large performance drop for the `Similarity` setting (SOTA method). To corroborate our results in low-data regimes, we run the same study on smaller scale datasets (as well as multiple different backbones) in which analogous conclusions can be drawn (Sec. 4.5). We hypothesise that the large performance drop of the `Similarity` setting can be due to a collapse at the local level and show empirical evidence to confirm that (Sec. 4.6). Additionally, we propose a correspondence matching analysis (both qualitative and quantitative) to observe the effect of the local losses (Sec. 4.7).

We mostly focus on classification downstream tasks as opposed to dense tasks *e.g*. object detection. Dense evaluations are usually solved by fine tuning Mask-RCNN [24] on top of the pretrained backbone. As such it is hard to distinguish whether a high downstream accuracy is due to a good pretraining or due to the added capacity of Mask-RCNN. Recent work (see Table 1 of [29]) even shows better downstream accuracy on a randomly initialized network than on a pretrained one with MoCo v3. **k-NN and linear evaluation bechmarks for classification are better candidates to evaluate the intrinsic quality of the pretraining** since they don't require much processing. We do evaluate a dense downstream task with little processing in Section 4.7.

## 4.2. Implementation details

Our backbone of choice is the Swin transformer [31] as it outputs the necessary local-representations required for our local loss. We follow the implementation details from [6] and [28]. We use the *adamw* optimizer [33] with a batch size of 512 and train for a total of 300 epochs. The learning rate is linearly increased during the first 10 epochs to its maximum value of $0.0005 * \text{batchsize}/256$ as proposed by [19]. It is then reduced throughout the training with a cosine schedule [32]. We also use the sharpening and centering tricks from [6] to avoid collapse. Regarding the augmentations, we use two global- and 8 local-crops (see appendix). We refer the reader to [6] for more details.

Table 2. **Comparison of multiple methods with similar throughput with ImageNet-1k pretraining.** Rows in blue are results coming from our own runs to study the benefit of the additional local self-distillation.

| Method | Backbone | #Params | FLOPS | #Epochs | Linear | k-NN |
|---|---|---|---|---|---|---|
| SimCLR [7] | ResNet-50 | 24M | 4B | 800 | 69.3 | - |
| SimCLR v2 [8] | ResNet-50 | 24M | 4B | 800 | 71.7 | - |
| BYOL [20] | ResNet-50 | 24M | 4B | 1000 | 74.3 | - |
| DINO [6] | ViT-S/P=16 | 21M | 4.6B | 800 | 77.0 | 74.5 |
| MoCo v3 [11] | ViT-S/P=16 | 21M | 4.6B | 600 | 73.4 | - |
| EsViT [28] | Swin-T/W=7 | 28M | 4.5B | 300 | 78.0 | 75.7 |
| `Vanilla` | Swin-T/W=7 | 28M | 4.5B | 300 | 77.0 | 74.2 |
| `Similarity`[3] | Swin-T/W=7 | 28M | 4.5B | 300 | 77.9 (+ 0.9) | 75.3 (+ 1.1) |
| `Geometric` | Swin-T/W=7 | 28M | 4.5B | 300 | 77.8 (+ 0.8) | 75.4 (+ 1.2) |

## 4.3. Evaluation benchmarks

We follow the two most common ImageNet [14] unsupervised benchmarks from the literature [6, 45, 23, 20] *i.e*. the linear and $k$-NN benchmarks. In both cases, the backbone network and MLP-heads are trained on the training set without using labels. For the linear evaluation, a linear layer is added on top of the frozen global-representation $\bar{\boldsymbol{z}}$ and is trained using the training set (data-augmentations are used) including the labels. The classification accuracy on the test set is evaluated using a center-crop of 224x224. This evaluation protocol is quite computationally intensive as the model needs to compute a forward pass for multiple epochs. The $k$-NN benchmark on the other hand only needs one pass. For each image in both the training and test set, the global-representation of a center-crop (224x224) is computed. Then, each image from the test set gets a label assigned based on a vote from the $k$ nearest neighbors in the training set (anchor points). We use $k = 20$ to stay consistent with previous works.

## 4.4. ImageNet-1k

Both the linear and $k$-NN benchmark results are reported in Table 2. The first block of rows compares previous works (including SOTA) with backbones of similar computational requirements. These include ResNet-50 [25], ViT-Small [17] and Swin-Tiny [31]. The second block of rows (in blue) are results coming from our own runs to study the benefit of the additional local cues. These runs were trained for 7 days on 8x NVIDIA A100. The `Similarity` and `Geometric` matchings outperform the `Vanilla` method which enforces coherence only at the global level confirming that the additional local regularization is helpful.

## 4.5. Other datasets

To get a better idea of the robustness of the additional self-supervised loss at the local level, we train all methods on other datasets. We introduce the local loss to get stronger

[3] In theory, this row should match the the row EsViT [28]. However, the authors of [28] do not report the downstream evaluations of the last epoch but select the best epoch. This explains the slightly lower performance of the blue row. More details can be found on their Github. All evaluations from our own runs evaluate the model after the final epoch of pretraining.

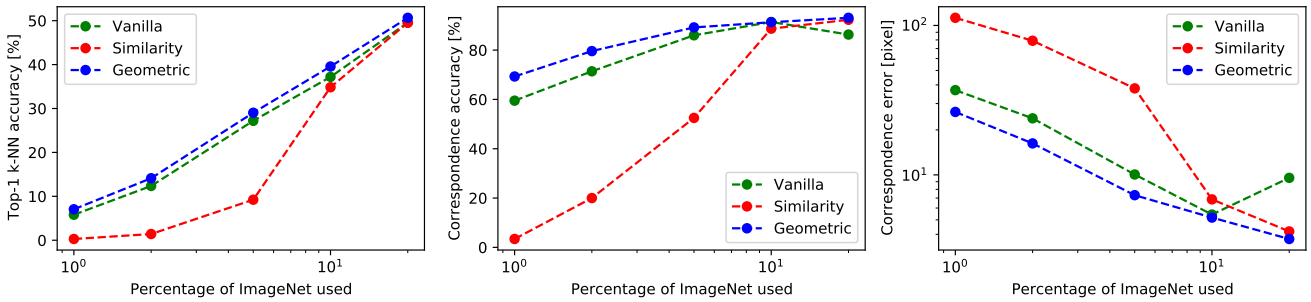


Figure 4. **Comparison of the performance between the three different settings on ImageNet-1k subsets.** Data points on the x-axis are at 1%, 2%, 5%, 10%, 20%. The left plot shows the Top-1 k-NN accuracy. The two other plots are linked to the correspondence matching Section 4.7 where the center and right plot respectively correspond to the accuracy and the error. The `Geometric` setting shows better performance in all metrics.

self-supervision and more efficient learning, which is best studied by looking at the behavior on small scale datasets. These include an artificial setting where we sample 1%, 2%, 5%, 10% and 20% subsets of ImageNet-1k in order to evaluate the relative performances in low-data regimes. Even though this is an artificial setting, these subsets are well curated, image-centered and contain a lot of diversity making them ideal for such a study. We also include evaluations on 3 other datasets: *Food-101* [4], *NCT-CRC-HE-100K* [27] and ImageNet-100 (100 class subset of ImageNet-1k).

#### 4.5.1 ImageNet-1k subsampled

The 1%, 2%, 5%, 10% and 20% subsets are obtained by sampling respectively 10, 20, 50, 100, 200 images from each class to avoid imbalances. Each method is independently trained on a subset and evaluated on the $k$-NN benchmark. Note that this evaluation can be done using the full training set or the training subsets. Since we are using very little training data (*e.g.* 1%), we choose to evaluate the anchor points on the full training data to make the metric more robust and fair across all subsets. There are two main observations from the left plot of Figure 4: 1) incorporating local cues using similarity matchings is hurtful for small subsets and 2) geometric matchings on the other hand are robust and provide additional accuracy on all subsets. Similarities between local-representations in low-data regimes are mostly based on low-level features leading to collapse of the matching function. We will confirm this in the following section. Note that the y-scale of the left plot of Figure 4 goes from 0 to 50%: the relative difference between the vanilla and the geometric matching method is in order of 1.5% which is highly significant on this benchmark. Still, the size of the dataset remains the dominant factor and regularization based on local correspondences cannot replace that.

Table 3. **Performance comparison of `Vanilla`, `Similarity` and `Geometric` on the $k$-NN and linear evaluation benchmarks.** Rows in different shades of gray are trained on the training set of Food-101, NCT-CRC-HE-100K, ImageNet-100 and evaluated on the corresponding test set. Each shade of gray represents a different model (backbones from top to bottom: Swin-T/7x7, Swin-T/14x14 and Swin-S/7x7). As a point of reference, the blue rows are trained on ImageNet-1k and evaluated similarly (backbone: Swin-T/7x7). NA entries mean the training crashed due to numerical instabilities.

| | `Vanilla` | | `Similarity` | | `Geometric` | |
|---|---|---|---|---|---|---|
| | $k$-NN | linear | $k$-NN | linear | $k$-NN | linear |
| Food-101 | 69.3 | 79.4 | 1.7 (-67.6) | 3.0 (-76.4) | 73.1 (+3.8) | 82.2 (+2.8) |
| NCT-CRC-HE-100K | 91.9 | 92.1 | 46.0 (-45.9) | 53.5 (-38.6) | 89.5 (-2.4) | 90.2 (-1.9) |
| ImageNet-100 | 76.5 | 82.0 | 76.2 (-0.3) | 81.4 (-0.6) | 79.5 (+3.0) | 84.4 (+2.4) |
| Food-101 | 69.5 | 78.5 | 0.9 (-68.6) | 2.1 (-76.4) | 72.6 (+3.1) | 80.7 (+2.2) |
| NCT-CRC-HE-100K | 90.8 | 90.1 | 44.1 (-46.7) | 34.1 (-56.0) | 91.0 (+0.2) | 89.4 (-0.7) |
| ImageNet-100 | 76.8 | 81.6 | 2.1 (-74.7) | 1.9 (-79.7) | 78.9 (+2.1) | 83.0 (+1.4) |
| Food-101 | 70.9 | 79.6 | NA | NA | 73.8 (+2.9) | 82.5 (+2.9) |
| NCT-CRC-HE-100K | 90.8 | 89.4 | NA | NA | 90.5 (-0.3) | 86.9 (-2.5) |
| ImageNet-100 | 78.6 | 83.1 | 76.1 (-2.5) | 80.9 (-2.2) | 80.2 (+1.6) | 84.1 (+1.0) |
| Food-101 | 67.7 | 81.4 | 68.4 (+0.7) | 82.2 (+0.8) | 68.6 (+0.9) | 82.2 (+0.8) |
| NCT-CRC-HE-100K | 89.2 | 93.3 | 90.8 (+1.6) | 93.1 (-0.2) | 90.3 (+1.1) | 94.2 (+0.9) |
| ImageNet-100 | 86.2 | 88.1 | 87.0 (+0.8) | 88.8 (+0.7) | 87.5 (+1.3) | 88.3 (+0.2) |

#### 4.5.2 Smaller scale datasets & different models

The evaluation on *Food-101* [4], *NCT-CRC-HE-100K* [27] and ImageNet-100 with different models (Swin-T/7x7, Swin-T/14x14 and Swin-S/7x7) can be found in Table 3. The datasets are chosen because they all contain about 100k images and images have a resolution similar to ImageNet-1k. Rows in light gray are trained from scratch on the training set of Food-101, NCT-CRC-HE-100K, ImageNet-100 and evaluated on the corresponding test set. Rows in darker gray are trained on ImageNet-1k and evaluated on the test set of Food-101, NCT-CRC-HE-100K, ImageNet-100. In most evaluations, the additional geometric local regularization improves the performance on the downstream tasks. A key finding from Table 3 is that the similarity based local loss used in SOTA work EsViT [28] is hurtful in low-data regimes while the geometric local loss is robust and shows improvements in almost all evaluations. The takeaways are similar irrespective of model size and window size. The performance drop due to collapse (see Sec. 4.6) does a appear a bit worse when using Swin-T/14x14 compared to Swin-T/7x7. This can be explained by the fact that a larger window size allows tokens to attend to a wider set of tokens in

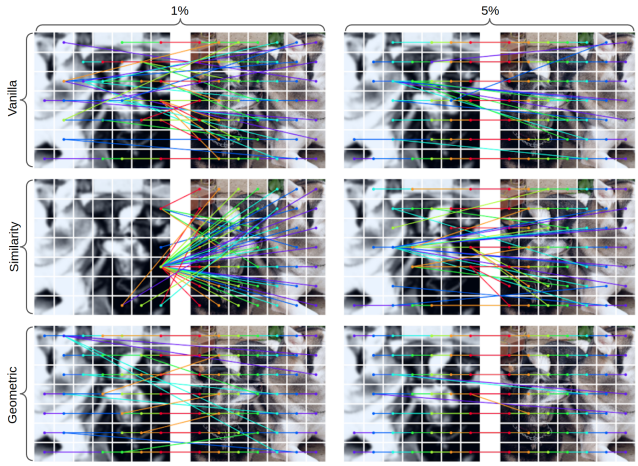


Figure 5. **Visual comparison of the similarity correspondence between two augmentations.** This is done for `Vanilla`, `Similarity` and `Geometric` settings on 2 training data regimes: 1% and 5% of the full ImageNet training data. Every location on the right side is matched to a location on the left image based on the distance in the learned feature space. Colors are used only to better distinguish different matchings. Best viewed in color and zoomed in.

general, or in particular, a wider set of similar tokens during the transition to a collapsed state.

### 4.6. Collapse of the similarity matching function

In general, self-supervised methods (even with a single global loss) are prone to mode collapse since they cannot use labels as targets for their outputs and instead have to bootstrap their own outputs during training. As such, a lot of care has to be put in the design of the training algorithm. In contrastive learning methods, a loss function with appropriate negative samples mitigates the issue [7]. Analogously in self-distillation methods, a careful tuning of the temperature parameters in the centering and sharpening trick is required [6]. In a dual global-local objective framework, collapse can also occur at the local level. Collapse at the local level can happen when using the `Similarity` setting proposed in [28]. Such failure cases are shown in the appendix. Due to the nature of the backbone, collapse at the local level implies collapse at the global level. That is because the global-representation $\bar{\boldsymbol{z}} = \bar{f}(\boldsymbol{x})$ is a direct function of the dense representation $\boldsymbol{z}$ i.e. $\bar{\boldsymbol{z}} = g(\boldsymbol{z})$ with $g$ an average pooling layer or attention layer. When collapse occurs (both global and local), we get that $\nabla_{\boldsymbol{x}} \bar{f}(\boldsymbol{x}) \approx \mathbf{0}$. That is, the model discards all information from the input image $\boldsymbol{x}$ leading to downstream evaluations close to a random accuracy of $\frac{1}{nb\_class}$ as can be seen in some entries of the `Similarity` column of Table 3.

If collapse of the matching function occurs, the method cannot recover because $\mathcal{L}_L^{sim}$ is enforced making the collapse even worse. The `Geometric` setting avoids this issue by construction leading to a more robust training.

### 4.7. Correspondence matching based on similarity

We analyze the learned representation by looking at the quality of correspondence matching based on the similarities of local-representations. Two augmentations of an input image from the validation set are computed using a tweaked data augmentation pipeline where there always exists an exact correspondence mapping between the local-representations *i.e*. augmentations are always cropped and resized in the same manner. This spatial correspondence mapping is used as ground truth and we evaluate the matchings obtained using token similarity for all three settings. In the center and right part of Figure 3, the results are shown using two metrics: 1) the classification accuracy (i.e., how many of the local representations are matched correctly) and 2) the distance error, both w.r.t. the ground truth correspondence mapping. The geometric matchings processes images in a way that better preserves the spatial information. Qualitative evaluations can be found in Figure 5.

## 5. Conclusion & Future work

Self-supervised training of visual transformers using self-distillation is becoming the standard way of obtaining visual representation from images by solving a proxy task at the image-level (global-level). We can leverage additional self-supervision by incorporating self-distillation at the local-level. This is done by enforcing coherence between pairs of local-representations (acts as a regularizer). We observe an improvement on downstream tasks using multiple datasets. We study the effect of the matching function used to generate pairs of local-representations from both augmentations. A geometry based matching function shows advantages over a similarity based matching function both in terms of 1) higher performance on downstream tasks and 2) better preservation of the spatial relations of the input images. This is particularly true in low-data regimes, in which case we observe a collapse of the similarity based matching function in some settings. We believe the insights from this paper can lead to a better crafting of a data-driven local-representation matching function to explicitly avoid collapse and that an upscaling of these methods to very large backbones can surpass the current state of the art.

## 6. Acknowledgements

This paper is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101021347). The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation Flanders (FWO) and the Flemish Government – department EWI.

## References


[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: BERT pre-training of image transformers. *CoRR*, abs/2106.08254, 2021.

[3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 517–526. PMLR, 06–11 Aug 2017.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *CoRR*, abs/2006.09882, 2020.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *CoRR*, abs/2011.10566, 2020.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *CoRR*, abs/2104.02057, 2021.

[12] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *CoRR*, abs/2102.10882, 2021.

[13] Ozan Ciga, Tony Xu, and Anne L. Martel. Resource and data efficient self supervised learning, 2021.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[15] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Unsupervised pretraining for object detection by patch reidentification. *CoRR*, abs/2103.04814, 2021.

[16] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[19] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021.

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019.

[27] Jakob Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nek Valous, Dyke Ferber, Lina Jansen, Constantino Reyes-Aldasoro, Inka Zoernig, Dirk Jäger, Hermann Brenner, Jenny Chang-Claude, Michael Hoffmeister, and Niels Halama. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Medicine*, 16:e1002730, 01 2019.

[28] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *CoRR*, abs/2106.09785, 2021.

[29] Y. Li, S. Xie, X. Chen, P. Dollar, K. He, and R. Girshick. Benchmarking detection transfer learning with vision transformers, 2021.

[30] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small-size datasets, 2021.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[32] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016.

[33] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018.

[34] T. N. Mundhenk, D. Ho, and B. Y. Chen. Improvements to context based self-supervised learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018.

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 69–84, Cham, 2016. Springer International Publishing.

[36] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[37] Pedro O. Pinheiro, Amjad Almahairi, Ryan Y. Benmalek, Florian Golemo, and Aaron C. Courville. Unsupervised learning of dense visual representations. In *NeurIPS*, 2020.

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021.

[39] Charlie Saillard, Olivier Dehaene, Tanguy Marchand, Olivier Moindrot, Aurélie Kamoun, Benoit Schmauch, and Simon Jegou. Self supervised learning improves dmmr/msi detection from histology slides across multiple cancers, 2021.

[40] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *CoRR*, abs/1803.02155, 2018.

[41] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *CoRR*, abs/2005.10243, 2020.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[43] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3024–3033, June 2021.

[44] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021.

[45] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *CoRR*, abs/1805.01978, 2018.

[46] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *CoRR*, abs/2105.04553, 2021.

[47] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16684–16693, June 2021.

[48] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Regioncl: Can simple region swapping contribute to contrastive learning? *CoRR*, abs/2111.12309, 2021.

[49] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. *CoRR*, abs/2102.08318, 2021.

[50] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *CoRR*, abs/2107.00641, 2021.

[51] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding, 2021.