Leverage Your Local and Global Representations: A New Self-Supervised Learning Strategy

Tong Zhang[1] Congpei Qiu[2] Wei Ke[2] Sabine Süsstrunk[1] Mathieu Salzmann[1]
[1] School of Computer and Communication Sciences, EPFL, Switzerland
[2] Xi'an Jiaotong University, China

Abstract

Self-supervised learning (SSL) methods aim to learn view-invariant representations by maximizing the similarity between the features extracted from different crops of the same image regardless of cropping size and content. In essence, this strategy ignores the fact that two crops may truly contain different image information, e.g., background and small objects, and thus tends to restrain the diversity of the learned representations. In this work, we address this issue by introducing a new self-supervised learning strategy, LoGo, that explicitly reasons about **Lo***cal and* **Glo***bal crops. To achieve view invariance, LoGo encourages similarity between global crops from the same image, as well as between a global and a local crop. However, to correctly encode the fact that the content of smaller crops may differ entirely, LoGo promotes two local crops to have dissimilar representations, while being close to global crops. Our LoGo strategy can easily be applied to existing SSL methods. Our extensive experiments on a variety of datasets and using different self-supervised learning frameworks validate its superiority over existing approaches. Noticeably, we achieve better results than supervised models on transfer learning when using only* $1/10$ *of the data.*[1]

1. Introduction

Building on the great success of supervised learning in visual tasks such as image classification [20, 25, 26] and object detection [15, 19], significant efforts have recently been dedicated to learning high-level representations without human annotations. Inspired by the pre-training stage in natural language processing, e.g. GPT [32] and BERT [13], such a self-supervised learning (SSL) approach aims to learn representations that extract useful information for a downstream task in an unsupervised manner, thus providing an effective initialization to start from when some annotated data for the downstream tasks become available. Recently, SSL has been proven to be as effective as supervised pre-training, or even more effective in some cases [6, 10].

The basic principle behind existing SSL approaches can be traced back to [17, 29] and consists of learning a representation that is shared across different views of the same input, yet carries discriminative information. In vision tasks, this is typically achieved by maximizing the similarity between two augmented views of the same image while penalizing trivial solutions using various techniques. For example, contrastive learning [9, 18] incorporates negative pairs, where one view comes from a different image, to prevent the network from constantly generating the same output; non-contrastive methods [11, 16] only rely on positive pairs by modifying the back-propagation mechanism to prevent collapse; clustering-based methods [2, 6] perform online clustering to keep the consistency between exemplar representations (the centroids of clusters) and different views of the same image.

Intuitively, one should expect the representations of random crops with smaller sizes to have a larger variance than that of larger crops because, as shown in Figure 1, they may truly encode entirely different content. Nevertheless, existing methods encourage *all* the random crops of the same image to have similar representations. This complicates the learning process and tends to lead the network to discarding valuable image information to achieve such invariance. This was, for example, observed in [7], where the multi-crop strategy of [6] was shown to yield a performance drop when applied to other SSL methods, such as BYOL [16], SimSiam [11], and MoCo [18]

In this paper, we address this limitation by introducing a new multi-crop SSL strategy, LoGo, which exploits the relationships between **lo**cal and **glo**bal image patches in different, well-adapted ways, and can easily be integrated into existing SSL frameworks. Specifically, we exploit two different kinds of crops: Large ones that encompass a global view of the input image, thus being well-suited to learn a view-invariant representation; and small ones with a higher

[1]Our code and pretrained models can be found at https://github.com/ztt1024/LoGo-SSL. Correspondence to Ke Wei (wei.ke@mail.xjtu.edu.cn).

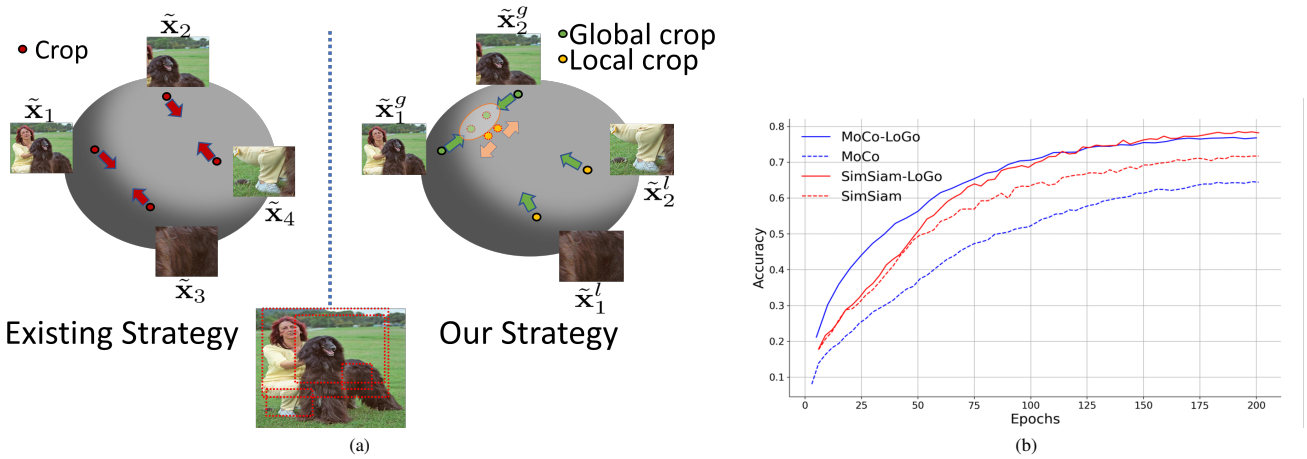

Figure 1. (a) Overview of our self-supervised learning strategy. To learn a view-invariant representation that nonetheless encodes semantic information about local objects, we seek to maximize the similarity between global crops while allowing the local crops to remain distant from each other, thus accounting for the fact that local crops may represent entirely different objects. (b) Monitoring of the KNN top-1 accuracy on ImageNet-100 with a ResNet-34 backbone evidences the benefits of our approach in different SSL strategies.

variance that focus on local image regions, thus allowing the model to encode information such as background, texture, and objects. As illustrated in Figure 1, we then design a loss function that (i) pulls the global representations of the same image close to each other, while also encouraging each local representation of that image to be close to the global ones; (ii) favors the different local representations to remain distant to account for the differences between the local patches. Altogether, this provides the model with the flexibility to keep apart the local representations that encode different regions while nonetheless encouraging the representations of all crops from the same image to cluster in the latent space.

Furthermore, to account for the fact that traditional distance metrics may be unreliable in a high dimensional space [1], we introduce a new approach to evaluate the similarity between the representations of two patches. Specifically, based on the assumption that the similarity of two local crops from the same image is greater than that of two local crops from different images with high probability, we train an MLP to discriminate between pairs of local crops from the same or from different images, and exploit its prediction as a similarity score.

Our contributions can be summarized as follows:

- We exploit both global and local views in SSL to encode rich semantic information. To this end, we encourage similarity across global crops to achieve view invariance, but allow the local crops to be dissimilar to maintain the diversity of local object representations.
- We introduce a learnable similarity measure to overcome the limitations of standard metrics in high dimensional feature space.
- Our approach generalizes to different SSL frameworks, including constrastive (e.g., MoCo [18]) and non-contrastive (e.g., SimSiam [11]) ones.
- Our approach allows the network to be trained on smaller datasets, which benefits downstream tasks where the training-testing domain gap is large.

We demonstrate the benefits of our approach over the state-of-the-art SSL techniques on several datasets. Importantly, our strategy enables the self-supervised models to surpass their supervised counterparts on dense prediction tasks with only 1/10 of the training data.

2. Related Work

SSL or representation learning frameworks can be roughly grouped into two categories: Those that are trained on pretext tasks, such as solving jigsaw puzzles [30] or predicting color from grayscale images [39], and those that optimize different learning objectives. Our work falls in this second category, and we, therefore, focus the discussion below on the methods that also do.

Contrastive learning methods. Contrastive learning aims to maximize a notion of affinity between pairs of positive samples while minimizing the affinity between negative pairs. This is typically achieved by optimizing the InfoNCE loss [31]. To obtain diverse and discriminative feature representations, contrastive learning typically leverages data augmentation. For example, Deep InfoMax [21], and its multi-scale version [3] aim to maximize the mutual information between the global and local features of an input image, that is, the feature vectors of the last layer after global pooling and the ones across all the channels

in each location. Their positive pairs are defined using a single view of an image, which limits the diversity of the learned representation. CMC [35] maximizes the mutual information between the feature representations of different modalities, e.g. semantic map, YCbCr, or depth map of the same image. SimCLR [9] is the first to augment each image twice and create positive pairs of distorted-original images and negative pairs using two different images. MoCo [18] improves the contrastive training by using a memory bank to store negative pairs and avoid degenerate solutions. MoCo-V2 [10] shows that stronger augmentations and the use of multiple crops boost the performance of self-supervised learning. Furthermore, Wang & Isola [36] provide theoretical proof of reinterpreting the InfoNCE loss as two terms: aligning features that belong to the same instance and spreading normalized learned features on a hypersphere. However, the theory can only be applied to the contrastive case and the empirical performance improvement is marginal.

Non-contrastive learning methods. One of the main difficulties in contrastive learning consists of defining meaningful negative pairs. To counteract this, BYOL [16] demonstrates that using only positive pairs is sufficient to avoid degenerate solutions when exploiting a Siamese network with one branch acting as a momentum encoder and used to supervise the other branch. Subsequently, SimSiam [11] proposes a simpler Siamese network, arguing that momentum is not required but that a predictor and stop-gradient are. This approach appends a predictor to one branch of the Siamese backbone and stops the gradient of that branch from being back-propagated to the backbone.

Clustering-based methods. Clustering itself has been an important research direction in unsupervised learning [5, 8, 22, 37, 38, 40, 41], and is nowadays used for representation learning. For example, DeepCluster [5] alternately clusters the learned representations and predicts the cluster assignments; SeLa [2] simultaneously learns the representation and the cluster assignments by using the Sinkhorn-Knopp algorithm to perform online updates; SwAV [6] utilizes the same technique within a Siamese network to compute soft assignments from one view, which supervise the feature distribution in the other view. SwAV [6] further demonstrates that using multiple crops for each image helps their training. However, SwAV does not reason about the potential lack of shared information between multiple local crops, which is what we achieve here. Furthermore, the above-mentioned methods require either additional memory bank [2, 5] or very large batch sizes [6] to yield a stable and robust optimization.

Recently, the transformer-based Dino [7] network, a follow-up work of SwAV, proposes to use global views as teachers to supervise the local views' probability-like representation. However, this method inherently encourages the local crops to have similar representations to the global ones even though they may contain different objects.

In short, all of the existing methods encourage all the crops, regardless of their actual semantic information, to have similar representations. As such, to achieve view invariance, they tend to discard relevant semantic information, thus undermining the ability to transfer the resulting representations to downstream tasks. Here, we, therefore, propose a new SSL strategy that addresses this limitation.

3. Methodology

Our goal is to develop a self-supervised learning approach that is able to handle complex images depicting objects of different semantics. We aim for our approach to be general, and thus applicable to both contrastive and non-contrastive learning strategies. Therefore, below, we first review the contrastive and non-contrastive paradigm together with a representative framework for each, namely MoCo [18] and SimSiam [11]. Subsequently, we introduce our hierarchical local-global model and our approach to learning a similarity measure.

Notation. We use τ^g and τ^l to denote the operation sets for global and local augmentation, with r_g and r_l denoting the lower bound of the global crops' size and the upper bound of the local crops' size, respectively. The global and local views, namely $\tilde{\mathbf{x}}^g$ and $\tilde{\mathbf{x}}^l$, are obtained by applying τ^g and τ^l to the same image $\mathbf{x} \in \mathbb{R}^{W \times H}$, where W and H are the image width and height. Similarly, $\mathbf{z} \in \mathbb{R}^n$ denotes the latent representation obtained by the encoder function $f_{\theta_e} : \mathbb{R}^{W \times H} \rightarrow \mathbb{R}^n$, and $\mathbf{z}^+$ and $\mathbf{z}^-$ are its corresponding positive and negative counterpart, respectively.

3.1. Similarity Loss

Learning a feature representation without supervision is typically achieved by maximizing the similarity between the samples in positive pairs, while optionally minimizing the similarity of those in negative pairs. Our approach can be applied to most SSL techniques. To illustrate this, we therefore consider two typical similarity loss functions: Info-NCE [9, 18, 31, 35], commonly-used in contrastive learning, and the cosine loss [11, 16], often employed in the non-contrastive scenario.

Info-NCE was introduced by CPC [31] and can be expressed as

$$\mathcal{L}^{NCE}(\mathbf{z}, \mathbf{z}^+, \mathbf{z}^-) = -\log \frac{\exp\left(\mathbf{z} \cdot \mathbf{z}^+ / \tau\right)}{\exp\left(\mathbf{z} \cdot \mathbf{z}^+ / \tau\right) + \sum \exp\left(\mathbf{z} \cdot \mathbf{z}^- / \tau\right)}, \tag{1}$$

where τ is a temperature hyper-parameter, and $\mathbf{z}$ is the feature representation of augmented images encoded by f_{θ_e}, i.e, $\mathbf{z} = f_{\theta_e}(\tilde{x})$. $\mathbf{z}^+$ is a positive sample and $\mathbf{z}^-$ is a negative one, which could be sampled from a memory bank [18] or obtained using a large batch size [9].

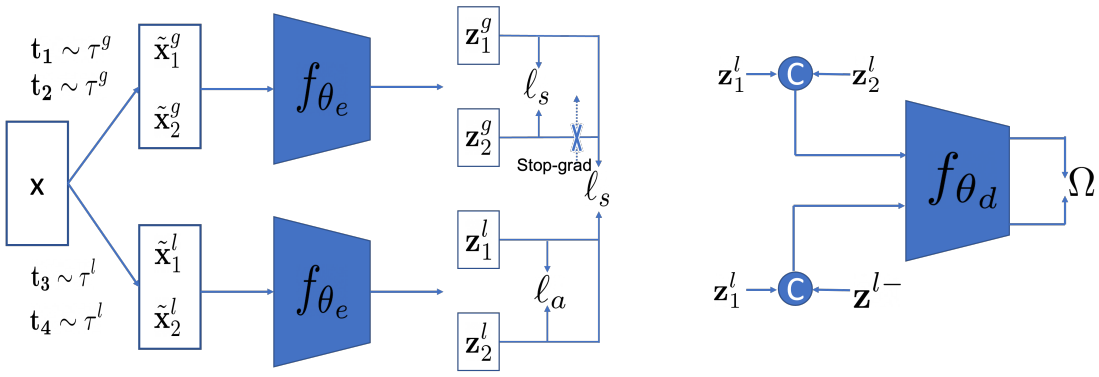

Figure 2. Our LoGo structure (left) and local affinity measure f_{θ_d} (right). f_{θ_e} represents the feature encoder, which includes a backbone network followed by a multi-layer perceptron. Each image is augmented into global and local crops which are fed to the encoder. We maximize the global-to-global and local-to-global similarity by optimizing ℓ_s, which can be either the cosine or InfoNCE loss. Simultaneously, we maximize the dissimilarity between pairs of local crops by optimizing the output of a learned similarity measure ℓ_a. Note that $\mathbf{z}^l$ is detached from the encoder, and no gradients are back-propagated to the encoder when training f_{θ_d}.

By contrast, the cosine loss used in SSL does not exploit negative samples. It can be written as

$$\mathcal{L}^{cos}(\mathbf{z}_1, \mathbf{z}_2) = -\frac{h(\mathbf{z}_1)}{\|h(\mathbf{z}_1)\|_2} \cdot \frac{\mathbf{z}_2}{\|\mathbf{z}_2\|}, \tag{2}$$

where h is an MLP layer used to predict the "mean" of the set of positive samples for $\mathbf{z}$. In this context, SimSiam [11] uses Siamese networks and stops the back-propagation for the $\mathbf{z}_2$ branch, whereas BYOL [16] uses a momentum encoder to update the encoder parameters.

3.2. Our Approach

In the presence of complex image content, such as multiple objects, existing approaches to generating positive, and optionally negative pairs, suffer from several drawbacks. First, depending on the random cropping, two different views of the same image might depict entirely different content. Conversely, two different images may share some content, and thus crops from these different images might in fact depict the same object category. Directly applying existing SSL strategies yields highly noisy and potentially contradictory constraints, thus complicating the learning process.

To address this, we exploit two different kinds of crops, local and global ones. Specifically, for each input image $\mathbf{x}$, we extract to two global views $\tilde{\mathbf{x}}^g_{1,2}$ and two local views $\tilde{\mathbf{x}}^l_{1,2}$ from the augmentation sets τ^g and τ^l, respectively. We then optimize the global-to-global, local-to-global, and local-to-local relationships respectively. Note that, below, we use ℓ_s to denote a general similarity loss, which in our experiments will be either Eq. 1 or Eq. 2.

Global-to-global. Because the global views encompass most of the semantic content of the original image, we aim to reach a consensus among their representations by maximizing the similarity between global views from the same image, while optionally minimizing the similarity between the global views of different images in contrastive cases. We therefore write a global-to-global loss as

$$\mathcal{L}_{gg} = \mathbb{E}_{\mathbb{P}_{\mathbf{z}^g}}[\ell_s(\mathbf{z}^g_1, \mathbf{z}^g_2)], \tag{3}$$

where $\mathbf{z}^g_1 = f_{\theta_e}(\tilde{\mathbf{x}}^g_1)$, $\mathbf{z}^g_2 = f_{\theta_e}(\tilde{\mathbf{x}}^g_2)$, and $\mathbb{P}_{\mathbf{z}^g}$ is the distribution of $\mathbf{z}^g$, where $\mathbf{z}^g \sim P(\mathbf{z}|\mathbf{x}^g)$.

Local-to-global. We use the global crops as "anchor" points for their local crops because their larger crop size ensures that they will share some semantic content with the local crops. We, therefore, define a loss function that makes the local representations move closer to their corresponding global ones. Because, here, the global representations act as supervisory signals to the local ones, we either fix the global representations in the momentum encoder or stop their gradient in the back-propagation process. This yields the loss:

$$\mathcal{L}_{lg} = \mathbb{E}_{\mathbb{P}_{\mathbf{z}^g, \mathbf{z}^l}}[\sum_{i=1,2}(\ell_s(\mathbf{z}^l_i, \text{sg}(\mathbf{z}^g_1)) + \ell_s(\mathbf{z}^l_i, \text{sg}(\mathbf{z}^g_2)))], \tag{4}$$

where $\text{sg}(\cdot)$ stands for either the stop gradient operation in, e.g., SimSiam, or the momentum encoder in, e.g., MoCo.

Local-to-local. In the presence of complex image content, we expect two local views from the same image to often depict different semantic objects. Therefore, instead of encouraging local view similarity as in most existing works, we encourage their dissimilarity, thus preventing degenerate solutions where all local patches converge to the same representation independently of their content. Given an affinity function ℓ_a, we express maximizing the local-to-local dissimilarity as minimizing the loss

$$\mathcal{L}_{ll} = \mathbb{E}_{\mathbb{P}_{\mathbf{z}^l}}[\ell_a(\mathbf{z}^l_1, \mathbf{z}^l_2)]. \tag{5}$$

While one could in principle use any standard similarity measure, such as the cosine similarity, as an affinity function ℓ_a, the high dimensionality of the feature space may lead to learning meaningless representations. Indeed, in high dimensional space, many directions allow one to push points away [1], and thus we need to find a direction that nonetheless encodes meaningful information.

To achieve this, we leverage the intuition that, although different images may contain local regions that depict the same semantic content, we expect on average local crops within an image to be more closely related than local crops from two different images. To encode this intuition, inspired by the Mutual Information Neural Estimator (MINE) [4], we make use of an auxiliary regressor $f_{\theta_d} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$, which outputs a similarity value between two local crops. The parameters θ_d of this regressor are trained jointly with the other parameters of our approach. To this end, we seek to maximize the cost function

$$\Omega(\theta_d) = \mathbb{E}_{\mathbb{P}_{\mathbf{z}_1^l,\mathbf{z}_2^l}}[f_{\theta_d}(\mathbf{z}_1^l, \mathbf{z}_2^l)] - \mathbb{E}_{\mathbb{P}_{\mathbf{z}_1^l \otimes \mathbf{z}^{l-}}}[f_{\theta_d}(\mathbf{z}_1^l, \mathbf{z}^{l-})]. \tag{6}$$

where $\mathbf{z}^{l-}$ is that of a local crop from a different image, which can be randomly sampled in the same batch and $\mathbb{P}_{\mathbf{z}_1^l \otimes \mathbf{z}^{l-}}$ is the product of two marginal distribution. By training it jointly with the encoder f_{θ_e}, the regressor f_{θ_d} will adjust its similarity value based on the feature space distribution.

We then leverage the trained regressor to define our affinity function. That is, given

$$\theta_d^* = \underset{\theta_d}{\operatorname{argmax}}\, \Omega(\theta_d)\ , \tag{7}$$

we define

$$\ell_a = f_{\theta_d^*}\ . \tag{8}$$

We then use this definition of ℓ_a in the loss function of Eq. 5. In other words, we train an affinity function to make local crops from the same image appear to be more similar than local crops from different images, and then train the encoder so as to minimize the resulting affinity between local crops from the same image to account for the fact that they will often depict different semantic contents.

Altogether, our SSL problem can therefore be formulated as the bi-level optimization problem

$$\begin{aligned} &\min_{\theta_e} \mathcal{L}_{gg} + \mathcal{L}_{lg} + \lambda \mathcal{L}_{ll}, \\ &\text{s.t.} \quad \ell_a = f_{\theta_d^*} \\ &\qquad \theta_d^* = \underset{\theta_d}{\operatorname{argmax}}\, \Omega(\theta_d), \end{aligned} \tag{9}$$

where λ is a hyper-parameter balancing the dissimilarity and similarity terms, accounting for the fact that ℓ_a differs from the other terms. The details of our LoGo SSL strategy are provided in Algorithm 1.

Algorithm 1 LoGo Pseudocode

Input: batch size N, global and local augmentation τ^g and τ^l,
Initialization: encoder f_{θ_e}, similarity measure f_{θ_d}
while not reach epoch limits **do**
  sample image minibatch $\mathbf{X}$
  for j < N **do**
    augment an image $\mathbf{X}(j)$ to get $\mathbf{x}_{1,2}^g$ and $\mathbf{x}_{1,2}^l$
    Get local and global representation, $\mathbf{z}_{1,2}^l(j) \leftarrow f_{\theta_e}(\mathbf{x}_{1,2}^l)$ and $\mathbf{z}_{1,2}^g(j) \leftarrow f_{\theta_e}(\mathbf{x}_{1,2}^g)$
  end for
  for j < N **do**
    Get positive local views pairs $\mathbf{z}_{1,2}^l(j)$
    Random pick a local view $\mathbf{z}_1^l(k), k \neq j$
    Maximize the loss by 6 and update the f_{θ_d}
    Evaluate global views similarity loss $\mathbf{z}_{1,2}^g(j)$ by 3
    Evaluate local to global views similarity loss between $\mathbf{z}_{1,2}^g(j)$ and $\mathbf{z}_{1,2}^l(j)$ by 4
    Evaluate the local to local loss for each $\mathbf{z}_{1,2}^l(j)$ as 5
    Minimize the total loss as 9 and update the f_{θ_e}
  end for
end while
Output: The encoder network f_{θ_e}

4. Main Empirical Results

We assess the performance and generality of our LoGo representation learning strategy by exploiting it within two popular SSL frameworks, namely MoCo [18] and SimSiam [11], and denote the resulting models as MoCo-LoGo and SimSiam-LoGo. We implemented our approach with Pytorch and run all the experiments on either 4 NVIDIA GeForce RTX 3090 or 2 NVIDIA V100 GPUs.

4.1. Implementation Details

Optimization. For our comparisons to be fair, we run all the SSL learning experiments for 200 epochs with a cosine learning decay scheduler [10], leading a learning rate $\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(t\pi/T))$. Following SimSiam and MoCo, we use the SGD optimizer and set the momentum value to 0.9 and the weight decay to 0.0001. For MoCo, we find the best temperature values to be $\tau = 0.1, 0.2, 0.07$ for CIFAR10, STL10 and IN-100, and keep the same learning rate and batch size as the original MoCo and SimSiam.

Data augmentation. We follow the same augmentation operations as in MoCo v2 [10], including random cropping and resizing for the global and local views, random horizontal flipping, followed by random color jittering operations (brightness, contrast, saturation, and hue), and grayscale transformations. For ImageNet-100, we further add Gaussian blur.

	KNN (acc %)	Linear (acc %)
MoCo	79.58	80.37
MoCo+LoGo	**84.44**	**85.59**
SimSiam	80.48	83.02
SimSiam+LoGo	**87.67**	**88.02**

Table 1. Training on CIFAR10 with a ResNet-18 backbone. We show the top-1 accuracy for a KNN and a linear classifier.

	KNN (acc %)	Linear (acc %)
MoCo	72.13	79.07
MoCo+LoGo	**76.79**	**80.73**
SimSiam	60.8	71.66
SimSiam+LoGo	**76.96**	**80.61**

Table 2. Training on STL10 with a small Alexnet backbone. We show the top-1 accuracy for a KNN and a linear classifier.

	KNN (acc %)	Linear (acc %)
MoCo	64.18	68.48
MoCo+LoGo	**76.82**	**79.32**
SimSiam	71.21	75.48
SimSiam+LoGo	**78.48**	**80.94**

Table 3. Training on ImageNet-100 with a ResNet-34 backbone. We show the top-1 accuracy for a KNN and a linear classifier.

	ResNet-34	ResNet-50
MoCo	68.48	74.84
MoCo-LoGo	79.32	85.14

Table 4. Linear classification accuracy (%) of the Moco and MoCo-LoGo on IN-100 with ResNet-34 and ResNet-50 as backbone feature encoder.

Regressor design. The regressor f_{θ_d} consists of five fully-connected layer+synchronized batch normalization+ ReLU blocks, followed by a projection head and a softplus activation function that outputs a scalar value indicating similarity. We use the same structure for all datasets and experiments. Note that λ in Eq. 9 will be different for the different frameworks because they use different similarity losses. However, for each framework, we use the same λ value for all datasets. Specifically, we fix λ to 0.0005 in MoCo-LoGo and 0.0001 as in SimSiam-LoGo. In practice, the λ is applied to the ratio of gradients between our regressor and backbone networks.

4.2. Training and Evaluating the Features

As a first set of experiments, we train the SSL models from scratch on CIFAR10 [24], STL10 [12], and ImageNet100 (IN-100) [34, 35] with different backbone networks to show that our strategy is robust to image sizes and datasets scale.

Evaluation. To evaluate the learned features on the respective validation sets, we use both a K-Nearest Neighbor (KNN) and a linear classifier. In the latter case, we train the linear classifier on the features extracted from the training set with the self-supervised pre-trained model. We train the classifier in the same way as in [9, 18]. Details of datasets and parameter will be included in the supplementary.

As can be seen from Tables 1, 2, 3, our LoGo strategy consistently improves the classification accuracy of the baseline for both KNN and linear classification. In Table 2, the performance of SimSiam drops significantly. This is because, unlike the average pooling in ResNets, the AlexNet used for this experiment relies on a fully connected layer outputting features in dimension 4096, which are difficult to handle by using cosine loss. Interestingly, our SimSiam-LoGo nonetheless performs as well as MoCo-LoGo, which implies that our regressor provides valuable information to the encoder.

4.3. Transfer Learning

One of the most important goals of representation learning is to obtain a backbone network extracting features that facilitate training on different datasets. We evaluate this on various datasets and downstream tasks. According to [9, 42], MoCo constitutes the state-of-the-art for transfer to other datasets and tasks. In this context, most methods use a ResNet-50 as the backbone, and we thus train by applying our LoGo strategy to MoCo, i.e., MoCo-LoGo. Table 4 shows that both MoCo and our MoCo-LoGo yield an improvement of around 6% when increasing the capacity of the backbone from ResNet-34 to ResNet-50. Importantly, the advantage of MoCo-LoGo over MoCo remains unchanged compared to our previous experiments.

To perform transfer learning, we, therefore, use our ResNet-50 pre-trained for 200 epochs. Below, we evidence the benefits of our approach for image recognition, object detection, and semantic segmentation using several datasets. For all the experiments in this section, we freeze the backbone networks and train the following task-dependent network modules according to the task at hand.

4.3.1 Image Recognition

Table 5 compares the results of our approach and of the baselines on different image recognition datasets using the linear evaluation protocol of [16, 23, 27]. Similarly to [27], we observed that the SSL strong augmentation methods and

	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech	Cars	Aircraft	DTD
Super(IN-100)	**86.16**	62.7	53.89	52.91	73.50	76.09	77.53	30.61	36.78	62.07
MoCo	83.71	60.59	58.21	57.54	64.30	85.56	74.12	32.63	46.23	60.64
MoCo+Aug*	85.26	**63.90**	60.78	63.36	73.46	85.70	78.93	37.35	39.47	66.22
MoCo-LoGo w/o L2L	85.19	61.47	63.66	65.45	71.74	90.2	77.91	37.22	48.21	65.74
MoCo-LoGo	86.09	63.43	**65.67**	**67.54**	**76.17**	**92.13**	**82.09**	**40.77**	**50.07**	**67.87**

Table 5. Transfer learning for image recognition. We report the recognition accuracy(%). Super(IN-100) denotes the same network as for SSL but trained on IN-100 with supervision. MoCo+Aug* [27] pre-trains the SSL encoder for 500 epochs, which is 300 epochs more than MoCo and our MoCo-LoGo. MoCo-LoGo w/o L2L indicates our model without local-to-local dissimilarity. We highlight the best results in **bold**.

long training epochs harm the supervised baseline model for transfer learning. Therefore, we use the standard supervised training setting and train the model for 100 epochs. Since the image style and semantic classes of CIFAR10 and CIFAR100 are very similar to IN-100, the performance of MoCo-LoGo is very close to the supervised counterpart and to MoCo+Aug*, which was pre-trained for 500 epochs. Our method significantly outperforms the other baselines methods, especially on fine-grained classification datasets, such as Flowers, Aircraft, Caltech, and Food. More information about the datasets can be found in the supplementary material. Altogether, these results show that our MoCo-LoGo enables the backbone to capture rich semantic information.

4.3.2 Dense Prediction

We use our SSL trained networks to conduct object detection experiments on both MS-COCO [28] and PASCAL VOC [14], as well as semantic segmentation experiments on MS-COCO [28]. We employ the Average precision (AP) to evaluate the results. Specifically, we use AP_{50} and AP, AP_{75}, where 50 and 75 indicate the that IoU threshold is set to 0.5 and 0.75, respectively, and AP denotes the average precision when the IoU threshold is varied from 0.5 to 1.0.

Setting. For these experiments, we follow one of the protocols in [18]. We adopt the pretrained ResNet 50-C4 as the backbone and fine-tune the Faster R-CNN [33] detector on both the VOC and COCO datasets. We apply $2\times$ schedulers on both datasets, which means that we train for approximately 23 epochs. In the VOC dataset, we use the 07+12 training sets to train the detector and the VOC *test2007* as test set; for COCO, we train on the *train2017* set (around 118k images) and evaluate on *val2017*.

As shown in the left portion of Table 6, our MoCo-LoGo achieves the best results in terms of AP and AP_{75} for the detection task. Note that these metrics are more strict than AP_{50}. On the segmentation task, our method achieves the best results in all three metrics. Remarkably, MoCo-LoGo also surpasses the supervised backbones trained on IN-1k in terms of the more strict metrics AP and AP_{75}. This indicates that a small number of images suffice for our strategy to capture important semantic information.

5. Ablation Study

In this section, we study the components introduced in our strategy to validate their functionality. Since stop-gradient, batch size and learning rate have been intensively studied by [9, 18], we will not discuss them here. We focus our analysis on our main contributions and on what the regressor learns.

5.1. Muti-crop and Similarity Loss

First, we remove the local-to-local dissimilarity term ℓ_a on the local patches to study the improvements contributed by our multiple crops and similarity loss used to encode global-to-global and local-to-global relationships. We use w/o L2L to denote this baseline.

We report the Top-1 KNN accuracy of w/o L2L and our full strategy in Table 7. Note that only performing multi-crop with similarity loss in the pretaining stage yields a better KNN accuracy than vanilla MoCo and SimSiam. However, there is still a gap with our full model. Consistent phenomena can be seen in Table 5, where we transfer the MoCo-LoGo w/o L2L pretrained model to other image recognition tasks. In some datasets, such as Pets and Caltech, it still falls behind our full strategy by a margin. This evidences the importance of encouraging dissimilarity between the local crops for fine-grained classification tasks.

5.2. Learnable Affinity Measure

To analyze what our regressor learns, we study the difference between employing our regressor or the cosine distance. We observed that maximizing the cosine distance between the local crops during training causes the model to collapse. In other words, when we use KNN to monitor the progress of training, the accuracy rapidly drops. To further analyze our learned measure, we compare the similarity of different images computed by using either the cosine distance or our regressor.

	Object Detection						Segmentation		
	VOC07+12			MS-COCO			MS-COCO		
	AP^{bb}_{50}	AP^{bb}	AP^{bb}_{75}	AP^{bb}_{50}	AP^{bb}	AP^{bb}_{75}	AP^{mk}_{50}	AP^{mk}	AP^{mk}_{75}
Super(IN-1k)	**81.30**	53.50	58.80	**59.90**	40.00	43.10	56.50	34.70	36.90
MoCo	78.65	52.43	57.22	59.11	39.38	42.55	55.83	34.52	36.84
MoCo-LoGo	81.12	**54.91**	**61.06**	59.74	**40.23**	**43.48**	**56.55**	**35.04**	**37.42**

Table 6. Transfer learning on object detection and semantic segmentation. Super (IN-1k) indicates supervised training on ImageNet-1k. MoCo and MoCo-LoGo are trained on IN-100. The best entries are shown in **bold**.

		CIFAR10	STL10	IN100
MoCo-LoGo	w/o L2L	82.31	74.26	69.00
	full	**84.44**	**76.79**	**76.82**
SimSiam-LoGo	w/o L2L	83.33	60.21	76.64
	full	**87.67**	**76.96**	**78.48**

Table 7. Comparison of the top-1 KNN classification accuracy (%) for the full LoGo strategy and LoGo without local-to-local strategy on both MoCo and SimSiam. The models are trained and tested on the same dataset.

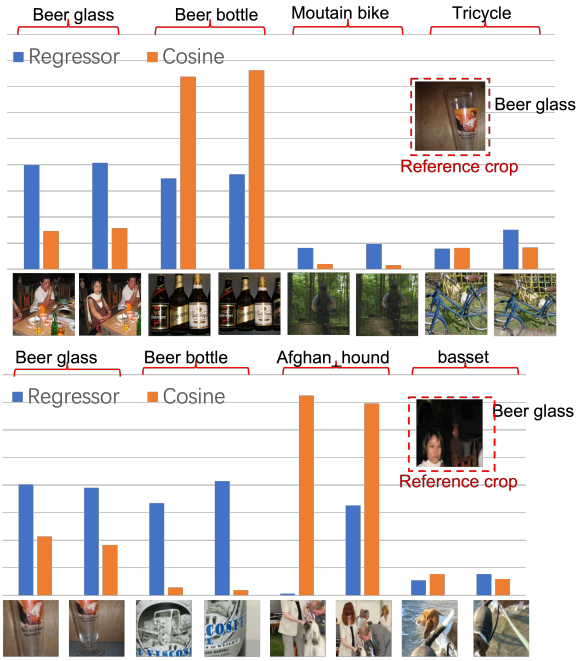

Figure 3. Comparison of our learned affinity measure and the cosine similarity. The values are the normalized regressor and cosine similarity between the reference crops and every crop on the x-axis. Higher values indicate a larger similarity.

To this end, we randomly obtain a crop from an image and compute both the cosine and regressor similarity with other 40 different crops (10 crops from 4 different images). In Figure 3, we visualize such similarities for 2 crops per class. As shown in the top portion of the figure, taking the beer glass crop as a reference, the cosine similarity wrongly assigns a larger similarity to the beer bottle than to its own class. By contrast, our regressor correctly preserves the beer glass information in such a complex scene, where the glass is not as salient as the people. In the lower portion of the figure, we show the similarities for a reference crop that depicts two different objects. In this case, the cosine distance fails and focuses on the person only. On the contrary, our learned affinity measure gives a higher similarity to the reference class and the semantically-close class. Interestingly, it yields very different values for the two different crops belonging to the Afghan hound class; a higher value for the crop where the person's face is visible, matching the fact that the reference image also contains a human face. More results are provided in the supplementary material. They further support the evidence that our learned measure encodes valuable semantic similarities between crops, thus providing stable supervision for the encoder.

6. Conclusion and Limitation

We have presented a new SSL strategy that leverages local and global views so as to better account for complex visual content. Our approach generalizes to existing SSL frameworks, consistently boosting their performance. Our learning strategy not only enables the global crops to preserve the invariant semantic information but allows the local crops to have diverse representations, thus not destroying their semantic meaning. Our extensive experiments have demonstrated the effectiveness of our strategy, further confirming the importance of every component in our approach. Our learnable affinity measure incurs more computation and parameters, however, it captures semantically-driven similarities between image patches and thus has the potential to be applied to other downstream tasks.

7. Acknowledgements

This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5−180359. C Qiu and W Ke were supported by National Natural Science Foundation of China under Grant No. 62006182.

References

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001. 2, 5

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 1, 3

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 2

[4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018. 5

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 1, 3

[8] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887, 2017. 3

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 3, 6, 7

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 3, 5

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2, 3, 4, 5

[12] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[16] 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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 3, 4, 6

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 1

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 3, 5, 6, 7

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[20] 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*, pages 770–778, 2016. 1

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[22] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *arXiv preprint arXiv:1709.02508*, 2017. 3

[23] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019. 6

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1

[27] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. 6, 7

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[29] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. 1

[30] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 1

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 7

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 3, 6

[36] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 3

[37] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016. 3

[38] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156, 2016. 3

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[40] Tong Zhang, Pan Ji, Mehrtash Harandi, Richard Hartley, and Ian Reid. Scalable deep k-subspace clustering. In *Asian Conference on Computer Vision*, pages 466–481. Springer, 2018. 3

[41] Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. In *International Conference on Machine Learning*, pages 7384–7393. PMLR, 2019. 3

[42] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 6