Intra-Instance VICReg: Bag of Self-Supervised Image Patch Embedding

Yubei Chen\textsuperscript{1*}, Adrien Bardes\textsuperscript{1,2*}, Zengyi Li\textsuperscript{3,4}, Yann LeCun\textsuperscript{1,5,6}
\textsuperscript{1} Meta AI
\textsuperscript{2} Inria, École normale supérieure, CNRS, PSL Research University
\textsuperscript{3} Redwood Center \textsuperscript{4} Physics Dept., UC Berkeley
\textsuperscript{5} Courant Institute \textsuperscript{6} Center for Data Science, New York University

Abstract

Recently, self-supervised learning (SSL) has achieved tremendous empirical advancements in learning image representation. However, our understanding and knowledge of the representation are still limited. This work shows that the success of the SOTA siamese-network-based SSL approaches is primarily based on learning a representation of image patches. Particularly, we show that when we learn a representation only for fixed-scale image patches and aggregate different patch representations linearly for an image (instance), it can achieve on par or even better results than the baseline methods on several benchmarks. Further, we show that the patch representation aggregation can also improve various SOTA baseline methods by a large margin. We also establish a formal connection between the SSL objective and the image patches co-occurrence statistics modeling, which supplements the prevailing invariance perspective. By visualizing the nearest neighbors of different image patches in the embedding space and projection space, we show that while the projection has more invariance, the embedding space tends to preserve more equivariance and locality. Finally, we propose a hypothesis for the future direction based on the discovery of this work.

1 Introduction

Self-supervised representation learning experienced tremendous advancements in the past few years in many fields. In terms of the quality of the learned feature, unsupervised learning has caught up with supervised learning or even surpassed the latter in many cases. This trend promises unparalleled scalability for data-driven machine learning in the future. One of the most successful paradigms in image self-supervised representation learning is based on instance-augmentation-invariant contrastive learning [Wu et al., 2018, Chen et al., 2020a,b]. This style of learning methods achieves the following general goal: 1) It brings the representation of two different views (augmentation) of the same instance (image) closer. 2) It keeps the representation informative of the input; in other words, avoids collapse. Several recent non-contrastive methods achieve competitive performance by explicitly achieving those two goals [Bardes et al., 2021, Li et al., 2022]. While we celebrate the empirical success of SSL in a wide range of benchmarks, our understanding and knowledge of this learning process are still very limited. In this work, we seek the principle behind the instance-based SSL methods and argue that the success largely comes from learning a representation of image patches based on their co-occurrence statistics in the images. To demonstrate this, we simplify the current SSL method to using a single crop scale to learn a representation of image patches of fixed size and establish a formal connection between our formulation and co-occurrence statistics modeling. The patch representation can be linearly aggregated (bag-of-words) to form the representation of the image. The learned representation achieves similar or better performance than the baseline representation, which is based on the entire image. In particular, even kNN classifier works surprisingly well with the
aggregated patch feature. These findings also resonate with recent works in supervised learning based on patch features [Brendel and Bethge 2019, Dosovitskiy et al. 2020, Trockman and Kolter 2022]. We also show that for baseline SSL methods pretrained with multi-scale crops, the whole-image representation is essentially an aggregation of different patch representations from the same instance. Further, given various SOTA baseline SSL models, we show that the same aggregation process can further improve the representation quality. Then we provide a cosine-similarity-based visualization of image patches representation on both ImageNet and CIFAR10 datasets. Particularly, we find that while the projection space has achieved significant invariance, the embedding space, frequently used for representation evaluation, tends to preserve more locality and equivariance.

Our discoveries may provide useful explanations and understanding for the success of the instance-augmentation-invariant SSL methods. The co-occurrence statistics modeling formulation and equivariance preserving property in the embedding space both supplement the current prevailing invariance perspective. Finally, these results motivate an interesting discussion of several potential future directions.

2 Related Works

2.1 Instance-Based Self-Supervised Learning: Invariance without Collapse

The instance contrastive learning [Wu et al., 2018] views each of the images as a different class and uses data augmentation [Dosovitskiy et al., 2016] to generate different views from the same image. As the number of classes is equal to the number of images, it is formulated as a massive classification problem, which may require a huge buffer or memory bank. Later, SimCLR [Chen et al. 2020a] simplifies the technique significantly and uses an InfoNCE-based formulation to restrict the classification within an individual batch. While it’s widely perceived that contrastive learning needs the “bag of tricks,” e.g., large batches, hyperparameter tuning, momentum encoding, memory queues, etc. Later works [Chen and He 2021, Yeh et al. 2021, HaoChen et al. 2021] show that many of these issues can be easily fixed. Recently, several even simpler non-contrastive learning methods [Bardes et al. 2021, Zbontar et al. 2021, Li et al. 2022] are proposed, where one directly pushes the representation of different views from the same instance closer while maintaining a non-collapsing representation space. Image SSL methods mostly differ in their means to achieve a non-collapsing solution. This include classification versus negative samples [Chen et al. 2020a], Siamese networks [He et al. 2020, Grill et al. 2020] and more recently, covariance regularization [Ermolov et al. 2021, Zbontar et al. 2021, Bardes et al. 2021, HaoChen et al. 2021, Li et al. 2022]. The covariance regularization has also long been used in many classical unsupervised learning methods [Roweis and Saul 2000, Tenenbaum et al. 2000, Wiskott and Sejnowski 2002, Chen et al. 2018], also to enforce a non-collapsing solution. In fact, there is a duality between the spectral contrastive loss [HaoChen et al. 2021] and the non-contrastive loss, which we prove in Appendix A.2.

All previously mentioned instance-based SSL methods pull together representations of different views of the same instance. Intuitively, the representation would eventually be invariant to the transformation that generates those views. We would like to provide further insight into this learning process: The learning objective can be understood as using the inner product to capture the co-occurrence statistics of those image patches. We also provide visualization to study whether the learned representation truly has this invariance property.

2.2 Patch-Based Representation

Many works have explored the effectiveness of patch-based image features. In the supervised setting, Bagnet [Brendel and Bethge 2018] and Thiry et al. [2021] showed that aggregation of patch-based features can achieve most of the performance of supervised learning on Image datasets. In the unsupervised setting, Giradis et al. [2020] performs SSL by requiring a bag-of-patches representation to be invariant between different views. Due to architectural constraints, Image Transformer based methods naturally use a patch-based representation [He et al. 2021, Bao et al. 2021].

2.3 Learning Representation by Modeling the Co-Occurrence Statistics

The use of word vector representation has a long history in NLP, which dates back to the 80s [Rumelhart et al. 1986, Dumais 2004]. Perhaps one of the most famous word embedding results, the
Figure 1: The pipeline of $I^2$ VICReg. From the same instance, fixed-size image patches are extracted, color-augmented, encoded to embedding and projection space. During training, different image patch projections from the same instance are pulled together while an anti-collapse regularization is applied. After training, different patch embeddings from the same instance are averaged to reach the image representation.

word vector arithmetic operation, was introduced in [Mikolov et al., 2013a]. Particularly, to learn this embedding, a task called “skip-gram” was used, where one uses the latent embedding of a word to predict the latent embedding of the word vectors in a context. A refinement was proposed in [Mikolov et al., 2013b], where a simplified variant of Noise Contrastive Estimation (NCE) was introduced for training the “Skip-gram” model. The task and loss are deeply connected to the SimCLR and its InfoNCE loss. Later, a matrix factorization formulation was proposed in [Pennington et al., 2014], which uses a carefully reprocessed concurrence matrix compared to latent semantic analysis. While the task in Word2Vec and SimCLR is relatively similar, the underlying interpretations are quite different. In instance-based SSL methods, one pervasive perception is that the encoding network is trying to build invariance, i.e., different views of the same instance shall be mapped to the same latent embedding. This work supplements this classical opinion and show that similar to Word2Vec, instance-based SSL methods can be understood as building a distributed representation of image patches by modeling the co-occurrence statistics.

Although there are image SSL methods inspired by word embedding learning [Gidaris et al., 2020] the proposed method still uses the invariance view and aims to learn the whole image feature while we focus on patch representations. Due to network architecture, many vision-transformer-based SSL methods also inherently learn a patch-based representation. For example, MAE [He et al., 2021] generates masked image patches conditioned on other image patches, and ImageBERT [Bao et al., 2021] predicts vector-quantized tokens based on nearby tokens in a context. Consistent with the co-occurrence interpretation, their success suggests that capturing correlation between image patches is fundamental to learning image representation.

3 Self-Supervised Image Patch Embedding and Co-Occurrence Statistics Modeling

As mentioned earlier, in contrast to the typical multi-scale augmentation used in the instance-based SSL methods, we use fixed-scale crops to learn a representation for fixed-size image patches. We show that any SSL objective can be used, which will be shown in Section 4 as long as they learn a non-collapsed representation where different image patches from the same context are close in the projection space, as shown in Figure [1]. In this work we mostly use covariance regularization based techniques [Bardes et al., 2021, Zhontar et al., 2021, Li et al., 2022, HaoChen et al., 2021], for which we present a general formulation:

**Definition 1.** Intra-instance variance-invariance-covariance regularization ($I^2$ VICReg):

$$\min_{\theta} - E_{p(x_1, x_2)} \left[ z_1^T z_2 \right], \text{ s.t. } E_{p(x)} \left[ zz^T \right] = \frac{1}{d_{emb}} \cdot 1$$

where $z = g(h)$ and $h = f(x; \theta)$. We call $h$ the embedding and $z$ the projection of an image patch, $x$. \{$x$\} all have the same size. The parametric function $f(\cdot; \theta)$ is a deep neural network with parameters

1In this work, the context refers to an image. But context could be generalized in straightforward ways.
θ, and g is typically a much simpler neural network with only one or a few fully connected layers. $d_{emb}$ is the dimension of an embedding vector, z. This general idea is shown in Figure 1. For an image, we extract fix-size image patches, which are color augmented before embedding f and projection g. Given an image patch $x_i$, which is in the red dash box in Figure 1 the objective tries to make its projection $z_i$ invariant to the projections of the other image patches within the instance. Further, the regularization tries to decorrelate different projection dimensions of $z$ while maintaining the variance of each dimension. VICReg was proposed in Bardes et al. [2021], and one concrete example of such VICReg objective is the following soft-constrained loss function proposed in [Li et al., 2022]:

$$\min_{\theta} \mathbb{E} \left[ -\operatorname{Tr} (Z_1^T Z_2) + \lambda \log \det \left( I + \frac{d_{emb}}{2B\epsilon^2} ZZ^T \right) \right]$$

(2)

where $Z = [Z_1, Z_2]$, each column of $Z_1, B$ is the batch size, $\epsilon$ is a chosen such that $\epsilon^2 \ll 1$. In this objective function, covariance regularization is achieved by maximizing the Total Coding Rate (TCR) [Ma et al., 2007].

Relationship to Co-Occurrence Statistics Modeling. Assume $x_1$ and $x_2$ are two color-augmented patches sampled from the same image. We denote their marginal distribution by $p(x_1)$ and $p(x_2)$, which includes variation due to sampling different locations within an image, random color augmentation, as well as variation due to sampling images from the dataset. We also denote their joint distribution by $p(x_1, x_2)$, which assume $x_1$ and $x_2$ are sampled from the same image. We show that contrastive learning can be understood by the following objective that approximates the normalized co-occurrence statistics by the inner product of the two embeddings $z_1$ and $z_2$ generated by $x_1$ and $x_2$:

$$\min \int p(x_1)p(x_2) \left[ wz_1^T z_2 - \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right]^2 dx_1 dx_2$$

(3)

where $w$ is a fixed weight used to compensate for scale differences.

Proposition 3.1. The above optimization problem can be rewritten as the following spectral contrastive form:

$$\min \mathbb{E}_{p(x_1, x_2)} \left[ -z_1^T z_2 \right] + \lambda \mathbb{E}_{p(x_1)p(x_2)} \left( z_1^T z_2 \right)^2$$

(4)

where $\lambda = \frac{w}{2}$. The proof is rather straightforward and is presented in Appendix A.1. As we can see that the first term resembles the similarity term in Eqn 1 and the second spectral contrastive term HaoChen et al. [2021] minimizes the inner product between two independent patch embeddings, which has the effect of orthogonalizing them. As we mentioned earlier, there exists a duality between the spectral contrastive regularization and covariance regularization term in Eqn 1. Please refer to the Appendix A.2 for a more in-depth discussion.

Bag-of-Feature Model. After we have learned an embedding for the fix-scale image patches, we can embed all of the image patches $\{x_{11}, \ldots, x_{HW}\}$ within an instance into the embedding space, $\{h_{11}, \ldots, h_{HW}\}$. Then the whole-image representation $R_{img}$ is a linear aggregation of all the patches’ embedding, as shown in Figure 1. Depending on the size of the image patches, aggregating a small subset of the patches from the same instance may suffice in practice. E.g., for scale = 0.2, we find 16 patches aggregation achieves similar performance compared to aggregating all of the patches. We may also aggregate the projections to get the whole-image representation, but the embedding typically contains more equivariance and locality, which leads to better performance. We will show this result in Section 5.

4 Quantitative Empirical Results

Through experiments, we demonstrate that representations learned by self-supervised learning method trained with fixed-size patches are nearly as strong as that learned with multi-scale crops. For several cases, pretraining with multi-scale crops and evaluating on the fixed central crop is equivalent in terms of performance to pretraining with fixed-size small patches and evaluating by averaging the embedding across the image. We further show that for a multi-scale pretrained model, averaging
Table 1: **Performance on CIFAR-10 for patch-based and standard self-supervised pretraining methods.** We evaluate the performance of a linear classifier for various pretraining methods, both with the *Patch-based training*, where patches of scale 0.2 are sampled during pretraining, and *Standard training*, where the patch scale is uniformly sampled between scale 0.08 and 1.0 during pretraining. The ‘Central’ evaluation is the standard evaluation protocol where the linear classifier is trained and evaluated on single fixed central patches of the image, which is the whole image for CIFAR dataset. For the n-patch evaluation, the classifier is trained and evaluated on the linearly-aggregated embedding of n patches, sampled with the same scale factor as during pretraining. Scale 0.2 and 0.08 correspond to $14 \times 14$ and $9 \times 9$ image patches respectively.

| Method | Central | 1 patch | 16 patches | 256 patches | Central | 1 patch | 16 patches | 256 patches |
|--------|---------|---------|------------|-------------|---------|---------|------------|-------------|
| TCR    | 46.0    | 82.2    | 90.4       | 90.8        | 90.1    | 86.5    | 91.5       | 91.8        |
| VICReg | 47.1    | 83.1    | 90.9       | 91.2        | 90.7    | 87.3    | 91.9       | 92.0        |
| BYOL   | 47.3    | 83.6    | 91.3       | 91.5        | 90.9    | 87.8    | 92.3       | 92.4        |

embedding of fixed-scale small image patches converges to the embedding generated by the center cropped image, as the number of aggregated patches increases. Thus, the standard practice of using multi-scale pretraining and center crop evaluation can be viewed as an efficient way to obtain the averaged patch embeddings. Further, we show that the patch aggregation evaluation can further improve the representation of the baseline models by a significant margin. Our experiments used the CIFAR-10, CIFAR-100, and the more challenging ImageNet-100 dataset. We also provide a short-epoch ImageNet pretraining to show that with small image patches, the training tends to have lower learning efficiency. In the last section, we will dive into the invariance and equivariance analysis of the patch embedding.

4.1 CIFAR

We first provide experimental results on the standard CIFAR-10 and CIFAR-100 datasets [Krizhevsky et al., 2009], which contain 10 and 100 classes, respectively. Both contain 50000 training and 10000 testing images. The results are shown in Figure 2, Tables 1 and 2. We show results obtained using the linear evaluation protocol and the kNN evaluation protocol which give consistent results with regard to each other. The evaluation is conducted in two different ways. The standard evaluation method generates the embedding using the full image, both during training of the linear classifier and at final evaluation (*Central* in the Figures and the tables). Alternatively, an image embedding is generated by inputting a certain number of patches (same scale as training time and upsampled) into the neural network and aggregating the patch embeddings by performing averaging. This is denoted as 1, 16, and 256 patches in the Figure.

The main observation we make is that pretraining on small patches and evaluating with the averaged embedding performs as well as or better than pretraining with random-scale patches and evaluating with the full image representation. On CIFAR-10 with the TCR method, the 256-patches evaluation with pretraining fixed-scale of 0.2 outperforms the full-image evaluation with random pretraining scale between 0.08 and 1, which is the standard scale range used. When only averaging 16-patches, the same model performs on par with full image evaluation. On the k-NN evaluation, pretraining with random-scale patches not spanning the full range 0.08 to 1.0 gives much worse performance comparatively, than linear evaluation. However, aggregated embedding does not see this comparatively worse performance, and can still outperform the full image evaluation. Using results from Table 1,2 and 3 we can draw the same conclusion on other datasets and other self-supervised methods (VICReg [Bardes et al., 2021] and BYOL [Grill et al., 2020]).

Implementation Details. For all the experiments, we pretrain a ResNet-34 for 600 epochs. We use a batch size of 1024, LARS optimizer, and a weight decay of $1e^{-0.4}$. The learning rate is set to 0.3, and follows a cosine decay schedule, with 10 epochs of warmup and a final value of 0. In the TCR loss, $\lambda$ is set to 30.0, and $\epsilon$ is set to 0.2. The projector network consists of 2 linear layers with respectively 4096 hidden units and 128 output units for the CIFAR-10 experiments and 512 output units for the CIFAR-100 experiments. All the layers are separated with a ReLU and a BatchNorm layers. The data augmentations used are identical to those of BYOL.
Figure 2: **Evaluation on CIFAR-10 for various RandomResizedCrop scales.** We evaluate the performance of a linear classifier (a) and a k-NN classifier (b) for pretraining with various patch sizes and various evaluation setups. During pretraining, the patches are sampled using RandomResizedCrop(scale, scale) for single values, and RandomResizedCrop(min_scale, max_scale) for scale values uniformly from min_scale to max_scale. The “Central” evaluation is the standard evaluation protocol where the classifier is trained and evaluated on single fixed central patches of the image, which is the entire image for CIFAR-10. For the $n$ patch evaluation, the classifier is trained and evaluated on the linearly-aggregated embedding of $n$ patches, sampled with the same scale factor as during pretraining. Scale 0.08, 0.1, 0.13, 0.2, 0.25 correspond to $9 \times 9$, $10 \times 10$, $13 \times 13$, $14 \times 14$, $16 \times 16$ image patches respectively. Please note that it is expected that “central” evaluation performs poorly on fix-scale pretraining as the model has never seen the entire image during pretraining.

Table 2: **Performance on CIFAR-100 for patch-based and standard self-supervised pretraining methods.** We evaluate the performance of a linear classifier for various pretraining methods, both with the Patch-based training, where patches of scale 0.2 are sampled, and Standard training, where the patch scale is uniformly sampled between scale 0.08 and 1.0. Scale 0.2 and 0.08 correspond to $14 \times 14$ and $9 \times 9$ image patches respectively.

| Method | Central 1 patch | Central 16 patches | Central 256 patches | Standard 1 patch | Standard 16 patches | Standard 256 patches |
|--------|-----------------|--------------------|---------------------|------------------|---------------------|---------------------|
| TCR    | 34.6            | 59.2               | 67.1                | 67.3             | 66.8                | 60.5                |
| VICReg | 35.5            | 60.1               | 68.0                | 68.3             | 67.6                | 61.4                |
| BYOL   | 37.4            | 60.9               | 68.9                | 69.2             | 68.8                | 62.3                |

4.2 **ImageNet-100 and ImageNet**

We provide experimental results on the ImageNet-100 and ImageNet dataset [Deng et al., 2009]. We present our results using the linear evaluation protocol in Table 3 and Figure 3. The behavior observed on CIFAR-10 generalizes to ImageNet-100. Averaging embeddings of 16 small patches produced by the patch-based pretrained model performs almost as well as the “central” evaluation of the embedding produced by the baseline model on the ImageNet-100 dataset, as shown in Table 3. In Figure 3(b), we show short-epoch pretrained models on ImageNet. As the patch-based pretrained model tends to see much less information compared to the baseline multi-scale pretraining, there is a 4.5% gap between the patch-based model and the baseline model.

**Implementation Details.** For all the experiments, we pretrain a ResNet-50 with the TCR loss for 400 epochs for ImageNet-100, and 100 epochs for ImageNet. We use a batch size of 1024, the LARS optimizer, and a weight decay of $1 \times 10^{-4}$. The learning rate is set to 0.1, and follows a cosine decay schedule, with 10 epochs of warmup and a final value of 0. In the TCR loss, $\lambda$ is set to 1920.0, and $\epsilon$ is set to 0.2. The projector network consists of 3 linear layers with each and 8192 units, separated by a ReLU and a BatchNorm layers. The data augmentations used are identical to those of BYOL.

4.3 **Patched-Aggregation Based Evaluation with Multi-Scale Pretrained Model**

Our results in the last section show that the best performance is obtained when the pretraining step is done using patches of various sizes, and the evaluation step is done using the aggregated
Table 3: Performance on ImageNet-100 with Patch-based and standard self-supervised pre-training methods. We evaluate the performance of a linear classifier with $I^2$ VICReg-TCR, both with the Patch-based training, where patches of scale 0.2 are sampled during pretraining, and Standard training, where the patch scale is uniformly sampled between scale 0.08 and 1.0. Scale 0.2 and 0.08 correspond to $100 \times 100$ and $64 \times 64$ image patches respectively.

| Method | Patch-based training | Standard training |
|--------|----------------------|-------------------|
|        | Central 1 patch 16 patches 48 patches | Central 1 patch 16 patches 48 patches |
| TCR    | 41.3 45.6 76.1 76.3 77.3 70.1 78.5 78.8 |

Figure 3: (a) Patch embedding convergence to the instance embedding. For a baseline multi-scale pretrained VICReg model, we show that the patch embedding aggregation converges to the whole-image embedding as the number of aggregated patches increases. (b) Linear evaluation on ImageNet for various RandomResizedCrop scales. (a) Evolution of the cosine similarity between the aggregation of $N$ embeddings of patches and the instance embedding which is the aggregation of all possible patches in the image. (b) Evaluation of the performance of a linear classifier for various pretraining patch sizes, on Central, 1 and 16 patches evaluation setups. Scale 0.02, 0.08, 0.2 and 1.0 correspond to $32 \times 32$, $64 \times 64$, $100 \times 100$ and $224 \times 224$ image patches respectively.

4.4 Convergence of Patch-Based Embedding to Whole-Instance Embedding.

In this experiment, we show that for a multi-scale pretrained SSL model, linearly aggregating the patch embedding converges to the instance embedding. We take a multi-scale pretrained VICReg baseline model and use randomly selected 512 images from the ImageNet dataset. For each image, we first get the embedding of the $224 \times 224$ center crop. Then we randomly aggregate $N$ embeddings of different $100 \times 100$ image patches and calculate the cosine similarity between the patch-aggregated embedding and the center crop embedding. Figure 3(a) shows that the aggregated representation converges to the instance embedding as $N$ increases from 1 to 16 to all the image patches.

5 Patch Embedding Visualization: Invariance or Equivariance?

The instance-augmentation-invariant SSL methods are primarily motivated from an invariance perspective. In this section, we provide CIFAR-10 nearest neighbor and ImageNet cosine-similarity

3“All”: extracting overlapped patches with stride 4 and totally aggregate about 1000 patches’ embeddings.
Table 4: Linear Evaluation with aggregated embedding on ImageNet with models trained with state-of-the-art SSL methods. Using aggregated embedding outperforms embedding from the center crop. Central: embedding from the center cropped image is used in training and testing using the standard linear evaluation protocol. 1, 16, and 48 patches: The linear classifier is trained and evaluated on the aggregated embedding of 1, 16, and 48 patches respectively, sampled with the same scale factor range as during pretraining (0.08, 1.0).

| Method  | Central | 1 patch | 16 patches | 48 patches |
|---------|---------|---------|------------|------------|
| VICReg  | 73.2    | 57.6    | 74.2       | 74.4       |
| BYOL    | 74.3    | 59.3    | 75.4       | 75.6       |
| SwAV    | 75.3    | 60.8    | 75.9       | 76.0       |

Figure 4: Visualization of kNN in the projection space and the embedding space for CIFAR10. Distance is calculated by cosine similarity. Query patch is in the top left corner encircled by red dash, green box indicates patches from other image of the same class. Patches without surrounding box is from the same image as the query. While the nearest neighbors are both from same-category instances, we can see that the embedding space tends to preserve the local part information, whereas the projection space may collapse different parts of the same category.

Heatmap visualization to further understand the learned representation. In the CIFAR-10 experiment, we take a model pre-trained with $14 \times 14$ image patches on CIFAR-10 and calculate the projection and embedding vectors of all different image patches from the training set. Then for a given $14 \times 14$ image patch (e.g. the ones circled by red dash boxes Fig. 4), we visualize its $k$ nearest neighbors in terms of cosine-similarity in both the projection and the embedding space. Figure 4 shows the results for two different image patches. The patches circled by green boxes are image patches from another instance of the same category, whereas the uncircled patches are from the same instance.

In the ImageNet experiment, we take a multi-scale pretrained VICReg model, then for a given image patch (e.g. circled by red dash boxes in Figure 5), we visualize the cosine-similarity between embedding from this patch and that from the other patches from the same instance. In this experiment, we use two different image patches scales, $71 \times 71$ and $100 \times 100$. The heatmap visualization is normalized to the same scale.

Overall, we observe that the projection vectors are significantly more invariant than the embedding vectors. This is apparent from both Figure 4 and Figure 5. For the CIFAR kNN patches, NNs in the embedding space are visually much more similar than NNs in the projection space. In fact, in the embedding space, the nearest NNs are mostly locally shifted patches of similar “part” information. For projection space, however, many NNs are patches of different “part” information from the same class. E.g., we can see in Figure 4 that an NNs of a “wheel” in the projection space might be a “door” or a “window”, however, the NNs in the embedding space all contain “wheel” information. In the second example, the NNs of a “horse legs” patch may have different “horse” body parts whereas the NNs in the embedding space are all “horse leg”.

The heatmap visualization on ImageNet also illustrates the same phenomenon. Let’s visualize a multi-scale pretrained VICReg model. The projection vector from a patch has a high similarity to that from the query patch whenever the patch has enough information to infer the class of the image.
Figure 5: Visualization of cosine similarity in the projection space and the embedding space. Query patch is indicated by red dash. Projection and Embedding cosine-similarity heatmaps use the same color scaling. The projection vectors are significantly more invariant compared to the embedding ones, and the embedding space contains localized information that is shared among similar patches, when the size of the patches is small enough. We can see that the embedding space tends to preserve more locality compared to the projection space.

While for embedding vectors, the similarity area is much more localized to the query patch, or to other patches with similar features (the other leg of the dog in Figure 5). This general observation is consistent with the results of the visualizations in Bordes et al. [2021]. We slightly abused the term and call this property of the embedding vector equivariant, in contrast to the invariance possessed by the projector vectors. A more thorough visualization is provided in the Appendix A.4.

6 Discussion

In this paper, we seek to provide an understanding of the success of instance-augmentation-invariant SSL methods. We demonstrate learning an embedding for fixed-size image patches (I\(^2\) VICReg) and linear aggregating them from the same instance can achieve on-par or even better performance than the multi-scale pretraining. On the other hand, with a multi-scale pretrained model, we show that the whole image embedding is essentially the average of patch embeddings. Conceptually we establish the close connection between I\(^2\) VICReg and modeling the co-occurrence statistics of patches.

Through visualizing nearest neighbors and cosine-similarity heatmaps, we find that the projector vector is relatively invariant while the embedding vector is instead equivariant, which may explain its higher discriminative performance. This result suggests that the SSL objective, which learns the co-occurrence statistics, encourages an invariant solution, while the more favorable property of equivariance is achieved by the implicit bias introduced by the projector. In the future, it is interesting to explore if it’s possible to directly encourage equivariance in the objective function in a more principled manner instead of relying on the projector head. For this, prior works in NLP may provide useful guidance. In Pennington et al. [2014], word embedding is learned by fitting the log co-occurrence matrix, which avoids the problem of getting dominated by large elements and allows the embedding to carry richer information. Similarly, an SSL objective that implicitly fits to the log-occurrence matrix may learn a more equivariant embedding, which may be an interesting direction for future work.

Lots of open questions still remain in the quest of understanding image SSL. For example, it’s still unclear why the projector \(g\) makes the embedding \(h\) more equivariant than the projection \(z\). For this, we hypothesize that the role of the projector can be understood as learning a feature representation for a kernel function in the embedding space. Since for \(h_1, h_2\), the dot product of \(g(h_1)\) and \(g(h_2)\) always represent some positive semi-definite kernel on the original space \(k(h_1, h_2) = g(h_1)^T g(h_2)\). It is possible that the flexible kernel function on the embedding alleviates the excess invariance problem caused by the objective on the projector vectors, which allows the embedding to be more equivariant and perform better. We leave further analysis of this hypothesis to future work.
References

H. Bao, L. Dong, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

F. Bordes, R. Balestriero, and P. Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2018.

W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Y. Chen, D. Paiton, and B. Olshausen. The sparse manifold transform. *Advances in neural information processing systems*, 31, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. A. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1734–1747, 2016.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology (ARIST)*, 38:189–230, 2004.

A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

Q. Garrido, Y. Chen, A. Bardes, L. Najman, and Y. Lecun. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022.

S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord. Learning representations by predicting bags of visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6928–6938, 2020.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021.

K. He, H. Fan, Y. Wu, S. Xie, and R. 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.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

A. Krizhevsky, G. Hinton, and et al. Learning multiple layers of features from tiny images. 2009.
Q. Le, A. Karpenko, J. Ngiam, and A. Ng. Ica with reconstruction cost for efficient overcomplete feature learning. *Advances in neural information processing systems*, 24, 2011.

Z. Li, Y. Chen, Y. LeCun, and F. T. Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022.

Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1546–1562, 2007.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

L. Thiry, M. Arbel, E. Belilovsky, and E. Oyallon. The unreasonable effectiveness of patches in deep convolutional kernels methods. *arXiv preprint arXiv:2101.07528*, 2021.

A. Trockman and J. Z. Koltzer. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

C.-H. Yeh, C.-Y. Hong, Y.-C. Hsu, T.-L. Liu, Y. Chen, and Y. LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.
A Appendix

A.1 Proof of Proposition 3.1

Proposition A.1. Equation \[ \text{Equation 3} \] can be rewritten in the following contrastive form:
\[
\mathbb{E}_{p(x_1, x_2)} \left[ -z_1^T z_2 \right] + \lambda \mathbb{E}_{p(x_1)p(x_2)} \left( z_1^T z_2 \right)^2
\]  
(5)
where \( \lambda = \frac{w}{2} \).

Proof. Since we are dealing with an objective, we can drop constants, which do not depend on the embedding \( z_1 \) and \( z_2 \), when they occur.
\[
L = \int p(x_1)p(x_2) \left[ wz_1^T z_2 - \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right]^2 dx_1 dx_2 \quad (6)
\]
\[
= \int p(x_1)p(x_2) \left[ (wz_1^T z_2)^2 - 2wz_1^T z_2 \cdot \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right] dx_1 dx_2 \quad (7)
\]
\[
= \int p(x_1)p(x_2) (wz_1^T z_2)^2 dx_1 dx_2 - 2w \int p(x_1, x_2) (z_1^T z_2) dx_1 dx_2 \quad (8)
\]
\[
= \mathbb{E}_{p(x_1, x_2)} \left[ -z_1^T z_2 \right] + \lambda \mathbb{E}_{p(x_1)p(x_2)} \left( z_1^T z_2 \right)^2 \quad (9)
\]
where \( \lambda = \frac{w}{2} \).
\[ \square \]

A.2 The duality between spectral contrastive regularization and covariance regularization.

For Objective \[ \text{Objective 2} \] and Objective \[ \text{Objective 4} \] as the similarity term is the same, we can focus our discussion on the regularization term, particularly with SGD optimizer. For simplicity, we assume that the embedding \( z \) is L2-normalized and each of the embedding dimension also has zero mean and normalized variance. Given a minibatch with size \( N \), the spectral regularization term \( \mathbb{E}_{p(x_1)p(x_2)} \left( z_1^T z_2 \right)^2 \) reduces to \( \|Z^T Z - I_d\|_F^2 \). By Lemma 3.2 from Le et al. [2011], we have:
\[
\|Z^T Z - I_N\|_F^2 = \|ZZ^T - I_d\|_F^2 = \|ZZ^T - \frac{N}{d} I_d\|_F^2 + C \quad (10)
\]
where \( C \) is a constant. The third equality follows due to that each of the embedding dimension is normalized. \( \|ZZ^T - \frac{1}{d} I_N\|_F^2 \) is the mini-batch version of the covariance regularization term \( \mathbb{E}_{p(x)} \left[ z z^T \right] = \frac{N}{d_{x_{\text{mb}}}} \cdot I \).

A thorough discussion is beyond the scope of this work. We refer the curious readers to Garrido et al. [2022] for a more general discussion on the duality between contrastive learning and non-contrastive learning.

A.3 ImageNet Intra-Instance Visualization

In this section, we provide further visualization of the multi-scale pretrained VICReg model, and the results are shown in Fig 6. Here we use image patches of scale 0.1 to calculate the cosine similarity heatmaps, the query patch is marked by the red-dash boxes. The embedding space contains more localized information, whereas the projection space is relatively more invariant, especially when the patch has enough information to determine the category.

A.4 CIFAR10 kNN Visualization

This section continues the visualization of the model pretrained with 14 × 14 patches. In this visualization, we primarily use kNN and cosine similarity to find the closest neighbors for the query patches, marked in the red-dash boxes. Again, green boxes indicate that the patches are from other instances of the same category; red boxes indicate that the patches are from other instances of a different category. Patchs that do not have a color box are from the same instance. In the following, we discuss several interesting aspects of the problem.
Additional Projection and Embedding Spaces Comparison. As we can see in Figure 7, the embedding space has a much lesser degree of collapse of the semantic information. The projection space tends to collapse different “parts” of a class to similar vectors, whereas the embedding space preserves more information about the details in a patch. This is manifested by higher visual similarity between neighboring patches.

Embedding Space with 256 kNN. In the previous CIFAR visualization, we only show kNN with 119 neighbors. In Figure 8 and Figure 9, we provide kNN with 255 neighbors, the same set of conclusions hold.

Different “Parts” in the Embedding Space. In Figure 10, we provide some more typical patches of “parts” and show their embedding neighbors. While many parts are shared by different instances, we also find some less ideal cases, e.g. Figure 10(4a)(2d), where the closest neighbors are nearly all from the same instance.

As we discussed earlier, the objective is essentially modeling the co-occurrence statistics of patches. If the same patch is not “shared” by different instances, it is relatively uninformative. While the exact same patch might not be “shared”, the color augmentation and deep image prior embedded in the network design may create approximate sharing. In Figure 11 and Figure 12, we provide two examples of the compositional structure of instances.
Figure 6: More visualization of cosine similarity heatmaps in the projection space and the embedding space. Here the query patch is marked by the red-dash boxes and its size is $71 \times 71$ and the instance image size is $224 \times 224$. 
Figure 7: Additional comparison between the projection space and the embedding space.
Figure 8: kNN in the embedding Space with 255 neighbors.
Figure 9: kNN in the embedding Space with 255 neighbors.
Figure 10: Different “parts” in the embedding space.
Figure 11: **The compositional structure of an airplane.** The “sky” part is shared by ships, birds, etc. The “wing” resembles the silhouettes of ships and is also shared by flying birds. The airscrew part is primarily shared by the other airplanes.

Figure 12: **The compositional structure of a horse.** The bottom left corner contains “shadow”, and the similar shadows are shared by deers and dogs. The bottom right part contains “legs”, which are also shared by deers and dogs. However, from the back to the thigh is shared by primarily other horses.