EQUIVARIANT SELF-SUPERVISED LEARNING: ENCOURAGING EQUIVARIANCE IN REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In state-of-the-art self-supervised learning (SSL) pre-training produces semantically good representations by encouraging them to be invariant under meaningful transformations prescribed from human knowledge. In fact, the property of invariance is a trivial instance of a broader class called equivariance, which can be intuitively understood as the property that representations transform according to the way the inputs transform. Here, we show that rather than using only invariance, pre-training that encourages non-trivial equivariance to some transformations, while maintaining invariance to other transformations, can be used to improve the semantic quality of representations. Specifically, we extend popular SSL methods to a more general fremework which we name Equivariant Self-Supervised Learning (E-SSL). In E-SSL, a simple additional pre-training objective encourages equivariance by predicting the transformations applied to the input. We demonstrate E-SSL's effectiveness empirically on several popular computer vision benchmarks. Furthermore, we demonstrate usefulness of E-SSL for applications beyond computer vision; in particular, we show its utility on regression problems in photonics science.

1 INTRODUCTION

Human knowledge about what makes a good representation and the abundance of unlabeled data has enabled the learning of useful representations via self-supervised learning (SSL) pretext tasks. State-of-the-art SSL methods encourage the representations not to contain information about the way the inputs are transformed, i.e. to be invariant to a set of manually chosen transformations. One such method is contrastive learning, which sets up a binary classification problem to learn invariant features. Given a set of datapoints (say images), different transformations of the same data point constitute positive examples, whereas transformations of other data points constitute the negatives (He et al., 2020; Chen et al., 2020). Beyond contrastive learning, many SSL methods also rely on learning representations by encouraging invariance (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Zbontar et al., 2021). In this paper, we refer to such methods as Invariant-SSL (I-SSL).

The natural question in I-SSL is to what transformations should the representations be insensitive (Chen et al., 2020; Tian et al., 2020; Xiao et al., 2020). Chen et al. (2020) highlighted the importance of transformations and empirically evaluated which transformations are useful for contrastive learning (e.g., see Figure 5 in their paper). Some transformations, such as *four-fold rotations*, despite preserving semantic information, were shown to be harmful for contrastive learning. This does not mean that four-fold rotations are not useful for I-SSL at all. In fact, predicting four-fold rotations is a good proxy task for evaluating the representations produced with contrastive learning (Reed et al., 2021). Furthermore, instead of being insensitive to rotations (invariance), training a neural network to predict them, i.e. to be *sensitive* to four-fold rotations, results in good image representations (Gidaris et al., 2018). These results indicate that the choice of making features *sensitive* or *insensitive* to a particular group of transformations can have a substantial effect on the performance of downstream tasks. However, the prior work in SSL has exclusively focused on being either entirely insensitive (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Zbontar et al., 2021) or sensitive (Agrawal et al., 2015; Doersch et al., 2015; Zhang et al., 2016; Noroozi & Favaro, 2016; Gidaris et al., 2018) to a set of transformations. In particular, the I-SSL literature has proposed to simply remove transformations that hurt performance when applied as invariance.

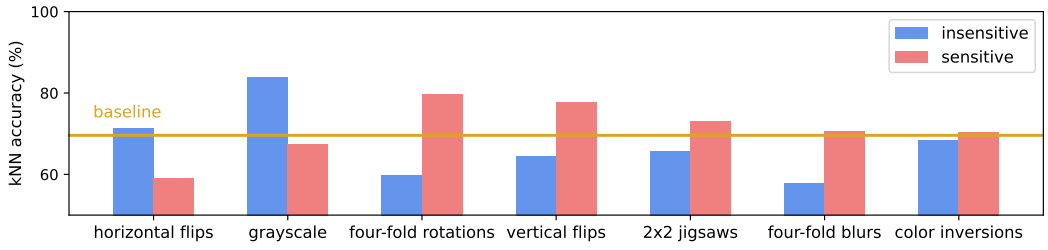

Figure 1: SSL representations should be encouraged to be either insensitive or sensitive to transformations. The baseline is SimCLR with random resized cropping only. Each transformation on the horizontal axis is combined with random resized cropping. The dataset is CIFAR-10 and the kNN accuracy is on the test set. More experimental details can be found in Section 3.

To understand how sensitivity/ insensitivity to a particular transformation affects the resulting features, we ran a series of experiments summarized in Figure 1. We trained and tested a simple I-SSL baseline, SimCLR (Chen et al., 2020), on CIFAR-10 using only the *random resized cropping transformation* (solid yellow line). The test accuracy is calculated as the retrieval accuracy of a k-nearest neighbors (kNN) classifier with a memory bank consisting of the representations on the training set obtained after pre-training for 800 epochs. Next, in addition to being invariant to resized cropping, we additionally encouraged the model to be either sensitive (shown in pink) or insensitive (shown in blue) to a second transformation. We varied the choice of this second transformation. We found that for some transformations, such as *horizontal flips* and *grayscale*, insenstivity results in better features, but is detrimental for transformations, such as *four-fold rotations*, *vertical flips*, *2x2 jigsaws* ($4! = 24$ classes), *four-fold Gaussian blurs* (4 levels of blurring) and *color inversions*. When we encourage sensitivity to these transformations, the trend is reversed. In summary, we observe that if invariance to a particular transformation hurts feature learning, then imposing sensitivity to the same transformation may improve performance. This leads us to conjecture that instead of choosing the features to be only invariant or only sensitive as done in prior work, it may be possible to learn better features by imposing invariance to certain transformations (e.g., cropping) and sensitivity to other transformations (e.g., four-fold transformations).

The concepts of sensitivity and insensitivity are both captured by the mathematical idea of equivariance (Agrawal et al., 2015; Jayaraman & Grauman, 2015; Cohen & Welling, 2016). Let G be a group of transformations. For any $g \in G$ let $T_g(\boldsymbol{x})$ denote the function with which g transforms an input image $\boldsymbol{x}$. For instance, if G is the group of four-fold rotations then $T_g(\boldsymbol{x})$ rotates the image $\boldsymbol{x}$ by a multiple of $\pi/2$. Let f be the encoder network that computes feature representation, $f(\boldsymbol{x})$. I-SSL encourages the property of "invariance to G," which states $f(T_g(\boldsymbol{x})) = f(\boldsymbol{x})$, i.e. the output representation, $f(\boldsymbol{x})$, does not vary with T_g. Equivariance, a generalization of invariance, is defined as, $\forall \boldsymbol{x} : f(T_g(\boldsymbol{x})) = T'_g(f(\boldsymbol{x}))$, where T'_g is a fixed transformation (i.e., without any parameters). Intuitively, equivariance encourages the feature representation to change in a well defined manner to the transformation applied to the input. Thus, invariance is a trivial instance of equivariance, where T'_g is the identity function, i.e. $T'_g(f(\boldsymbol{x})) = f(\boldsymbol{x})$. While there are many possible choices for T'_g (Cohen & Welling, 2016; Bronstein et al., 2021), I-SSL uses only the trivial choice that encourages f to be insensitive to G. In contrast, if T'_g is not the identity, then f will be sensitive to G and we say that the "equivariance to G" will be non-trivial.

Therefore, in order to encourage potentially more useful equivariance properties, we generalize SSL to an Equivariant Self-Supervised Learning (E-SSL) framework. In our experiments on standard computer vision data, such as the small-scale CIFAR-10 (Torralba et al., 2008; Krizhevsky, 2009) and the large-scale ImageNet (Deng et al., 2009), we show that extending I-SSL to E-SSL by also predicting four-fold rotations improves the semantic quality of the representations. We show that this approach works for other transformations too, such as vertical flips, 2x2 jigsaws, four-fold Gaussian blurs and color inversions, but focus on four-fold rotations as the most promising improvement we obtain with initial E-SSL experiments in Figure 1.

We also note that the applications of E-SSL in this paper are task specific, meaning that the representations from E-SSL may work best for a particular downstream task that benefits from equivariance

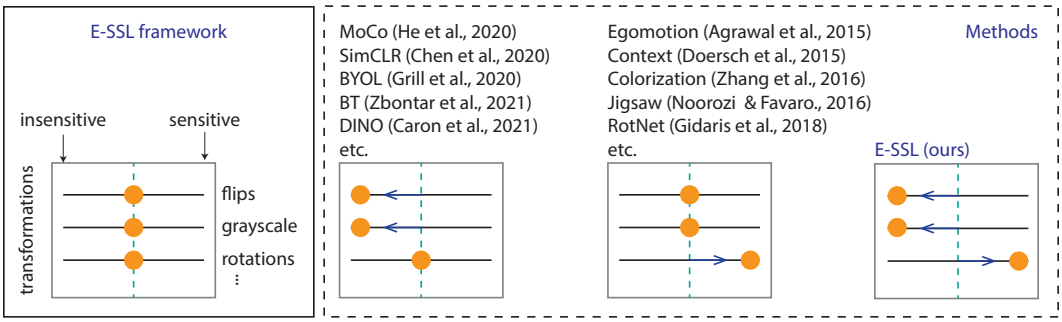

Figure 2: E-SSL framework. Left: framework. Right: methods. Egomotion, Context, Colorization and Jigsaw use other transformations than rotations, but their pattern looks like that of RotNet's.

dictated by the available data. E-SSL can be further extended to applications in science; in particular, we focus on predictive tasks using (unlabelled and labelled) data collected via experiments or simulations. The downstream tasks in prediction problems in science are often fixed and can be aided by incorporating scientific insights. Here, we also explore the generality of E-SSL beyond computer vision, on a different application: regression problems in photonics science and demonstrate examples where E-SSL is effective over I-SSL.

Our contributions can be summarized as follows:

- We introduce E-SSL, a generalization of popular SSL methods.
- We improve state-of-the-art SSL methods on CIFAR-10 and ImageNet by encouraging equivariance to four-fold rotations.
- We demonstrate the usefulness of E-SSL beyond computer vision with experiments on regression problems in photonics science.

The rest of the paper is organized as follows. In Section 2 we introduce our experimental method for E-SSL. In Section 3 we present our main experiments in computer vision. In Section 4 we elaborate on related work. In Section 5 provide a discussion around our work that extends our study beyond computer vision and studies formally the transformations we use in E-SSL. In Section 6 we conclude and point to future work.

2 Method

Our method is designed to test our primary conjecture that a *hybrid approach* of sensitive and insensitive representations learns better features. Surprisingly, this hybrid approach is not yet present in SSL, as Figure 2 illustrates. In this figure, we can view transformations in SSL as "levers." Each downstream task has an optimal configuration of the levers, which should be tuned in the SSL objective: left for insensitive and right for sensitive representations. E.g., make representations insensitive to horizontal flips and grayscale and sensitive to four-fold rotations, vertical flips, 2x2 jigsaws, Gaussian blurs or color inversions. We introduce the details behind a training method that achieves our goal below.

Let $f(\cdot;\boldsymbol{\theta}_f)$ with trainable parameters $\boldsymbol{\theta}_f$ be a backbone encoder. Analogously, let $p_1(\cdot;\boldsymbol{\theta}_{p_1})$ be a projector network for the I-SSL loss. There might be an extra prediction head and parameters, depending on the objective, which we omit for simiplicity. Let $p_2(\cdot;\boldsymbol{\theta}_{p_2})$ be the predictor network for encouraging sensitivity, which we will call "predictor for equivariance." We share the backbone encoder f jointly for I-SSL and the objective of predicting the transformations from the backbone representations. Let $\ell_{\text{I-SSL}}$ be the I-SSL loss and $\ell_{\text{E-SSL}}$ be the added E-SSL loss that encourages sensitivity to a particular transformation. Let the parameter λ be the strength of the E-SSL loss. The optimization objective for an image $\boldsymbol{x}$ with views $\{\boldsymbol{x}'\}$ in the batch is given as follows

$$\underset{\boldsymbol{\theta}_f,\boldsymbol{\theta}_{p_1},\boldsymbol{\theta}_{p_2}}{\arg\min}\ \ell_{\text{SSL}}(p_1(f(\{\boldsymbol{x}'\};\boldsymbol{\theta}_f);\boldsymbol{\theta}_{p_1})) + \lambda\ell_{\text{E-SSL}}(p_2(f(\{\boldsymbol{x}'\};\boldsymbol{\theta}_f);\boldsymbol{\theta}_{p_2})), \tag{1}$$

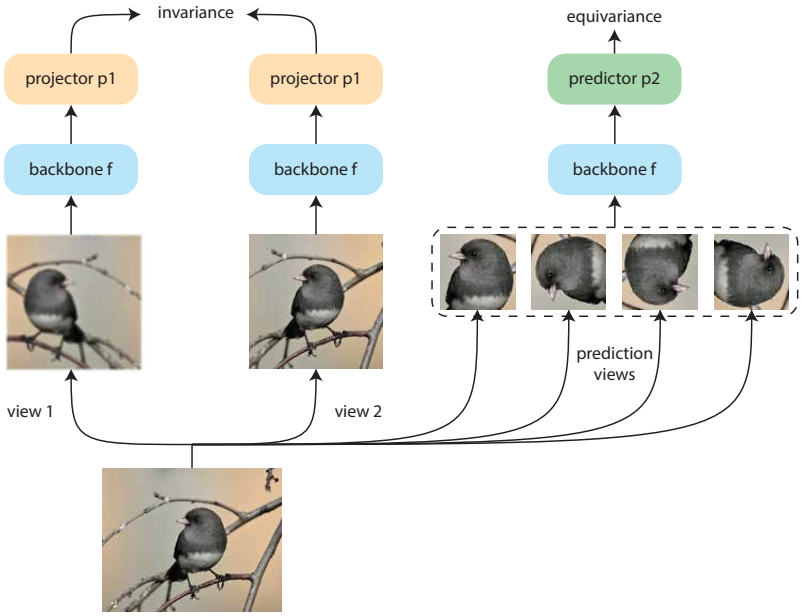

Figure 3: Sketch of E-SSL with four-fold rotations prediction, resulting in a backbone that is sensitive to rotations and insensitive to flips and blurring. ImageNet example n01534433:169.

where $\ell_{\text{E-SSL}}$ can take either one or all of the views, but we take only one for simplicity.

Figure 3 sketches how our construction works. E-SSL can be constructed for any semantically meaningful transformation r (see for example, Figure 1). For simplicity of exposition, below we show how we can construct E-SSL for four-fold rotations. We denote with $G = \{e, r, r^2, r^3\}$ the set of four-fold rotations with axis of rotation the center of the image and a generator element r that is $\pi/2$ counter-clockwise rotation about the axis. We use the convention $r^0 \equiv e$, the unit element (no transformation). Then, the standard $\text{CrossEntropyLoss}(\hat{\boldsymbol{z}}, j) = -\log \hat{\boldsymbol{p}}_j$ for $\hat{\boldsymbol{p}} = \text{softmax}(\hat{\boldsymbol{z}})$ allows us to define the rotation loss for class $j \in \{0, 1, 2, 3\}$ as

$$\ell_{\text{E-SSL}} := \frac{1}{4} \sum_{i=0}^{3} \text{CrossEntropyLoss} \left(p_2 \left(f \left(r^j(\boldsymbol{x}'); \boldsymbol{\theta}_f \right); \boldsymbol{\theta}_{p_2} \right), j \right). \tag{2}$$

From Figure 1 we choose four-fold rotations as the most promising transformation and we fix it for the upcoming section. As a minor motivation, we also present empirical results about the similarities between four-fold rotations prediction and I-SSL in Appendix B. In particular, both tasks benefit from the same data augmentation.

3 Experiments

In this section we demonstrate that our framework improves popular I-SSL methods. Our main experiments are on two standard computer vision datasets: small-scale CIFAR-10 and large-scale ImageNet. Additionally, we list the experimental details about our synthetic experiments in science, but we defer the results to Section 5, because of the novelty of the experimental setup.

3.1 Setups

CIFAR-10 setup. We use the CIFAR-10 experimental setup from (Chen & He, 2021). We consider two simple I-SSL methods: SimCLR (with InfoNCE loss (Oord et al., 2018) and temperature 0.5) and SimSiam (Chen & He, 2021). We were able to obtain baseline results close to those in (Chen & He, 2021). The predictor for equivariance takes a smaller crop with size 16x16. We report performance on the standard linear probe. We tune λ to 0.4 both for SimCLR and SimSiam (full tuning in Table 5 in Appendix C). Remaining experimental details can be found in Appendix C.

ImageNet setup. We use the original augmentation setting for each method. The predictor for equivariance takes a smaller crop with size 96x96. We use a ResNet-50 (He et al., 2016) backbone for each method. In terms of optimizer and batch size settings, we follow the standard training recipe for each method. For our SimCLR experiments we use a slightly more optimal implementation that uses BYOL's augmentations (i.e. it includes *solarization*), initializes the ResNet with zero BatchNorm weights and uses the InfoNCE loss with temperature 0.2.

Photonic-crystals setup. Photonic crystals (PhC) are periodically-structured materials engineered for wide ranging applications by manipulating light waves (Yablonovitch, 1987; Joannopoulos et al., 2008). The density-of-states (DOS) is often used as a design metric to engineer the desired properties of these crystals and thus here, we consider the regression task of predicting the DOS of PhCs. Examples of this dataset are depicted in Section 5 and further details can be found in Appendix E. The use of symmetry or invariance knowledge is common in scientific problems; here, the DOS labels are invariant to several physical transformations of the unit cell, namely, rolling translations (due to its periodicity) and operations arising from the symmetry group (C_{4v}) of the square lattice, i.e. rotations and mirror flips. We construct an encoder network comprising of simple convolutional and fully-connected layers (see Appendix E) and create various synthetic datasets to investigate encouragement of equivariance. After SSL/ E-SSL, we fine-tune the network with L1 loss; for better interpretability of prediction accuracies, we use a relative error metric following Liu et al. (2018) for evaluation, given by $\ell_{\text{DOS}} = (\sum_\omega \left|\text{DOS}^{\text{pred}} - \text{DOS}\right|)/(\sum_\omega \text{DOS})$, which is reported in percentage form.

The predictor p_2 for E-SSL. The predictor is a 2 layer MLP for CIFAR-10 and Photonic-crystals, and a 3 layer MLP for ImageNet, followed by a linear head that produces the logits for the an n-way classification (for example four-fold rotations is 4-way classification). The predictor's hidden dimension is shared across all layers and it equals 2048 for CIFAR-10 and ImageNet and 512 for PhC. After each linear layer, there is a Layer Normalization (Ba et al., 2016) followed by ReLU. We experimented with Batch Normalization (Ioffe & Szegedy, 2015) (with trainable affine parameters) instead of Layer Normalization, but did not observe any significant changes. For some experiments, we discovered that removing the last ReLU from the MLP improves the results slightly. In particular, for SimSiam on CIFAR-10 and for all models on ImageNet we omit the last ReLU.

Finally, Algorithm 1 presents pseudocode for E-SSL with four-fold rotations on ImageNet. In our implementation, we use smaller resolution for the rotated images, so that we can fit all views on the same batch and have minimal overhead for pre-training (additional details in Table 8 in Appendix D).

Algorithm 1 PyTorch-style pseudocode for E-SSL, predicting four-fold rotations.

```
# f: backbone encoder network
# p1: projector network for I-SSL
# p2: predictor network for E-SSL
# ssl_loss: loss function for I-SSL
# lambda: weight of the E-SSL


for x in loader:
    # large views for SSL and small view for EE
    V_large = augment(x, small_crop=False) # list of views
    v_small = augment(x, small_crop=True) # change: crop with size=96 and scale=(0.05, 0.14)

    # loss
    loss_invariance = ssl_loss(p1(f(V_large))
    labels = [0] * N + [1] * N + [2] * N + [3] * N # 4Nx1
    v_cat = cat([v_small] * 4, dim=0) # 4Nx3x96x96
    v_equivariance = rot90(v_cat, labels) # constructing the rotated views

    logits = p2(f(v_equivariance)) # 4Nx4
    loss_equivariance = CrossEntropyLoss(logits, labels) # rotation prediction
    loss = loss_invariance + lambda * loss_equivariance

    # optimization step
    loss.backward()
    optimizer.step()
```

Table 1: Linear probe accuracy (%) on CIFAR-10. Models are pre-trained for 800 epochs. Baseline results are from Appendix D in (Chen & He, 2021). Standard deviations are from 5 different random initializations for the linear head. Deviations are small because the linear probe is robust to the seed.

Method	SimCLR (Chen et al., 2020)	SimSiam (Chen & He, 2021)
Baseline (Chen & He, 2021)	91.1	91.8
Baseline (our reproduction)	92.0$\pm$ 0.0	91.6$\pm$ 0.0
E-SSL (ours)	**94.1**$\pm$ 0.0	**94.2**$\pm$ 0.1
Ablating E-SSL		
Single random rotation	93.4$\pm$ 0.0 (↓ 0.7)	92.6$\pm$ 0.0 (↓ 1.6)
Linear enhancing predictor	93.3$\pm$ 0.0 (↓ 0.8)	93.4$\pm$ 0.0 (↓ 0.8)
No SSL augmentation in the enhanced views	92.7$\pm$ 0.1 (↓ 1.4)	92.0$\pm$ 0.1 (↓ 2.2)
Alternatives to E-SSL		
Disentangled representations	91.3$\pm$ 0.0 (↓ 2.7)	91.1$\pm$ 0.0 (↓ 3.1)
Insensitive instead of sensitive	86.3$\pm$ 0.1 (↓ 7.8)	86.1$\pm$ 0.1 (↓ 8.1)

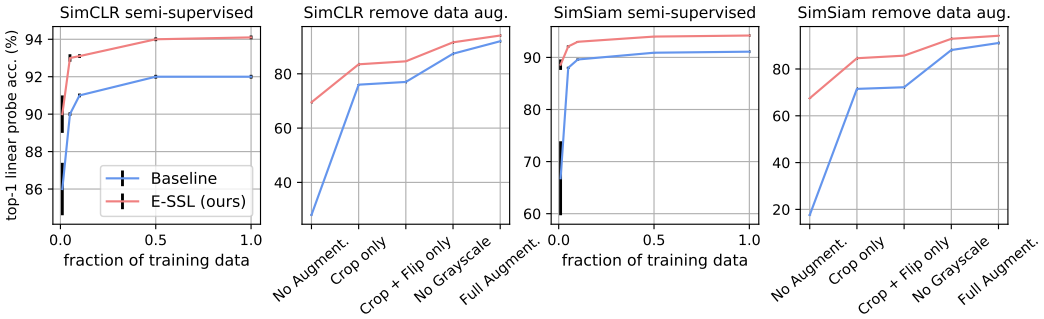

Figure 4: Reducing the labels for training and the data augmentation for pre-training on CIFAR-10. Error bars for 5 different training data splits.

3.2 Main results

CIFAR-10 results. To highlight the benefits of our method, Table 1 demonstrates the improvement we obtain by using E-SSL on top of SimCLR and SimSiam and then shows different ablations and alternative methods. We label the E-SSL extensions as E-SimCLR and E-SimSiam respectively. We observe that we can increase a tuned baseline accuracy by about $2-3\%$. When ablating E-SSL, we see that each component of E-SSL is important. Most useful is the SSL augmentation applied on top of the rotated views. We also study alternatives to E-SSL. With "Disentangled representations" we investigate whether a "middle ground" is optimal for E-SSL: half of the representation to be insensitive to a transformation and the other half to be sensitive to the same transformation. This results in degradation of performance, which reflects our hypothesis that the representations should be either insensitive or sensitive. We conducted this experiment by using four-fold rotations in I-SSL for half of the representation and E-SSL for the other half. Finally, making the representations "Insensitive instead of sensitive" to four-fold-rotations hurts the performance significantly, as it is also observed in Figure 1, and in (Chen et al., 2020; Xiao et al., 2020).

Figure 4 reveals that E-SSL is more robust to removing transformations for I-SSL or reducing the labels for training. For example, E-SimCLR and E-SimSiam with only random resized cropping obtain 83.5% and 84.6% accuracies. It is promising that encouraging sensitivity to one transformation, namely four-fold-rotations, can reduce the need for selecting many transformations for I-SSL. Furthermore, with only 1% of the training data, E-SimCLR and E-SimSiam achieve 90.0$\pm$ 1.0% and 88.6$\pm$ 1.0% respectively.

Table 2: Linear probe accuracy (%) on ImageNet. Each model is pre-trained for 100 epochs. Baseline results are from Table B.1 in (Chen et al., 2020) from Table 4 in (Chen & He, 2021). Numbers marked with * use a less optimal setting than our reproduction for SimCLR (see ImageNet setup).

Method	SimCLR (Chen et al., 2020)	SimSiam (Chen & He, 2021)	Barlow Twins (Zbontar et al., 2021)
Baseline (Chen et al., 2020)	64.7*	-	-
Baseline (Chen & He, 2021)	66.5*	68.1	-
Baseline (our reproduction)	67.3	68.1	66.9
E-SSL (ours)	**68.3**	**68.6**	**68.2**

ImageNet results. Table 2 demonstrates our main results on the linear probe on ImageNet after pre-training with various state-of-the-art I-SSL methods and their E-SSL versions. By only sweeping λ and slightly reducing the original learning rate for SimSiam we obtain consistent 1%/ 0.5%/ 1.2% improvements for SimCLR/ SimSiam/ Barlow Twins respectively. The results are encouraging, because the objectives of these three methods are quite different from each other, which suggests that E-SSL can improve different types of state-of-the-art representations in computer vision.

4 Related Work

To encourage non-trivial equivariance, we observe that a simple task that predicts the synthetic transformation applied to the input, works well and improves I-SSL already; some prediction tasks create representations that can be transferred to other tasks of interest, such as classification, object detection and segmentation. While prediction tasks alone have been realized successfully before in SSL (Agrawal et al., 2015; Doersch et al., 2015; Zhang et al., 2016; Noroozi & Favaro, 2016; Zamir et al., 2016; Gidaris et al., 2018), to our knowledge we are the first to combine simple predictive objectives of synthetic transformations with I-SSL, and successfully improve the semantic quality of representations.

To improve representations with pretext tasks, Gidaris et al. (2018) use four-fold rotations prediction as a pretext task for learning useful visual representations via a new model named RotNet. Feng et al. (2019) learn decoupled representations: one part trained with four-fold rotations prediction and another with non-parametric instance discrimination (Wu et al., 2018) and invariance to four-fold rotations. Yamaguchi et al. (2021) use a joint training objective between four-fold rotations prediction and image enhancement prediction. Xiao et al. (2020) propose to learn representations as follows: for each atomic augmentation from the contrastive learning's augmentation policy, they leave it out and project to a new space on which I-SSL encourages invariance to all augmentations, but the left-out one. The resulting representation could either be a concatenation of all projected left-out views' representations, or the representation in the shared space, before the individual projections. Our method differs from the above contributions in that E-SSL is the only hybrid framework that encourages both insensitive representations for some transformations and sensitive representations for others and does not require representations to be sensitive and insensitive to a particular transformation at the same time.

To obtain performance gains from transformations, Tian et al. (2020) study which transformations are the best for contrastive learning through the lens of mutual information. Reed et al. (2021) use four-fold rotations prediction as an evaluation measure to tune optimal augmentations for contrastive learning. Wang & Qi (2021) use strong augmentations to improve contrastive learning by matching the distributions of strongly and weakly augmented views' representation similarities to a memory bank. A growing body of work encourages invariance to domain agnostic transformations (Tamkin et al., 2021; Lee et al., 2021; Verma et al., 2021) or strengthens invariance with regularization (Foster et al., 2021). Our framework is different from the above works, because we work with transformations that encourage equivariance beyond invariance.

To understand and improve equivariant properties of neural networks, Lenc & Vedaldi (2015) study emerging equivariant properties of neural networks and (Cohen & Welling, 2016; Bronstein et al., 2021) construct equivariant neural networks. In contrast, our work does not enforce strict equivariance, but only encourages equivariant properties for the encoder network through the choice of

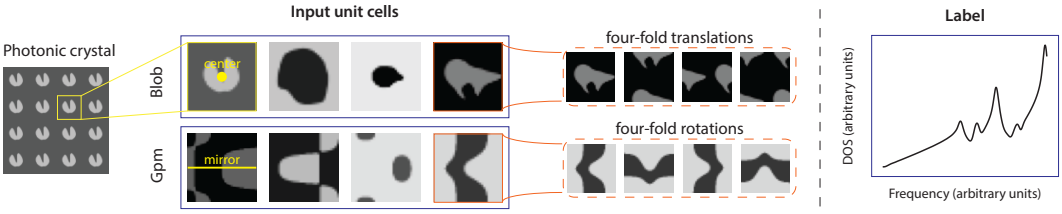

Figure 5: PhC datasets with transformations for sensitivity. The regression task is to predict the DOS labels (an example of a label in $\mathbb{R}^{400}$ is shown on the right) from 2D square periodic unit cells (examples of the inputs in $\mathbb{R}^{32\times 32}$ are shown on the left). We consider two types of input unit cells; at the top is the Blob dataset where the feature variation is always centered; at the bottom is the Group pm (Gpm) dataset where inputs have a horizontal mirror symmetry.

the loss function. While strict equivariance is concerned with groups, some of the transformations, such as random resized cropping and Gaussian blurs, may not even be groups, but they could still be analyzed in the E-SSL framework. Thus, ours is a flexible framework, which allows us to consider a variety of transformations and how the encoder might exhibit equivariant properties to them.

5 Discussion

5.1 Is E-SSL a more general framework beyond computer vision?

To show that other domains benefit from E-SSL in a qualitatively similar way to the applications in the previous section, here we introduce two datasets in photonics science. Figure 5 depicts the datasets, i.e. input-label pairs consisting of 2D square periodic unit cells of PhCs and their associated DOS. The physics of the problem dictates that the DOS is invariant to (rolling) translations, and operations of the C_{4v} symmetry group, i.e. rotations and mirror flips. In choosing the transformations that E-SSL should encourage sensitivity to, we observe that the transformations that have worked for CIFAR-10 and ImageNet disturb the natural setting of the data (e.g. rotations disturb the natural upright setting of images). Thus, we encourage sensitivity to transformations that fit this observation, and insensitivity to the rest of the transformations.

In Figure 5, the top dataset is a "Blob" dataset where the shape variation in each image is centered. We encourage sensititivy to the group of *four-fold translations*, given by $G = \{e, h, v, hv\}$, where h and v are 1/2-unit cell translations in the horizontal and vertical axis, respectively, e is the unit element (no transformation) and hv is the composition of h and v. In the bottom dataset of Figure 5, the PhC unit cells are generated to have a horizontal mirror symmetry, i.e. we use the 2D wallpaper (or crystallographic plane) group **pm**. We encourage sensitivity to the group of *four-fold rotations* (the same group we used for CIFAR-10 and ImageNet), since rotating any of the images disturbs the (horizontal) mirror symmetry. More accurately, since only $\pm\pi/2$ rotations disturb the symmetry, we separate them in two classes, $\{\pi/2, -\pi/2\}$ and $\{0, \pi\}$, and perform binary prediction in E-SSL.

Table 3: Fine-tuning the backbone on PhC datasets using 3000/ 2000 labelled train/ test samples. Relative error (%) is $\ell_{\text{DOS}} = (\sum_\omega |\text{DOS}^{\text{pred}} - \text{DOS}|)/(\sum_\omega \text{DOS})$. Lower is better. SimCLR for Blob includes C_{4v} (rotations and flips); SimCLR for Gpm includes rolling translations and mirrors. E-SimCLR encourages the features to be sensitive to the selected transformation explained in the text (four-fold translations for Blob and four-fold rotations for Gpm). "+ Transform" means adding this transformation to SimCLR. Error bars are for 3 different training data splits.

PhC Dataset	Supervised	SimCLR	SimCLR + Transform	E-SimCLR (ours)
Blob	1.068 $\pm$ 0.015	0.987$\pm$ 0.005	0.999$\pm$ 0.005	**0.974**$\pm$ 0.009
Gpm	3.212$\pm$ 0.041	3.122$\pm$ 0.002	3.139$\pm$ 0.005	**3.091**$\pm$ 0.006

Table 3 demonstrates the results when we fine-tune the backbone and an additional DOS-predictor head (see Appendix E) with 3000 labelled samples for this regression task. We observe that encour-

aging sensitivity to the selected transformations described in the previous paragraph (via E-SimCLR) leads to the largest reduction in the error. On the contrary, including these transformations to SimCLR (indicated by "+ transform") increases the error. This supports our observations about the usefulness of E-SSL over I-SSL and demonstrates E-SSL's generality beyond computer vision.

5.2 What transformations could work for E-SSL?

A common property of the successful transformations we have studied up to this point is that they form groups in the mathematical sense, i.e. (*i*) each transformation is invertible, (*ii*) composition of two transformations is part of the set of transformations and (*iii*) compositions are associative. We used four-fold rotations and four-fold translations, which are groups, to improve SSL on CIFAR-10, ImageNet and PhC for classification and regression tasks.

In this paper, we encourage equivariance to a group of transformation by predicting them. This does not guarantee that the encoder we learn will be strictly equivariant to the group. In fact, strict equivariance is possible, i.e. there exists an encoder that is non-trivially equivariant, under a reasonable assumption which is formulated as follows. Let X be the set of all images. Let G be a group whose elements $g \in G$ transform X via the function $T_g \colon X \to X$. Let $X' = \{T_g(\boldsymbol{x}) \mid g \in G, \boldsymbol{x} \in X\}$ be the set of all transformed images. Let $f(\cdot; \boldsymbol{\theta}) \colon X' \to S$ be an encoder network that we learn with parameters $\boldsymbol{\theta}$. We write $f(\cdot) \equiv f(\cdot; \boldsymbol{\theta})$ for simplicity. Finally, let $S = \{f(\boldsymbol{x}') \mid \boldsymbol{x}' \in X\}$ be the set of all representations of the images in X'. The following is our statement.

Proposition 1 (Non-trivial Equivariance)**.** *Given $T_g \colon X' \to X'$ for the group G, there exists an encoder $f \colon X' \to S$ that is non-trivially equivariant to the group G under the assumption that if $f(T_g(\boldsymbol{x})) = f(T_{g'}(\boldsymbol{x}'))$ then $g = g'$ and $\boldsymbol{x} = \boldsymbol{x}'$ for all $g, g' \in G$ and $\boldsymbol{x}, \boldsymbol{x}' \in X$.*

We defer the proof to Appendix A. The significance of this proof is that it explicitly constructs a non-trivially equivariant encoder network for groups G if the assumption is satisfied. The intuition of the assumption is that if the representations of two transformed inputs are the same, the inputs should coincide, and likewise the transformations. We speculate that satisfying this assumption is reasonable for the datasets in this work, since we observe a natural setting of the data, e.g. horizontal mirror symmetry in Gpm, and we consider transformations that disturb this natural setting.

Could other transformations still help? To motivate our work, in Figure 1 we observed additional transformations that could be useful, such as vertical flips, 2x2 jigsaws, four-fold Gaussian blurs and color inversions. All of these transformations are groups, except for four-fold Gaussian blurs. Each element of Gaussian blurs is invertible (de-blurring), but the inverse is not a transformation in the set. Interestingly, we observe that four-fold Gaussian blurs still improve the baseline, which means the success of E-SSL may not be limited to groups.

We might also consider combining the prediction of multiple transformations to encourage sensitivity to all of them. However, the gains we saw in Figure 1 may not add up when we combine transformations, because they may not be independent. The gains may also depend on the transformations that we choose for I-SSL. While we see combinations of transformations as promising future work, we focused on encouraging sensitivity to a single transformation to make a clear presentation of the E-SSL framework.

6 Conclusion and Future Work

In this paper we motivated the generalization of state-of-the-art methods in self-supervised learning to the more general framework of equivariant self-supervised learning (E-SSL). In E-SSL rather than using only invariance as a trivial case of equivariance, we encouraged non-trivial equivariance and improved state-of-the-art methods on common computer vision benchmarks and regressions tasks in photonics science. We also discussed that there are many types of equivariance we can consider for E-SSL. We observed that most of the successful transformations for E-SSL that we explored form groups, but that potentially many more transformations could be explored.

For future work one could use explicit constructions of equivariant properties as an objective to learn transformations, instead of setting them manually. Thus, the concept of E-SSL could potentially be extended to natural language processing or other science domains, whose transformations for SSL are less well-understood.

REPRODUCIBILITY STATEMENT

Algorithm 1, the original public code for each of the I-SSL methods we use in the paper and the experimental setups in Section 3.1, and in Appendices C, D and E, can be used for reproducibility. Code with annotation is available as supplementary material. We will make our code public if the paper is accepted.

REFERENCES

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 37–45, 2015.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *NeurIPS Deep Learning Symposium*, 2016.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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

Thomas Christensen, Charlotte Loh, Stjepan Picek, Domagoj Jakobović, Li Jing, Sophie Fisher, Vladimir Ceperic, John D. Joannopoulos, and Marin Soljačić. Predictive and generative machine learning models for photonic crystals. *Nanophotonics*, 9(13):4183–4192, October 2020. ISSN 2192-8614. doi: 10.1515/nanoph-2020-0197. URL `https://www.degruyter.com/document/doi/10.1515/nanoph-2020-0197/html`.

Thomas Christensen, Hoi Chun Po, John D. Joannopoulos, and Marin Soljačić. Location and topology of the fundamental gap in photonic crystals, 2021.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10364–10374, 2019.

Adam Foster, Rattana Pukdee, and Tom Rainforth. Improving transformation invariance in contrastive representation learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=NomEDgIEBwE`.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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*, pp. 9729–9738, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade. *Photonic Crystals: Molding the Flow of Light*. Princeton University Press, 2 edition, 2008. URL `http://ab-initio.mit.edu/book/`.

Steven G. Johnson and J. D. Joannopoulos. Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis. *Optics Express*, 8(3):173–190, January 2001. ISSN 1094-4087. doi: 10.1364/OE.8.000173. URL `https://www.osapublishing.org/oe/abstract.cfm?uri=oe-8-3-173`.

Samuel Kim, Peter Y. Lu, Charlotte Loh, Jamie Smith, Jasper Snoek, and Marin Soljačić. Scalable and Flexible Deep Bayesian Optimization with Auxiliary Information for Scientific Problems. *arXiv:2104.11667*, April 2021. URL `http://arxiv.org/abs/2104.11667`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *technical report*, 2009.

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*, 2021.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Boyuan Liu, Steven G. Johnson, John D. Joannopoulos, and Ling Lu. Generalized Gilat-Raubenheimer method for density-of-states calculation in photonic crystals. *Journal of Optics*, 20(4):044005, April 2018. ISSN 2040-8978, 2040-8986. doi: 10.1088/2040-8986/aaae52. URL `http://arxiv.org/abs/1711.07993`.

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

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

Colorado Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Evaluating self-supervised pretraining without using labels. In *CVPR*, 2021.

Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=enoVQWLsfyL`.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. doi: 10.1109/TPAMI.2008.128.

Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10530–10541. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/verma21a.html`.

Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *arXiv preprint arXiv:2104.07713*, 2021.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *CVPR*, 2018.

Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020.

Eli Yablonovitch. Inhibited Spontaneous Emission in Solid-State Physics and Electronics. *Physical Review Letters*, 58(20):2059–2062, May 1987. doi: 10.1103/PhysRevLett.58.2059. URL `https://link.aps.org/doi/10.1103/PhysRevLett.58.2059`.

Shin'ya Yamaguchi, Sekitoshi Kanai, Tetsuya Shioda, and Shoichiro Takeda. Image enhanced rotation prediction for self-supervised learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 489–493. IEEE, 2021.

Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3d representation via pose estimation and matching. In *European Conference on Computer Vision*, pp. 535–553. Springer, 2016.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

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

A Proof of Proposition 1

Proof. To construct a non-trivially equivariant f, we first need to show that both X' and S are G-sets, i.e. that there is a group action T_g of G on X', which is given by the statement of the proposition, and another (non-trivial) group action T'_g of G on S, which we will construct. Then, we need to show that f commutes with the group action, i.e. that $f(T_g(\boldsymbol{x}')) = T'_g(f(\boldsymbol{x}'))$.

Group actions. Note that by the setup of the problem, we are already given how G acts on the input X', i.e. T_g is known. For example, if G is the group of four-fold rotations, then T_g is the rotation of the input by a multiple of $\pi/2$. We proceed to construct the non-trivial group action T'_g of G on S.

Define the function $T' \colon G \times S \to S$ as $T'(g, s) = f(T_g(T_{g'}(\boldsymbol{x}')))$, where $s = f(T_{g'}(\boldsymbol{x}'))$. Note that T' is well-defined, because $gg' \in G$ by the closure of the group and s is uniquely written as $s = f(T_{g'}(\boldsymbol{x}'))$. To see why s is uniquely written, it suffices to show that if $f(T_{g'}(\boldsymbol{x}')) = f(T_{g''}(\boldsymbol{x}''))$ then both $g' = g''$ and $\boldsymbol{x}' = \boldsymbol{x}''$, which follows directly from our assumption in the statement.

Now, to prove that T' is a group action, it suffices to show two properties.

- Identity: $T'(e, s) = s$ for $s = f(g'(\boldsymbol{x}'))$ and e is the unit element of the group. To show that, note that by definition $T'(e, s) = f(T_e(T_{g'}(\boldsymbol{x}'))) = f(T_{g'}(\boldsymbol{x}'))$, because $eg' = g'$.
- Compositionality: $T'(g, T'(h, f(T_{g'}(\boldsymbol{x}')))) = T'(gh, f(T_{g'}(\boldsymbol{x}')))$. To show this, we expand the LHS and use the definition of T' to obtain as follows $T'(g, T'(h, f(T_{g'}(\boldsymbol{x}')))) = T'(g, f(T_h(T_{g'}(\boldsymbol{x}')))) = f(T_g(T_{hg'}(\boldsymbol{x}'))) = f(T_{ghg'}(\boldsymbol{x}')) = f(T_{gh}(T_{g'}(\boldsymbol{x}'))) = T'(gh, f(T_{g'}(\boldsymbol{x}')))$, because the group operation is associative.

Hence, T' is a group action, and thus S is a G-set, and we can write $T'(g, \cdot) \equiv T'_g(\cdot)$.

Commuting with the group action. To see this property, note that $T'_{g'}(f(\boldsymbol{x}')) = T'_{g'}(f(T_g(\boldsymbol{x}))) = f(T_{g'}(T_g(\boldsymbol{x}))) = f(T_{g'}(\boldsymbol{x}'))$ as desired. Note that T'_g is non-trivial.

Therefore, we can conclude that f, which satisfies the constructed group action T'_g, is not-trivially equivariant to the group G. □

B ROTATION PREDICTION AND I-SSL BENEFIT FROM SIMILAR DATA AUGMENTATION.

Recently, rotation prediction with a linear head from the frozen backbone representations proved to be useful for validating the augmentation policies of contrastive learning (Reed et al., 2021). This shows that the two tasks of classification of *ground truth classes* and *synthetic rotation classes* from frozen backbone representations benefit from similar augmentation policies. We took this experiment a step further, and performed rotation prediction with the augmentation policies, typically used in contrastive learning.

The result is in Table 4. Interestingly, RotNet benefits from augmentations, typically used in contrastive learning, and the RotNet training shares the same sweet spot (Tian et al., 2020) as kNN classification. There are several takeaways from this experiment: (*i*) we can find good augmentations for contrastive learning by doing RotNet *alone*, i.e. without doing *any* contrastive learning; (*ii*) RotNet benefits from augmentations needed in contrastive learning; (*iii*) we may be able to combine four-fold rotations prediction and contrastive learning.

Table 4: RotNet's augmentation sweet spot. kNN and Rotation Prediction have the same sweep spot (Level 4) which gives best accuracy in both columns. RotNet is trained on CIFAR-10 for 100 epochs with the same optimization setup as in our I-SSL experiments. Accuracies are on the test split. (↓ ·) marks the deviation from the sweet spot. Every new level adds a new augmentation to the previous level incrementally.

Level	Added Augmentation	Supervised kNN Acc. (%)	Rotation Prediction Acc. (%)
0	none	44.8 (↓ 19.8)	90.2 (↓ 4.8)
1	random resized cropping	59.2 (↓ 5.4)	93.7 (↓ 1.3)
2	horizontal flips w.p. 0.5	59.4 (↓ 5.2)	94.5 (↓ 0.5)
3	color jitter w.p. 0.8	64.3 (↓ 0.3)	94.9 (↓ 0.1)
4	grayscale w.p. 0.2	**64.6**	**95.0**
5	Gaussian blur w.p. 0.2	64.1 (↓ 0.5)	94.5 (↓ 0.5)
6	random rotation ($\pm\pi/6$)	59.4 (↓ 5.2)	93.1 (↓ 1.9)
7	vertical flip w.p. 0.5	51.9 (↓ 12.7)	90.6 (↓ 4.4)

C CIFAR-10 EXPERIMENTS

C.1 EXPERIMENTAL SETUP

Our experiments use the following architectural choices: ResNet-18 backbone (the CIFAR-10 version has kernel size 3, stride 1, padding 1 and there is no max pooling afterwards); 512 batch size (only our baseline SimSiam model uses batch size 1024); 0.03 base learning rate for the baseline SimCLR and SimSiam and 0.06 base learning rate for E-SimCLR and E-SimSiam; 800 pre-training epochs; standard cosine decayed learning rate; 10 epochs for the linear warmup; two layer projector with hidden dimension 2048 and output dimension 2048; for SimSiam a two layer (bottleneck) predictor with hidden dimension 512 whose learning rate is not decayed; the last batch normalization for the projector does not have learnable affine parameters; 0.0005 weight decay value; SGD with momentum 0.9 optimizer. The augmentation is Random Resized Cropping with scale (0.2, 1.0), aspect ratio (3/4, 4/3) and size 32x32, Random horizontal Flips with probability 0.5, Color Jittering (0.4, 0.4, 0.4, 0.1) with probability 0.8 and Grayscale with probability 0.2. Some of our evaluations use a kNN-classifer with 200 neighbors, cosine similarity and Gaussian kernel with temperature 0.1. This evaluation correlates well with the standard linear probe, but it is more efficient to calculate. We report the kNN accuracy in % at the end of the 800 epochs of training. For our main results, we report a linear probe accuracy from training a linear classifier for 100 epochs on top of the frozen representations with SGD with momentum 0.9 and cosine decay of the learning rate, batch size 256 and initial laerning rate of 30. For linear probe experiments we try 5 different initalizations of the linear head and report mean and standard deviations. The deviations are negligible because the linear probe is robust to the random seed. All parameters are reported in a Pytorch-like style.

For Figure 1 we use resolution of 32x32 for the transformations studied. The 4 levels of the Gaussian blur are for kernel sizes 0, 5, 9 and 15 in the default Gaussian blur torchvision implementation. The prediction of the transformations follows the experimental setup in Section 2. When we apply the transformations in I-SSL, we add them in the beginning of the augmentation policy with probability 1. The same setup is used for "Disentangled representations" and "Insensitive instead of sensitive" in Table 1.

C.2 Additional experiments

Explored hyperparameters. Both for SimCLR and SimSiam we ran a grid search over the following hyperparameters: base learning rate: {0.01, 0.03, 0.06}, batch size: {512, 1024}, λ (for E-SSL): {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}, predictor's MLP depth: {2, 3, 4}, predictor's normalization: {None, BatchNorm, LayerNorm}, nonlinearity at the last MLP layer of the predictor: {True, False}.

Tuning λ. Table 5 shows tuning of the CIFAR-10 results. We observe noticeable improvements over the SSL baselines by using E-SSL instead.

Table 5: Tuning the λ parameter for CIFAR-10.

Method	Baseline	E-SSL				
	0.0	0.2	0.4	0.6	0.8	1.0
SimCLR	92.0± 0.0	93.6± 0.0	**94.1± 0.0**	94.0± 0.0	**94.1± 0.0**	93.5± 0.0
SimSiam	91.1± 0.0	94.1± 0.0	**94.2± 0.1**	93.7± 0.0	93.8± 0.0	93.3± 0.0

Sensitivity to transformations for I-SSL. Table 6 demonstrates that E-SSL can produce good representation with as few SSL transformations for I-SSL as possible. We observe that E-SSL is less sensitive than SSL to the choice of data augmentation.

Table 6: Comparing the augmentation sensitivity for CIFAR-10. Levels: 0 is no transformations; 1 adds random resized cropping; 2 adds horizontal flips; 3 adds color jitter; 4 adds grayscale.

Method	Augmentation Level				
	0	1	2	3	4
SimCLR	28.0± 0.1	76.0± 0.1	77.0± 0.0	87.4± 0.0	92.0± 0.0
E-SimCLR	69.5± 0.0	83.5± 0.0	84.6± 0.1	91.6± 0.0	**94.1± 0.0**
SimSiam	17.6± 0.1	71.5± 0.0	72.2± 0.0	88.1± 0.0	91.1± 0.0
E-SimSiam	67.5± 0.1	84.6± 0.0	85.7± 0.1	92.9± 0.0	**94.2± 0.1**

The importance of complete invariance or sensitivity. Table 7 studies whether a middle ground for the representations exist, i.e. whether it is possible to have part of the representation invariant and the other part sensitive to the transformation. If we apply the E-SSL loss only to half of the representation, then there is a very small drop in the performance. Furthermore, we observe that having a disjoint mix between insensitivity and sensitivity in the representation is noticeably harmful.

Table 7: Studying the effect of disjoint representations on CIFAR-10. Split Representation means that we encourage similarity only on one half of the backbone representation. Disentangled Representation means that one half of the representation is trained to be insensitive to four-fold rotations and the other half is sensitive four-fold rotations. Linear probe accuracy (%) after 800 epochs.

Method	Baseline	Split Representation	Disentangled Representation
E-SimCLR	**94.1**± 0.0	94.1± 0.0 (↓ 0.0)	91.3± 0.0 (↓ 2.7)
E-SimSiam	**94.2**± 0.1	93.8± 0.0 (↓ 0.4)	91.1± 0.0 (↓ 3.1)

D IMAGENET EXPERIMENTS

We had limited computational resources, so we kept the learning rates the same as in the original methods. Only for SimSiam we found that choosing a smaller learning rate 0.08 leads to better results for E-SimSiam. We only swept the λ parameter, where for SimCLR and SimSiam the sweep was between 0 and 1 and for Barlow Twins it was between 0 and 100. The optimal λ is 0.2 for SimCLR, 0.08 for SimSiam, 8 for Barlow Twins.

Table 8 lists the overhead from using rotation prediction in our experiments.

Table 8: Overhead in doing rotation prediction. Reported GPU hours for an experiment on 100 epochs.

	SimCLR	SimSiam	Barlow Twins
Baseline	256	295	246
E-SSL (ours)	307	364	294
Overhead	20%	23%	19%

E PHC EXPERIMENTS

Dataset generation. 2D Photonic crystals (PhCs) are characterized by a periodically varying permitivitty $\varepsilon(x, y)$; here, for simplicity we consider a "two-tone" permitivitty profile i.e. $\varepsilon \in \{\varepsilon_1, \varepsilon_2\}$, with $\varepsilon_i \in [1, 20]$ discretized to a resolution of 32×32. To generate the unit cells in the "blob" dataset, we follow the proceedure in Christensen et al. (2020). For the Gpm dataset, the unit cells are defined using a level set of a 2D Fourier sum function like in Kim et al. (2021), with additional constraints applied to the lattice to create the mirror symmetry adopted from the method in Christensen et al. (2021). We then follow the procedure in Kim et al. (2021) to compute, and subsequently process, the density-of-states (DOS) of each unit cell, specifically, via the MIT Photonics Bands (MPB) software (Johnson & Joannopoulos, 2001) and the Generalized Gilat-Raubenheimer method in an implementation from Liu et al. (2018).

Network architecture. We use an encoder network composing of simple convolutional (CNN) and fully-connected (FC) layers for the backbone; specifically, our backbone begins with 3 CNN layers, all with a kernel size of 7 and channel dimensions given by $[64, 256, 256]$. The output is flattened and fed into 2 FC layers each with 1024 nodes (i.e. the representations have dimension 1024). We include BatchNorm (Ioffe & Szegedy, 2015), ReLU and MaxPooling for the CNNs, and ReLU only for the first FC layer. The projector and predictor networks, p_1 and p_2 are 2-layer MLPs with hidden dimension 512, with BatchNorm and ReLU between each layer except the last and the projection dimension for p_1 is 256. Additionally, since this is a regression task and the label space is much larger than in image classification tasks, we include a dense DOS-predictor head after the representations, which is fine-tuned with 3000 labelled samples after SSL or E-SSL. The DOS-predictor has 4 FC layers, with number of nodes given by $[1024, 1024, 512, 400]$. We explore two fine-tuning protocols of the DOS-predictor: freezing the backbone (discussed later in the Appendix) or fine-tuning the backbone (discussed in the main text).

Hyperparameters. For SSL and E-SSL, we performed 250 pre-training epochs using the SGD optimizer with a standard cosine decayed learning rate; the batch size was fixed to 512. The pre-trained model was saved at various epochs {20, 50, 100, 180, 250} for further fine-tuning. Fine-tuning was performed for 100 epochs using Adam optimizer and a fixed batch size of 64. We ran a grid search over the following hyperparameters; for pre-training, base learning rate: $\{10^{-3}, 10^{-4}, 10^{-5}\}$, λ (for E-SSL): {0.2, 1.0, 2.0, 5.0, 10.0}, and for fine-tuning: a learning rate in $\{10^{-3}, 10^{-4}, 10^{-5}\}$.

Frozen backbone experiment. In Table 9 we present our results from freezing the backbone encoder while fine-tuning the DOS-predictor head. We observe similar trends as in Table 3 where we allowed fine-tuning of the backbone. Relative error is reported in % and the lower the error is, the better. SimCLR for Blob includes C_{4v} (rotations and flips) and SimCLR for Gpm includes

rolling translations and flips. E-SimCLR encourages the features to be sensitive to the selected transformation (four-fold translations for Blob and four-fold rotations for Gpm), which improves the performance of SimCLR. On the contrary, adding the selected transformation to SimCLR, as indicated by "+ Transform", increases the error of SimCLR. Error bars are reported for 3 different choices of training data. Supervised (frozen) refers to the impractical situation of freezing a random backbone and fine-tuning the DOS-predictor.

Table 9: Frozen backbone experiment on PhC datasets for 3000/ 2000 labelled train/ test samples.

PhC Dataset	Supervised (frozen)	SimCLR	SimCLR + Transform	E-SimCLR (ours)
Blob	1.686$\pm$ 0.014	1.237$\pm$ 0.005	1.242$\pm$ 0.013	**1.165**$\pm$ 0.020
Gpm	5.450$\pm$ 0.077	3.214$\pm$ 0.048	3.313$\pm$ 0.029	**3.187**$\pm$ 0.000