# Correcting Three Existing Beliefs on Mutual Information in Contrastive Learning

**Anonymous authors**
Paper under double-blind review

## Abstract

Contrastive learning has played a pivotal role in the recent success of unsupervised representation learning. It has been commonly explained with instance discrimination and a mutual information loss, and some of the fundamental explanations are based on mutual information analysis. In this work, we develop new methods that enable rigorous analysis of mutual information in contrastive learning. Using the methods, we investigate three existing beliefs and show that they are *incorrect*. Based on the investigation results, we address two issues in the discussion section. In particular, we question if contrastive learning is indeed an *unsupervised* representation learning method because the current framework of contrastive learning relies on validation performance for tuning the augmentation design.

## 1 Introduction

Contrastive learning (Oord et al., 2018; Chen et al., 2020a) has achieved a remarkable success in the field of unsupervised representation learning, and one of the known success factors is the *InfoNCE loss* (Poole et al., 2019) that is used as the training objective. InfoNCE loss not only plays a key role for achieving a robust and outstanding performance, but it also provides an elegant interpretation where the representation learning can be understood as a *Mutual Information* (MI) maximization between the two augmented views ($X$ and $Y$) of a given image (Oord et al., 2018; Hjelm et al., 2018; Bachman et al., 2019; Tian et al., 2020a; Sordoni et al., 2021). Numerous works have studied contrastive learning based on the theoretical interpretation, and some have become fundamental and crucial for understanding contrastive learning.

An analysis based on MI of $X$ and $Y$, however, can be tricky and misleading. First of all, an exact evaluation of MI requires the joint distribution function $p(x, y)$, but $p(x, y)$ is not directly accessible for practical problems. For practical problems with complex neural representations and intractable $p(x, y)$, the neural estimators based on variational bounds are known to be the most reliable (Belghazi et al., 2018; Poole et al., 2019). The neural estimators, however, do not guarantee a sound analysis because they can only provide estimates and we cannot tell if the estimates are sufficiently accurate for the analysis of interest. This problem can be alleviated if we can have a practical dataset with known true MI values, but many of the previous works simply assumed the estimates to be exact in the absence of the true MI values. Second, the limitations of the MI estimators should be carefully related to the limitations of what contrastive learning can learn. Because of the use of InfoNCE loss as the objective of contrastive learning, where InfoNCE is also a popular MI estimator (Oord et al., 2018; Poole et al., 2019; Song and Ermon, 2019; Tschannen et al., 2019), many of the previous works incorrectly assumed the limitations to be the same for both MI estimation and contrastive learning. Third, a precaution is needed for interpreting the MI value where MI is simply a measure of the shared information between the two views. Without carefully examining what information is actually being shared by the two views, the interpretation can be completely misleading.

In this work, our main contribution is to develop a set of rigorous methods for analyzing MI in contrastive learning and to show that the following three existing beliefs are incorrect.

1. A small batch size is undesirable for contrastive learning because of InfoNCE's $\mathcal{O}(\log K)$ bound (Hjelm et al., 2018; Tian et al., 2020a; Bachman et al., 2019; Wu et al., 2020; Song and Ermon, 2020; Chen et al., 2020a; Sordoni et al., 2021).

2. MI cannot measure how effective the representation is for the downstream task's performance (Tschannen et al., 2019). Instead, other metrics such as uniformity (Wang and Isola, 2020; Wang and Liu, 2021), alignment (Wang and Isola, 2020), tolerance (Wang and Liu, 2021), and linear CKA (Nguyen et al., 2020; Song et al., 2012; Nguyen et al., 2022) are more relevant and useful than MI.
3. For designing optimal views, task-irrelevant information needs to be discarded for a better generalization (Tian et al., 2020b; Tsai et al., 2020; Xiao et al., 2020; Chen et al., 2021a).

For a rigorous investigation, we develop an analysis framework based on three key elements. First, we clarify that *the choice of augmentation design* dictates the shared information between the two views. While this can sound obvious, it is a crucial step for cautiously investigating contrastive learning because the choice of augmentation design directly commands the joint distribution $p(x, y)$, in turn $p(x, y)$ decides the MI of learning, and ultimately the MI determines what will be learned as the representation. A specific choice of augmentation, named *same-class sampling* in our work, plays a pivotal role in our study. It is special because it only shares the class information between the two views and because its true MI can be proven to be the same as the class entropy $H(C)$ under a mild assumption. Second, we use a dedicated phase of MI estimation that is called *post-training MI estimation*. In the previous works, MI estimation was typically performed concurrently during the training phase because the InfoNCE can be conveniently used not only as the training loss but also as the variational estimator. Separating MI estimation into a post-training phase allows us to compare a wide scope of representation encoders because it is applicable to any representation encoder (e.g., a basic supervised network learned with the cross-entropy loss). Third, we introduce *CDP dataset* that allows information to be embedded in an image by varying color, digit, and position. Thanks to the way CDP dataset is constructed, the true MI value can be easily manipulated by controlling the dependency among the three attributes over the two views. Using the CDP dataset, we were able to construct a few experiments without any ambiguity in interpretation. Also, we were able to confirm that the MI estimation values in our experiments are accurate. This was made possible by comparing the theoretically derived true MI values with the estimated MI values.

Finally, we discuss two essential issues based on our investigation results. First, we clarify that a properly chosen MI is an excellent metric for evaluating representations, but the same metric is not an effective training objective for a successful representation learning. Second, we raise the question of whether contrastive learning is really an unsupervised representation learning method. The current framework heavily relies on a heuristic and extensive tuning of the augmentation design based on a validation dataset. Apparently, it still remains open to develop a further advanced representation learning framework compared to the contrastive learning.

# 2 Method

## 2.1 Background – InfoNCE for contrastive learning and MI estimation

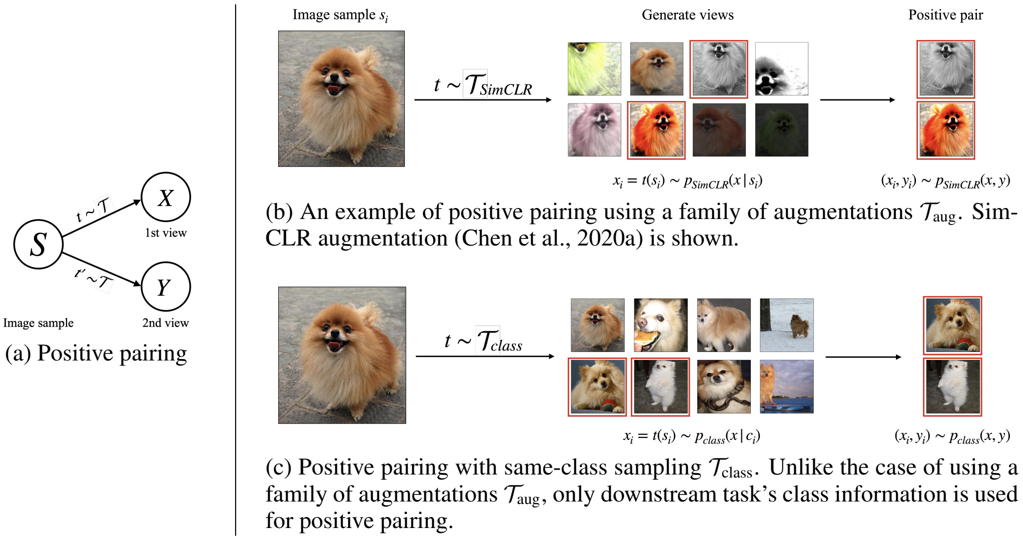
Given a dataset $\mathcal{D} = \{s_i | s_i \in \mathbb{R}^m\}$, we can sample an image $s_i$, generate its views with a family of augmentations $\mathcal{T}$, and randomly select two of them to form a positive pair $(x_i, y_i)$. See Figure 1(b) for an example where $\mathcal{T}$ is a family of SimCLR (Chen et al., 2020a) augmentations. After repeating it $K$ times, InfoNCE loss for a batch can be calculated as

$$\mathcal{L} = \frac{1}{2K} \sum_{k=1}^{K} [l(2k-1, 2k) + l(2k, 2k-1)] \text{ with } l(i,j) = -\log \frac{\exp(z_{i,i}/\tau)}{\sum_{j=1}^{2K} \mathbb{1}_{[j \neq i]} \exp(z_{i,j}/\tau)}, \quad (1)$$

where $z_{i,j} = \text{sim}(f(x_i), f(y_j))$; $f = f_p \circ f_e$ with $f_e(\cdot)$ as the encoder and $f_p(\cdot)$ as the projection head; $\text{sim}(u, v) = u^T v / ||u|| ||v||$ denotes the dot product between $l_2$ normalized $u$ and $v$ (i.e. cosine similarity); $\tau$ denotes a temperature scalar; and $K$ is the batch size. We denote the encoded representation vector of an input $X$ as $h_X = f_e(X)$. While the InfoNCE loss can be used for training, it can be slightly modified to the following InfoNCE bound and used for Mutual Information (MI) estimation, too.

$$\hat{I}(h_X; h_Y) = \log(2K - 1) - \mathcal{L} \leq \log(2K - 1) \quad (2)$$

See Supplementary A.3 for the well-known derivation. From Eq. (2), we can see that minimizing InfoNCE loss $\mathcal{L}$ is equivalent to maximizing InfoNCE bound $\hat{I}(h_X; h_Y)$.


S
X
Y
$t \sim \mathcal{T}$
$t' \sim \mathcal{T}$
1st view
2nd view
Image sample
Image sample $s_i$
Generate views
Positive pair
$t \sim \mathcal{T}_{SimCLR}$
$x_i = t(s_i) \sim p_{SimCLR}(x \mid s_i)$
$(x_i, y_i) \sim p_{SimCLR}(x, y)$
$t \sim \mathcal{T}_{class}$
$x_i = t(s_i) \sim p_{class}(x \mid c_i)$
$(x_i, y_i) \sim p_{class}(x, y)$

(a) Positive pairing

(b) An example of positive pairing using a family of augmentations $\mathcal{T}_{\text{aug}}$. SimCLR augmentation (Chen et al., 2020a) is shown.

(c) Positive pairing with same-class sampling $\mathcal{T}_{\text{class}}$. Unlike the case of using a family of augmentations $\mathcal{T}_{\text{aug}}$, only downstream task's class information is used for positive pairing.

Figure 1: Positive pairing method implicitly determines the joint distribution – $p(x, y)$ is determined by the choice of $\mathcal{T}$.

### 2.2 Expanding the concept of augmentation to positive pairing

In contrastive learning, the chosen family of augmentations $\mathcal{T}_{\text{aug}}$ plays the critical role of implicitly determining the joint distribution $p(x, y)$ and the marginal distribution $p(x)p(y)$. For the actual training, however, we do not need to know the exact distributions. Instead, we just need to be able to sample with the distributions. Therefore, the concept of augmentation ($\mathcal{T}_{\text{aug}}$) can be certainly expanded to the concept of positive pairing ($\mathcal{T}$) as shown in Figure 1. Positive pairing can be performed with an augmentation function as shown in Figure 1(b) or without any augmentation function as shown in Figure 1(c).

In our study, we heavily rely on a simple yet special positive pairing method called *same-class sampling*, $\mathcal{T}_{\text{class}}$. As shown in Figure 1(c), the same-class sampling only relies on the downstream task's label information and it does not utilize any augmentation at all. Same-class sampling is special because the only shared information between the two views is the downstream task's class information. In this case, the true MI for its joint distribution $p_{\text{class}}(x, y)$ can be proven to be upper bounded by the entropy of the class distribution, $H(C)$. We provide a simple proof in Supplementary A.1, and we also provide a stronger result of an equality proof under a mild assumption in Supplementary A.2. The $H(C)$ upper bound on same-class sampling reveals that the downstream-task information, with its entropy $H(C)$, is the only meaningfully shared information between a pair of positive examples. This result can be conveniently utilized in our empirical investigations because the calculation of $H(C)$ is trivial for uniformly distributed class labels. Note that the same-class sampling is a supervised method because it utilizes class information. *We are introducing this supervised method only for the purpose of theoretical study and empirical investigation, and we are not suggesting its use for a practical purpose.* We denote the true MI of same-class sampling as $I_{\text{class}}(h_X; h_Y)$ and its estimate as $\hat{I}_{\text{class}}(h_X; h_Y)$.

Unlike the same-class sampling, MI of augmentation-based methods such as $\mathcal{T}_{\text{SimCLR}}$ (Chen et al., 2020a), $\mathcal{T}_{\text{AutoAugment}}$ (Cubuk et al., 2018), and $\mathcal{T}_{\text{RandAugment}}$ (Cubuk et al., 2020) are intractable because the shared information is dependent on the particular choice of $\mathcal{T}_{\text{aug}}$ whose joint distribution is unknown. In our study, we select $\mathcal{T}_{\text{SimCLR}}$ as the representative example of $\mathcal{T}_{\text{aug}}$ because it has been widely used in the previous works (Chen et al., 2020b; Chen and He, 2021; Caron et al., 2020; Grill et al., 2020; Zbontar et al., 2021; Bardes et al., 2021; Tomasev et al., 2022).

### 2.3 Training vs. MI estimation

As explained in Section 2.1, InfoNCE can be used as a training loss or as a bound for MI estimation. Let's consider the training first. As shown in Figure 2(Top), training is not only dependent on the

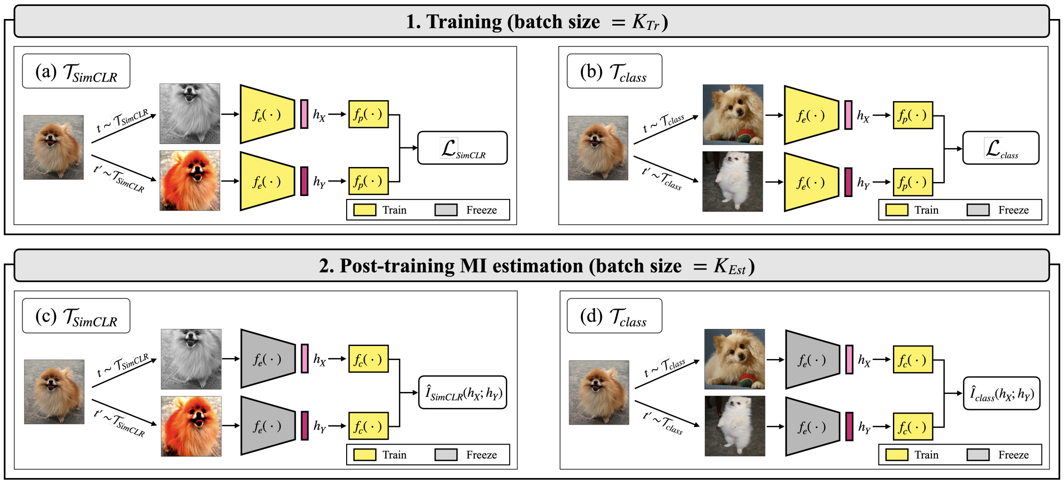

1. Training (batch size $= K_{Tr}$)
(a) $\mathcal{T}_{SimCLR}$
$t \sim \mathcal{T}_{SimCLR}$
$t' \sim \mathcal{T}_{SimCLR}$
$f_e(\cdot)$
$h_X$
$h_Y$
$f_p(\cdot)$
$\mathcal{L}_{SimCLR}$
Train
Freeze
(b) $\mathcal{T}_{class}$
$t \sim \mathcal{T}_{class}$
$t' \sim \mathcal{T}_{class}$
$\mathcal{L}_{class}$
2. Post-training MI estimation (batch size $= K_{Est}$)
(c) $\mathcal{T}_{SimCLR}$
$f_c(\cdot)$
$\hat{I}_{SimCLR}(h_X; h_Y)$
(d) $\mathcal{T}_{class}$
$\hat{I}_{class}(h_X; h_Y)$

Figure 2: Training and MI estimation. (Top) Training: We train the encoder $f_e(\cdot)$ and the projection head $f_p(\cdot)$ to minimize the InfoNCE loss $\mathcal{L}$. (a) With augmentation-based method $\mathcal{T}_{\text{SimCLR}}$. (b) With same-class sampling $\mathcal{T}_{\text{class}}$. (Bottom) Post-training MI estimation: We train the critic $f_c(\cdot)$ to maximize the InfoNCE bound $\hat{I}(h_X; h_Y)$ while $f_e(\cdot)$ is frozen. (c) With augmentation-based method $\mathcal{T}_{\text{SimCLR}}$. (d) With same-class sampling $\mathcal{T}_{\text{class}}$.

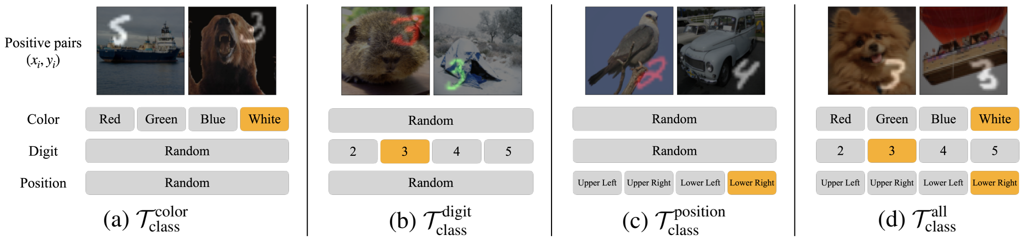
choice of loss but also on the choice of positive pairing $\mathcal{T}$. For brevity, we denote the loss as $\mathcal{L}_{\text{SimCLR}}$ and $\mathcal{L}_{\text{class}}$ when InfoNCE loss in Eq. (1) is used with $\mathcal{T}_{\text{SimCLR}}$ and $\mathcal{T}_{\text{class}}$, respectively. Because same-class sampling $\mathcal{T}_{\text{class}}$ requires class label, training with $\mathcal{L}_{\text{class}}$ implies a supervised training.

Second, let's consider MI estimation. Most, if not all, of the previous works have estimated MI during the training. This imposes a limitation where $\mathcal{T}$ for training and $\mathcal{T}$ for MI estimation cannot differ. Furthermore, the encoder weights are not fixed during training and thus the MI of a moving target needs to be estimated. To overcome the limitations, we propose *post-training MI estimation* that is illustrated in Figure 2(Bottom). With our post-training MI estimation, we have the flexibility to estimate MI that corresponds to any positive pairing and its joint distribution including $p_{\text{SimCLR}}(x, y)$ and $p_{\text{class}}(x, y)$. Also, we have the flexibility to choose any network pre-trained in a supervised or unsupervised way because the encoder network is kept frozen during the MI estimation phase. Overall, we can examine either $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ or $\hat{I}_{\text{class}}(h_X; h_Y)$ of any pre-trained network $f_e(\cdot)$ using the post-training MI estimation process shown in Figure 2(Bottom). Also, separating MI estimation into a post-training phase can improve the estimation accuracy because we can use a larger batch size for the estimation without affecting the choice of batch size for training and the resulting learning dynamics of the encoder.

For training, a projection head $f_p(\cdot)$ is used as shown in Figure 2(Top). For MI estimation, a critic function $f_c(\cdot)$ is used as shown in Figure 2(Bottom). We use a common MLP network for both $f_p(\cdot)$ and $f_c(\cdot)$ to ensure a fair comparison. See Supplementary B.2 for the details. Again, the introduction of supervised $\mathcal{L}_{\text{class}}$ is not for a practical purpose but only for in-depth investigations.

### 2.4 CDP DATASET

In the existing MI analyses that are related to practical contrastive learning, only the estimated MI value has been studied simply because evaluating the true MI value has not been possible. For a dataset that allow the class label to be clearly identified for each image, however, the true MI value for same-class sampling can be proven to be equal to the class label entropy, $H(C)$. The proof is provided in Supplementary A.2. To take advantage of this special case, we introduce a synthetic dataset named *CDP dataset*. In CDP dataset, each image is constructed by uniformly choosing a color $c_{\text{color}}$ from $\{\text{Red}, \text{Green}, \text{Blue}, \text{White}\}$, a digit $c_{\text{digit}}$ from $\{2, 3, 4, 5\}$, and a position $c_{\text{position}}$ from $\{\text{Upper left}, \text{Upper right}, \text{Lower left}, \text{Lower right}\}$. The three attributes are independently chosen for each image. Because of the uniform selection, the entropy of each class label is clearly



(a) $\mathcal{T}_{\text{class}}^{\text{color}}$ (b) $\mathcal{T}_{\text{class}}^{\text{digit}}$ (c) $\mathcal{T}_{\text{class}}^{\text{position}}$ (d) $\mathcal{T}_{\text{class}}^{\text{all}}$

Figure 3: Manipulating true MI with CDP dataset. When only one of color, digit, and position is matched by same-class sampling as shown in (a), (b), and (c), the true MI is 2 bits ($I_{\text{class}}(X;Y) = 2$). When all three are consistently matched as shown in (d), the true MI is 6 bits ($I_{\text{class}}(X;Y) = 6$).

$H(C_{\text{color}}) = H(C_{\text{digit}}) = H(C_{\text{position}}) = 2$ bits. Note that random ImageNet examples are linearly mixed up in the background to make the dataset realistic.

Thanks to the way the CDP dataset is constructed, the true MI under same-class sampling can be easily manipulated as shown in Figure 3. If only the color attribute is consistently chosen for each pair (Figure 3(a)), it corresponds to a downstream task whose class label is the color information and the positive pairing is denoted as $\mathcal{T}_{\text{class}}^{\text{color}}$. In this case, the true MI is $I_{\text{class}}(X;Y) = H(C_{\text{color}}) = 2$ bits. Similarly, $I_{\text{class}}(X;Y) = 2$ bits for Figure 3(b) and Figure 3(c). When all three attributes are consistently chosen for each pair (Figure 3(d)), it corresponds to a downstream task whose class label is the combination of color, digit, and position information. Then, the true MI is $I_{\text{class}}(X;Y) = H(C_{\text{color}}) + H(C_{\text{digit}}) + H(C_{\text{position}}) = 6$ bits. Note that the entropies add up because looking at one of the pair provides the exact information of the color, digit, and position of the other image.

# 3 Main Results

## 3.1 A small batch size is a limiting factor for MI estimation but not for contrastive learning.

**Existing belief 1:**

A small batch size is undesirable for contrastive learning because of InfoNCE's $\mathcal{O}(\log K)$ bound (Hjelm et al., 2018; Tian et al., 2020a; Bachman et al., 2019; Chen et al., 2020a; Sordoni et al., 2021; Wu et al., 2020; Song and Ermon, 2020).

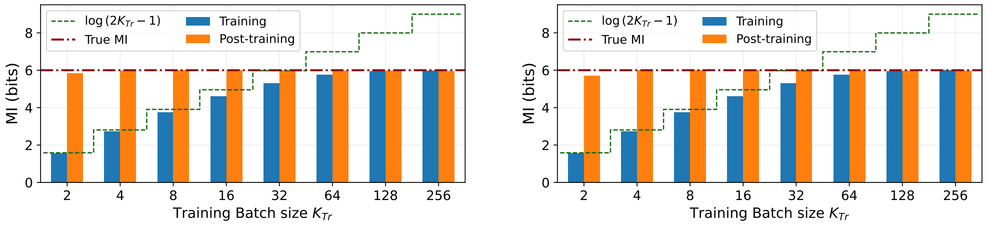
**Correction 1:**

A small batch size limits the training loss, but it limits neither the information in the learned representation nor the downstream-task performance.

It is a well-known fact that the estimated MI in Eq. (2) is upper bounded by $\log(2K_{\text{Tr}} - 1)$ (Oord et al., 2018; Sordoni et al., 2021; McAllester and Stratos, 2020; Poole et al., 2019), where $K_{\text{Tr}}$ is the batch size of training. See Supplementary A.3 for the derivation. Because of the bound, it has been often believed that a small batch size affects the contrastive learning negatively. To overcome this limitation, many of the previous works have increased the batch size (Hjelm et al., 2018; Tian et al., 2020a; Bachman et al., 2019) or have modified the InfoNCE loss (Sordoni et al., 2021; Wu et al., 2020; Song and Ermon, 2020). The existing works, however, have estimated MI concurrently during the training phase. Using CDP dataset and $\hat{I}_{\text{class}}(h_X; h_Y)$, we have performed two sets of experiments as shown in Figure 4. Even though the estimated MI with the training loss is limited by $\log(2K_{\text{Tr}} - 1)$, we can see that the post-training MI estimation is almost the same as the true MI (= 6 bits) and that the performance is over 96% for all the cases. Clearly, $\log(2K_{\text{Tr}} - 1)$ bound is not necessarily harmful and a small batch size does not limit the representation learning. We also note that $\hat{I}_{\text{class}}(h_X; h_Y)$ is almost identical to the ground-truth MI, i.e., $\hat{I}_{\text{class}}(h_X; h_Y) \approx 6\text{bits} = H(C)$. Thus, this result supports that the CDP dataset satisfies $I_{\text{class}}(X;Y) = H(C)$.

| $K_{\text{Tr}}$ | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|---|---|
| Acc. (%) | 98.8 | **99.8** | 99.7 | 99.7 | 99.6 | 99.4 | 99.4 | 99.2 |

| $K_{\text{Tr}}$ | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|---|---|
| Acc. (%) | 96.0 | **99.7** | **99.7** | **99.7** | 99.5 | 99.2 | 99.0 | 98.6 |



(a) ResNet-18

(b) ResNet-50

Figure 4: True MI, training MI (conventional estimating of MI at the time of training using the training loss), and our post-training MI. For CDP dataset, we train two ResNet models using $\hat{I}_{\text{class}}(h_X; h_Y)$ as the loss (as in Figure 2(b)). We evaluate $\hat{I}_{\text{class}}(h_X; h_Y)$ at the end of training (blue) and post-training (orange). During training, the MI is upper bounded by $\log(2K_{\text{Tr}} - 1)$ (dashed lines of green color). After the training is complete, the network is frozen and we evaluate the MI using a large batch size of $K_{\text{Est}} = 256$. Even though the training MI is limited by the $\log(2K_{\text{Tr}} - 1)$ bound, the post-training MI turns out to be almost the same as the true MI ($= 6$ bits). Obviously, the trained model can represent sufficiently large amount of information.

### 3.2 AUGMENTATION-BASED MI AND OTHER METRICS ARE NOT EFFECTIVE, BUT $\hat{I}_{\text{CLASS}}(h_X; h_Y)$ IS EFFECTIVE.

**Existing belief 2:**

- MI *cannot* measure how effective the representation is for the downstream task's performance (Tschannen et al., 2019).
- Instead, other metrics such as uniformity (Wang and Isola, 2020; Wang and Liu, 2021), alignment (Wang and Isola, 2020), tolerance (Wang and Liu, 2021), and linear CKA (Nguyen et al., 2020; Song et al., 2012; Nguyen et al., 2022) are more relevant and useful than MI.

**Correction 2:**

The only metric that is strongly relevant to the downstream-task performance is the MI of the downstream-task information itself.

The early contrastive learning studies (Oord et al., 2018; Hjelm et al., 2018; Bachman et al., 2019; Sordoni et al., 2021; Tian et al., 2020a) have regarded the minimization of InfoNCE loss to be equivalent to the maximization of MI. The existing belief in Section 3.1 is an example. Then, Tschannen et al. (2019) empirically showed that the estimated MI does not correlate well with the downstream-task performance. The analysis method in the work, however, was not rigorous in that only a particular choice of augmentation and the corresponding joint distribution $p_{\text{aug}}(x, y)$ were studied. Without addressing exactly what information is shared by $p_{\text{aug}}(x, y)$, the analysis can be quite misleading.

Subsequent works have suggested a variety of metrics to evaluate and explain the representation quality. Well-known metrics include alignment (Wang and Isola, 2020), uniformity (Wang and Isola, 2020; Wang and Liu, 2021), tolerance (Wang and Liu, 2021), and linear CKA (Nguyen et al., 2020; Song et al., 2012; Nguyen et al., 2022). The description of the metrics can be found in Supplementary B.3. While the suggested metrics have become popular because they are intuitive and enlightening, there has been no attempt to provide a comprehensive analysis on how reliable the metrics are.

To investigate the existing beliefs, we have designed an experiment where the representations of many pre-trained networks can be carefully compared. To better understand the existing beliefs, we have followed the previous works and examined the relationship between each metric and the downstream-task performance. The first experiment's results can be found in Table 1. By examining

Table 1: Post-training MI estimation results for ResNet-50 on ImageNet-100 and ImageNet-1k. Sixteen pre-trained models in Goyal et al. (2021); Wightman (2019); Khosla et al. (2020) are used to evaluate the effectiveness of $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ and $\hat{I}_{\text{class}}(h_X; h_Y)$.

| Algorithm | *ImageNet-100* | | | *ImageNet-1k* | | |
|---|---|---|---|---|---|---|
| | Acc. (%) | $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ | $\hat{I}_{\text{class}}(h_X; h_Y)$ | Acc. (%) | $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ | $\hat{I}_{\text{class}}(h_X; h_Y)$ |
| SupCon (Khosla et al., 2020) | 94.40 | 7.889 | 6.100 | 78.72 | 8.722 | 7.783 |
| Supervised pretrained | 93.00 | 7.598 | 5.816 | 74.11 | 8.378 | 6.761 |
| SwAV (Caron et al., 2020) | 92.52 | 8.541 | 5.560 | 74.78 | 9.428 | 6.214 |
| DeepCluster-v2 (Caron et al., 2020) | 92.38 | 8.540 | 5.559 | 73.65 | 9.416 | 6.232 |
| DINO (Caron et al., 2021) | 92.22 | 8.443 | 5.539 | 74.22 | 9.313 | 6.133 |
| Barlow Twins (Zbontar et al., 2021) | 90.80 | 8.528 | 5.513 | 72.82 | 9.407 | 6.157 |
| PIRL (Misra and Maaten, 2020) | 90.58 | 8.584 | 5.480 | 70.51 | 9.481 | 6.247 |
| SeLa-v2 (Caron et al., 2020) | 89.50 | 6.020 | 5.039 | 69.66 | 7.354 | 5.774 |
| SimCLR (Chen et al., 2020a) | 89.40 | 8.669 | 5.546 | 69.12 | 9.580 | 6.277 |
| MoCo-v2 (Chen et al., 2020b) | 87.54 | 8.592 | 5.490 | 63.89 | 9.499 | 6.221 |
| NPID++ (Misra and Maaten, 2020) | 79.60 | 8.190 | 4.792 | 56.60 | 9.009 | 4.692 |
| MoCo (He et al., 2020) | 76.94 | 8.338 | 4.904 | 47.05 | 9.155 | 4.907 |
| NPID (Wu et al., 2018) | 76.68 | 8.039 | 4.188 | 52.70 | 8.821 | 3.836 |
| ClusterFit (Yan et al., 2020) | 75.66 | 8.016 | 4.155 | 48.81 | 8.773 | 3.915 |
| RotNet (Gidaris et al., 2018) | 66.90 | 7.020 | 2.916 | 41.54 | 7.696 | 2.802 |
| Jigsaw (Noroozi and Favaro, 2016) | 56.74 | 6.339 | 2.510 | 30.85 | 7.155 | 2.583 |
| Pearson's correlation coefficient $\rho$ with Acc. | | 0.510 | 0.967 | | 0.535 | 0.943 |
| Kendall's rank correlation coefficient $\tau_K$ with Acc. | | 0.233 | 0.883 | | 0.233 | 0.617 |

Table 2: Summary of Pearson's correlation for seven scenarios. All of the popular metrics turned out to be ineffective for assessing downstream-task performance. For alignment and uniformity, smaller values indicate better representations so we flipped the signs.

| Encoder | Dataset | Metrics | | | | | |
|---|---|---|---|---|---|---|---|
| | | Alignment | Uniformity | Tolerance | Linear CKA | $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ | $\hat{I}_{\text{class}}(h_X; h_Y)$ |
| Is label information utilized? | | No | No | Yes | Yes | No | Yes |
| *Pearson's Correlation Coefficient $\rho$ with linear accuracy* | | | | | | | |
| ResNet-$\{18, 50\}$ | CDP | $-0.977$ | $-0.058$ | 0.956 | **0.992** | $-0.988$ | 0.990 |
| ResNet-$\{18, 50\}$ | CIFAR-10 | $-0.738$ | $-0.319$ | 0.121 | $-0.503$ | $-0.041$ | **0.634** |
| ResNet-$\{18, 50\}$ | ImageNet-100 | 0.165 | $-0.197$ | 0.214 | 0.410 | 0.085 | **0.805** |
| ResNet-50(Pretrained) | ImageNet-100 | 0.286 | 0.265 | $-0.227$ | 0.722 | 0.510 | **0.967** |
| ResNet-50(Pretrained) | ImageNet-1k | 0.175 | 0.157 | $-0.132$ | 0.451 | 0.535 | **0.943** |
| ViT(Pretrained) | ImageNet-100 | $-0.102$ | 0.623 | $-0.395$ | 0.856 | 0.721 | **0.974** |
| ViT(Pretrained) | ImageNet-1k | $-0.077$ | 0.561 | $-0.392$ | 0.203 | 0.783 | **0.977** |
| Average | | $-0.181$ | 0.147 | 0.021 | 0.447 | 0.229 | **0.899** |

Pearson's correlation and Kendall's rank correlation, the conclusion by Tschannen et al. (2019) can be confirmed for $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$. For $\hat{I}_{\text{class}}(h_X; h_Y)$ whose joint distribution is directly related to the downstream task's class label information, however, the MI correlates very well with the downstream-task performance. Therefore, we can see that it is misleading to say that MI in general does not correlate well with the downstream-task performance. Clearly, $\hat{I}_{\text{class}}(h_X; h_Y)$, the MI that is directly associated with the downstream task's class label information, correlates with the downstream-task performance very well.

The experiment was repeated for five other scenarios, and the summary of Pearson's correlation results can be found in Table 2. In the table, we are also showing the results for the other metrics. Surprisingly, none of the known metrics shows a high correlation. The only metric that consistently shows a high correlation is the $\hat{I}_{\text{class}}(h_X; h_Y)$, implying that the downstream-task information itself (i.e. class label information) is the only metric that correlates well with the downstream-task performance. Note that the class label information is also utilized by tolerance and linear CKA. So, they are also supervised metrics like $\hat{I}_{\text{class}}(h_X; h_Y)$, but they fail to achieve a high correlation. The full experimental results of all the scenarios can be found in Supplementary C.1.

**A short note on the recent theoretical bounds:** Same-class sampling has been also utilized in recent theoretical works where theoretical bounds are derived to connect contrastive learning and supervised learning (Arora et al., 2019; Nozawa and Sato, 2021; Ash et al., 2021; Bao et al., 2022). Unlike the practical and popular $\mathcal{T}_{\text{aug}}$, the supervised $\mathcal{T}_{\text{class}}$ provides strong structures and enables the deriving of meaningful results. All of the theoretical bounds, however, fail to correlate well with the downstream-task performance (see Table 7 in Supplementary C.1). Furthermore, contrastive training based on $\mathcal{T}_{\text{class}}$ does not guarantee a high performance as we will discuss in Section 4.1. Overall, the theoretical works are insightful, but somewhat disconnected from the practical issue of downstream-task performance.

### 3.3 Minimizing task-irrelevant information is not always necessary.

**Existing belief 3:**

For designing optimal views, task-irrelevant information needs to be discarded for a better generalization (Tian et al., 2020b; Tsai et al., 2020; Xiao et al., 2020; Chen et al., 2021a).

**Correction 3:**

Task-irrelevant information do not necessarily harm the generalization of the downstream task.

The choice of augmentation is known to determine which type of invariance will be learned during contrastive learning (Tian et al., 2020b; Tsai et al., 2020; Xiao et al., 2020; Chen et al., 2021a). Tian et al. (2020b) formalized this idea into the InfoMin principle: ‘a good set of views are those that share the minimal information necessary to perform well at the downstream task’.

We examine the belief with two experiments. In the first experiment, we have investigated the CDP dataset where $\mathcal{T}^{\text{color}}_{\text{class}}$, $\mathcal{T}^{\text{digit}}_{\text{class}}$, $\mathcal{T}^{\text{position}}_{\text{class}}$, and $\mathcal{T}^{\text{all}}_{\text{class}}$ are considered for training and $C_{\text{color}}$, $C_{\text{digit}}$, $C_{\text{position}}$, and $C_{\text{all}}$ are considered as the downstream task. The results are shown in Figure 5. As an example for ResNet-18, it can be seen that when $\mathcal{T}^{\text{digit}}_{\text{class}}$ is used for training, the performance for classifying color is 80.6%. If the InfoMin holds strongly, we would expect only the diagonal elements (same information for training and evaluation) to achieve a high performance. But the result shows that there are many non-diagonal elements that achieve a high performance. For instance, we can see that the performance of $C_{\text{digit}}$ is higher when $\mathcal{T}^{\text{all}}_{\text{class}}$ is used for training (99.2%; four types of information are retained in the representation) than when $\mathcal{T}^{\text{digit}}_{\text{class}}$ is used for training (98.9%). Post-training MI estimation results are provided in Supplementary C.2.

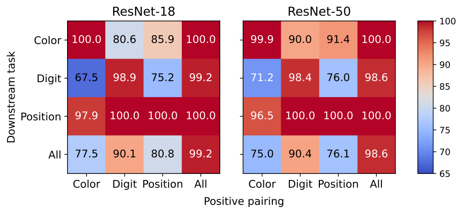


Figure 5: Linear evaluation performance of CDP dataset for task-dependent training. The task in $x$-axis indicates the positive pairing $\mathcal{T}$ used for training. The task in $y$-axis indicates the evaluated downstream task $C$.

There is another interesting topic that can be noticed from Figure 5. When a specific positive pairing is used for training (e.g. $\mathcal{T}^{\text{color}}_{\text{class}}$), we would expect only the corresponding information (e.g. $C_{\text{color}}$) to be learned in the representation. The results in Figure 5, however, show that a task-irrelevant information is frequently learned in the representation regardless of the positive pairing chosen for training. In particular, position information is always learned in our example. This indicates that targeting only for a specific type of information in contrastive learning might be quite challenging.

As the second experiment, we expand our experiment to two well-known augmentations of $\mathcal{T}_{\text{aug}}$. Following (Tian et al., 2020b), we utilize color jittering and random resized crop augmentations by varying the strength parameter. The details of the experiment and the results can be found in Supplementary C.3 where the same conclusions as in the first experiment can be arrived.

## 4 Discussion

### 4.1 Downstream task’s MI is an excellent performance metric, but it is not an effective learning objective.

Because we have shown that $\hat{I}_{\text{class}}(h_X; h_Y)$ is a very effective metric of downstream task’s linear evaluation performance (Section 3.2), it is reasonable to ask if the corresponding loss $\mathcal{L}_{\text{class}}$ in Figure 2(b) can learn a superior representation and achieve an excellent performance. A quick answer for this question is ‘no’. The experimental results with a full discussion can be found in Supplementary D.1. Surprisingly, a carefully designed unsupervised learning can outperform a supervised contrastive learning that is based on the downstream-task information only. Here, a careful design basically means a well-crafted augmentation where the augmentation may have been designed in a heuristic manner or through an extensive tuning.

Recently, the limitations of MI-based contrastive learning have been becoming clear. Many of the recent works have developed non-contrastive learning methods that can outperform MI-based contrastive learning (Caron et al., 2020; Grill et al., 2020; Zbontar et al., 2021). Even for contrastive learning, small modifications in the loss function have been shown to be useful (Yeh et al., 2021), indicating that the loss function's deviation from an exact MI formulation can be advantageous. In Supplementary D.2, we additionally show that the viewpoint of *Noise Contrastive Estimation* (NCE) in (Gutmann and Hyvärinen, 2010) can be more relevant for enhancing the performance of unsupervised representation learning than the viewpoint of InfoNCE.

While a high performing network must have its representation express the downstream-task information very well as we have shown in Section 3.2, the *training* of such a network requires additional learning signals regardless of the presence of the downstream-task information. All the cases discussed above strengthen the idea that MI of downstream task is an outstanding metric but clearly not an excellent learning objective. Finally, we would like to make it clear that our analysis and result are fundamentally different from the work of Tschannen et al. (2019). Please refer to Supplementary D.1.

### 4.2 Rethinking contrastive learning - is it really an unsupervised learning method?

If the only metric that is truly effective for predicting downstream task's performance is the downstream-task information itself as we have shown in Section 3.2, how is it possible to learn effective representations in an unsupervised way? First of all, it is crucial to recognize that the augmentation design is not completely unsupervised because the validation performance (linear evaluation performance) is used for the selection of augmentation design. The validation data does not directly affect the network parameters (i.e., no gradient descent with the validation data), but it indirectly affects the network parameters because the selection of augmentation design affects the joint distribution $p(x, y)$, in turn $p(x, y)$ defines the MI of the learning, and the MI affects the goal of learning as well as the learning dynamics.

The success of contrastive learning methods, and the closely related non-contrastive learning methods, seem to be due to two main reasons. First, compared to the early techniques such as pretext learning (Doersch et al., 2015; Pathak et al., 2016; Noroozi and Favaro, 2016; Gidaris et al., 2018), augmentation design can be successfully and efficiently completed within a limited design search space. Typically, effective augmentation techniques for supervised learning are already known for each application area, and properly combining the known techniques is a good start for achieving a high performance with an unsupervised contrastive learning. Second, the learned representation seems to generalize better than the traditional methods. This seems to be surprisingly true for the popular benchmark problems, but a careful study is still needed to confirm it for a wider set of applications and datasets.

Despite the amazing success of contrastive learning, it still remains open to develop a further advanced representation learning framework where a heuristic search of augmentation design per application area can be avoided.

## 5 Conclusion

In this work, we have examined three existing beliefs on mutual information in contrastive learning. For a rigorous investigation, we made use of same-class sampling, post-training MI estimation, and CDP dataset. We have empirically shown that the existing beliefs are incorrect or misleading, and provided adequate corrections. We discussed how maximizing the MI of downstream task's information is necessary but not sufficient for an unsupervised representation learning. A limitation of our study is that we have focused on image classification as the only downstream task. Our study can be extended to other downstream tasks such as object detection and to other datasets such as NLP datasets.

## REFERENCES

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

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

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180, 2019.

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

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

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

Alessandro Sordoni, Nouha Dziri, Hannes Schulz, Geoff Gordon, Philip Bachman, and Remi Tachet Des Combes. Decomposed mutual information estimation for contrastive representation learning. In *International Conference on Machine Learning*, pages 9859–9869. PMLR, 2021.

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

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037*, 2020.

Jiaming Song and Stefano Ermon. Multi-label contrastive predictive coding. *arXiv preprint arXiv:2007.09852*, 2020.

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

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2021.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(5), 2012.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. On the origins of the block structure phenomenon in neural network representations. *arXiv preprint arXiv:2202.07184*, 2022.

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*, 2020b.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*, 2020.

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

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34, 2021a.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

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

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

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

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.

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.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884, 2020.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Kento Nozawa and Issei Sato. Understanding negative samples in instance discriminative self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Dipendra Misra. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

Han Bao, Yoshihiro Nagano, and Kento Nozawa. On the surrogate gap between contrastive and supervised losses. In *International Conference on Machine Learning*, pages 1585–1606. PMLR, 2022.

Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. Vissl. `https://github.com/facebookresearch/vissl`, 2021.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

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

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018.

Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6509–6518, 2020.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

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

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

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*, pages 1422–1430, 2015.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021b.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Hongkai Xiong, and Qi Tian. Seed the views: Hierarchical semantic alignment for contrastive representation learning. *arXiv preprint arXiv:2012.02733*, 2020.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

# Supplementary materials for the paper "Correcting Three Existing Beliefs on Mutual Information in Contrastive Learning"

## A Proofs

### A.1 Proof of $I_{\text{CLASS}}(X;Y) \leq H(C)$

*Proof.* From the construction of same-class sampling, $X \leftarrow C \rightarrow Y$. The dependency is Markov equivalent to $X \rightarrow C \rightarrow Y$. Then,

$$I(X;Y) \leq I(X;C) = H(C) - H(C|X) \leq H(C)$$

where the first inequality follows from the data processing inequality and the second inequality follows from the positiveness of entropy for the case of discrete random variable $C$. □

### A.2 Proof of $I_{\text{CLASS}}(X;Y) = H(C)$ when an error-free classification function $f_{class}(\cdot)$ exists

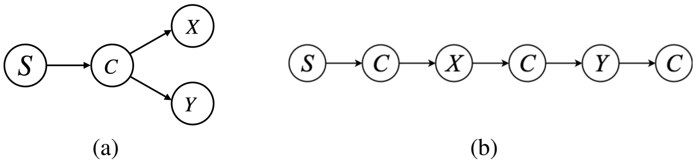


Figure 6: The Markov process of same-class-sampling, $\mathcal{T}_{\text{class}}$. $S$ denotes the anchor image, $C$ denotes the image's downstream-task class label, and $X$ and $Y$ correspond to the positive pair chosen for same-class-sampling of the image $S$. (a) The original Markov process of same-class-sampling. (b) An equivalent Markov process of the same-class-sampling.

*Proof.* The Markov dependency of $\mathcal{T}_{\text{class}}$ can be summarized as shown in Figure 6(a). For the same-class-sampling, $C$ is the common class label of $S$, $X$ and $Y$ (i.e., $c_i = f_{class}(s_i) = f_{class}(x_i) = f_{class}(y_i)$, where $f_{class}(\cdot)$ is a function that returns the class label information). Then, $H(C|X) = 0$ and $H(C|Y) = 0$ due to the deterministic nature of each image's class label and the below directly follows.

$$I(X;C) = H(C) - H(C|X) = H(C) \tag{3}$$

$$I(Y;C) = H(C) - H(C|Y) = H(C) \tag{4}$$

Because $C$ can be perfectly determined from either $X$ or $Y$, the Markov process in Figure 6(a) can be alternatively expressed as $S \rightarrow C \rightarrow X \rightarrow C \rightarrow Y \rightarrow C$ as shown in Figure 6(b). Here, the first part of the new Markov process is the same as in Figure 6(a): we start from $s_i$, read its class label $c_i = f_{class}(s_i)$, and sample an example $x_i$ using the class label $c_i$. In Figure 6(b), however, we can alternatively read $x_i$'s class label without any uncertainty to recover $c_i = f_{class}(x_i) = f_{class}(s_i)$ and then use the class label to sample $y_i$. Because $c_i = f_{class}(y_i)$ can be recovered from $y_i$, the last part of dependency, $Y \rightarrow C$, follows. For the equivalent Markov process in Figure 6(b), we derive an upper bound and a lower bound to complete the proof.

**Upper bound:** We apply data processing inequality Cover (1999) to the Markov dependency $X \rightarrow C \rightarrow Y$ in the middle part of Figure 6(b).

$$I(X;Y) \leq I(X;C) \tag{5}$$

$$= H(C) - H(C|X) \tag{6}$$

$$= H(C) \tag{7}$$

Eq. (5) is the data processing inequality, Eq. (6) is from the definition of MI, and Eq. (7) is because of $H(C|X) = 0$ as in Eq. (3).

**Lower bound:** We apply data processing inequality Cover (1999) to the Markov dependency $C \rightarrow X \rightarrow C \rightarrow Y \rightarrow C$ part of Figure 6(b). The following directly follows from the data processing inequality.

$$\begin{aligned} &I(C;C) \leq I(X;Y) \tag{8} \\ \Rightarrow &H(C) \leq I(X;Y) \tag{9} \end{aligned}$$

Note that we have $C$ in the beginning and at the end of the Markov dependency. The first $C$ in $I(C;C)$ corresponds to the $C$ in the beginning and the second $C$ in $I(C;C)$ corresponds to the $C$ at the end of the Markov dependency. Eq. (9) is because $I(C;C)$ is the self-information that is the same as $H(C)$.

Therefore, the true mutual information value of same-class-sampling, $I_{\text{class}}(X;Y)$, is the same as the class label's entropy, $H(C)$. □

### A.3 Proof of $\hat{I}(X;Y) \leq \log{(2K-1)}$

The inequality is well-known as described in (Oord et al., 2018). We provide the derivation for a typical augmentation (SimCLR-like augmentation) to make it clear that $\log{(2K-1)}$, instead of $\log{(K)}$, is due to the number of terms in the denominator.

$K$ is the batch size and $q(x|y) = \frac{p(x)}{Z(y)} e^{\text{sim}(f(x),f(y))/\tau}$, where $Z(y) = \mathbb{E}_{p(y)}[e^{\text{sim}(f(x),f(y))/\tau}]$; $f = f_p \circ f_e$, where $f_e$ is the encoder network and $f_p$ is the projection head; $\text{sim}(u,v) = u^T v/||u||||v||$ denotes the dot product between $l_2$ normalized $u$ and $v$ (i.e. cosine similarity); and $\tau$ denotes a temperature parameter.

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x|y)}{p(x)}\right] \tag{10} \\ &= \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] + \mathbb{E}_{p(y)}[KL(p(x|y)||q(x|y))] \tag{11} \\ &\geq \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] \tag{12} \\ &= \mathbb{E}_{p(x,y)}\left[\log \frac{e^{\text{sim}(f(x),f(y))/\tau}}{Z(y)}\right] \tag{13} \\ &\approx \mathbb{E}\left[\log \frac{e^{\text{sim}(f(x_i),f(y_i))/\tau}}{\frac{1}{2K-1}\sum_{j=1}^{2K} \mathbb{1}_{[j \neq i]} e^{\text{sim}(f(x_i),f(y_j))/\tau}}\right] \tag{14} \\ &= \log{(2K-1)} + \mathbb{E}\left[\log \frac{e^{\text{sim}(f(x_i),f(y_i))/\tau}}{\sum_{j=1}^{2K} \mathbb{1}_{[j \neq i]} e^{\text{sim}(f(x_i),f(y_j))/\tau}}\right] \tag{15} \\ &= \log{(2K-1)} - \mathcal{L} \tag{16} \\ &\triangleq \hat{I}(X;Y) \tag{17} \end{aligned}$$

Inequality in Eq. (12) is due to the non-negativeness of KL-divergence and the approximation in Eq. (14) is due to the replacement of the expectation with its empirical mean. Finally, the negative loss $-\mathcal{L}(x_i)$ in Eq. (16) is always negative because the argument of the second $\log$ term in Eq. (15) is always between zero and one. Therefore, $\hat{I}(X;Y) \leq \log{(2K-1)}$.

# B Implementation details

## B.1 SimCLR augmentation ($\mathcal{T}_{\text{SimCLR}}$)

As a representative case of unsupervised positive pairing $\mathcal{T}_{\text{aug}}$, we adopt the SimCLR augmentation (Chen et al., 2020a). The details of the code implementation of each dataset are provided here. We use PyTorch and torchvision library.

### B.1.1 CDP dataset

```
img_size = 32; strength = 0.5
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(size=img_size),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    torchvision.transforms.ToTensor()])
```

For Table 12, We empirically found the $\mathcal{T}_{\text{aug}}$ shown below by searching for the performance.

```
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(
        size=img_size, scale=(0.5, 1.0)),
    torchvision.transforms.RandomApply([color_jitter], p=0.5),
    torchvision.transforms.ToTensor()])
```

### B.1.2 CIFAR-10

```
img_size = 32; strength = 0.5
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform_train = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(size=img_size),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize(
        mean=[0.4914, 0.4822, 0.4465], std=[0.2023, 0.1994, 0.2010])])
```

### B.1.3 ImageNet

```
img_size = 224; strength = 1.; ksize = 23
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(
        size=img_size, scale=(0.2, 1.0)),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    GaussianBlur(kernel_size=ksize),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize(
        mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])])
```

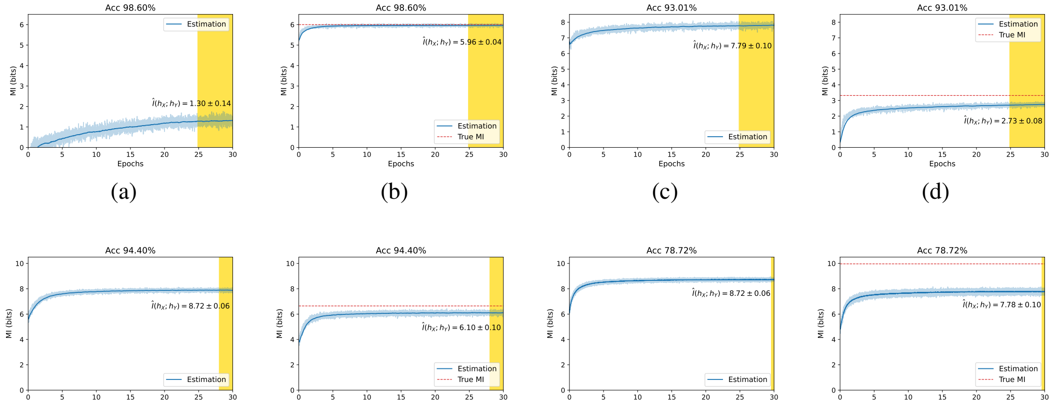
## B.2 HYPERPARAMTERS

### B.2.1 TRAINING

In our study, we train the encoder $f_e(\cdot)$ of ResNet-18 and ResNet-50 and the projection head $f_p(\cdot)$ of 2-layer MLP with batch normalization for 100 epochs. We set the batch size $K_{\text{Tr}}$ as 256 for CDP and CIFAR-10, and 128 for ImageNet-100 and ImageNet-1k. We set the temperature scalar $\tau$ as 0.5 for CIFAR-10 and 0.2 for other datasets. We optimize the InfoNCE loss using SGD with learning rate of 0.001 and weight decay of $1e^{-4}$ for CDP and CIFAR-10, and with learning rate of 0.4 and weight decay of 0.00002 for ImageNet. We also use linear warm-up for the first 3 epochs (10 for ImageNet), and decay the learning rate with the cosine decay schedule without restarts (Loshchilov and Hutter, 2016; Goyal et al., 2017). We carried out all the experiments using PyTorch on a single Nvidia RTX 3090 GPU.

### B.2.2 POST-TRAINING MI ESTIMATION

The critic $f_c(\cdot)$ can be flexibly chosen as explained in (Poole et al., 2019; Song and Ermon, 2019), but we set it identical in architecture and hyperparameters as the projection head $f_p(\cdot)$ of the training stage. The estimation is performed with the epoch size of 30. We have chosen the epoch size based on the learning curves of a variety of post-training MI estimation results shown in Figure 7. We empirically found that 30 is sufficiently large for the estimations to converge. MI estimation aims to maximize the lower bound of MI, and we define the final estimated MI as the average of the last 1000 steps (as highlighted in the figures) to deal with the estimation variance. To prevent the $\log(2K_{\text{Est}} - 1)$ becoming a limiting factor of the MI estimation, we have chosen the MI estimation batch size $K_{\text{Est}}$ to be sufficiently large. We set $K_{\text{Est}}$ as 256 for CDP and CIFAR-10 and 512 for ImageNet-100 and ImageNet-1k. Note that $K_{\text{Est}}$ is independently chosen from $K_{\text{Tr}}$, the batch size of training. Unlike the training stage, MI estimation is not affected by the temperature scalar $\tau$, and we set $\tau = 0.1$ throughout our study.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Examples of post-training MI estimation: (a) CDP, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (b) CDP, $\hat{I}_{\text{class}}(h_X; h_Y)$, (c) CIFAR-10, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (d) CIFAR-10, $\hat{I}_{\text{class}}(h_X; h_Y)$, (e) ImageNet-100, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (f) ImageNet-100, $\hat{I}_{\text{class}}(h_X; h_Y)$, (g) ImageNet-1k, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (h) ImageNet-1k, $\hat{I}_{\text{class}}(h_X; h_Y)$. Note that the MI estimation $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ in (a) is relatively smaller when compared to the $\hat{I}_{\text{class}}(h_X; h_Y)$ in (b). This is an example where $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ does not properly reflect the downstream-task performance while $\hat{I}_{\text{class}}(h_X; h_Y)$ does.

### B.3 REPRESENTATION EVALUATION METRICS

The metrics in Table 2 are summarized below. For the implementation, we either adopt the authors' code (Wang and Isola, 2020) or implement it by ourselves based on the equations in the paper (Wang and Liu, 2021; Nguyen et al., 2020).

**Alignment (Wang and Isola, 2020):** expected distance between positive pairs defined by $\mathcal{T}_{\text{aug}}$. Two views of positive pair should be mapped to nearby features, and thus be (mostly) invariant to unneeded noise factors. Representations are more aligned when the metric is smaller.

$$\text{Alignment} = \mathbb{E}_{(x,y)\sim p_{\text{pos}}}\left[||f(x) - f(y)||_2^{\alpha}\right]$$

**Uniformity (Wang and Isola, 2020):** the logarithm of the average pairwise Gaussian potential. Feature vectors should be roughly uniformly distributed on the unit hypersphere, preserving as much information of the data as possible. Representations are more uniform when the metric is smaller.

$$\text{Uniformity} = \log \mathbb{E}_{x,y\sim p_{\text{data}}}\left[e^{-t||f(x)-f(y)||_2^2}\right]$$

**Tolerance (Wang and Liu, 2021):** mean similarity of samples of the same class. It utilize the supervised information. Representations are more tolerant when the metric is higher.

$$\text{Tolerance} = \mathbb{E}_{x,y\sim p_{\text{data}}}\left[(f(x)^T f(y)) \cdot \mathbb{1}_{c_x = c_y}\right]$$

**Linear CKA (Centered Kernel Alignment) (Nguyen et al., 2020; Song et al., 2012; Nguyen et al., 2022):** the similarity between pairs of representations. We adopt the minibatch estimators and set the batch size as 200. Representations are more similar when the metric is higher. It is defined as

$$\text{Linear CKA} = \frac{1}{n(n-3)}\left(\text{tr}(\tilde{K}\tilde{L}) + \frac{\mathbf{1}^T\tilde{K}\mathbf{1}\mathbf{1}^T\tilde{L}\mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^T\tilde{K}\tilde{L}\mathbf{1}\right),$$

where $K = XX^T$, $L = YY^T$, $\tilde{K}$ and $\tilde{L}$ are obtained by setting the diagonal entries of $K$ and $L$ to zero, and X and Y denote the representation matrix for each view. This metric is not directly used to evaluate contrastive learning, and we assume $(x, y) \sim p_{\text{class}}(x, y)$. Therefore, it also utilizes the supervised information.

# C Additional results

## C.1 Full results of Table 2

In Table 2, the summary of seven experiments is provided. Here, we report the full results of the seven experiments. For alignment and uniformity, a smaller value is better (↓). For tolerance and linear CKA, a higher value is better (↑). Note that class label information is utilized by tolerance, linear CKA, and $\hat{I}_{\text{class}}(h_X; h_Y)$.

Table 3: Summary of seven experiments. Except for $\hat{I}_{\text{class}}(h_X; h_Y)$, all the other known metrics turn out to be ineffective for assessing downstream-task performance. For alignment and uniformity, smaller values indicate better representations, so we flipped the signs. Note that the label information is also utilized by tolerance and linear CKA.

| Encoder | Dataset | Metrics | | | | | |
|---|---|---|---|---|---|---|---|
| | | Alignment | Uniformity | Tolerance | Linear CKA | $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ | $\hat{I}_{\text{class}}(h_X; h_Y)$ |
| Is label information utilized? | | No | No | Yes | Yes | No | Yes |
| *Pearson's Correlation Coefficient ρ with linear accuracy* | | | | | | | |
| ResNet-$\{18, 50\}$ | CDP | −0.977 | −0.058 | 0.956 | **0.992** | −0.988 | 0.990 |
| ResNet-$\{18, 50\}$ | CIFAR-10 | −0.738 | −0.319 | 0.121 | −0.503 | −0.041 | **0.634** |
| ResNet-$\{18, 50\}$ | ImageNet-100 | 0.165 | −0.197 | 0.214 | 0.410 | 0.085 | **0.805** |
| ResNet-50(Pretrained) | ImageNet-100 | 0.286 | 0.265 | −0.227 | 0.722 | 0.510 | **0.967** |
| ResNet-50(Pretrained) | ImageNet-1k | 0.175 | 0.157 | −0.132 | 0.451 | 0.535 | **0.943** |
| ViT(Pretrained) | ImageNet-100 | −0.102 | 0.623 | −0.395 | 0.856 | 0.721 | **0.974** |
| ViT(Pretrained) | ImageNet-1k | −0.077 | 0.561 | −0.392 | 0.203 | 0.783 | **0.977** |
| Average | | −0.181 | 0.147 | 0.021 | 0.447 | 0.229 | **0.899** |
| *Kendall's Rank Correlation Coefficient $\tau_K$ with linear accuracy* | | | | | | | |
| ResNet-$\{18, 50\}$ | CDP | −0.545 | 0.061 | 0.485 | 0.333 | −0.727 | **0.545** |
| ResNet-$\{18, 50\}$ | CIFAR-10 | −0.600 | −0.067 | 0.333 | −0.467 | −0.067 | **0.467** |
| ResNet-$\{18, 50\}$ | ImageNet-100 | −0.200 | 0.333 | −0.067 | **0.467** | 0.067 | **0.467** |
| ResNet-50(Pretrained) | ImageNet-100 | 0.293 | 0.008 | 0.092 | 0.410 | 0.233 | **0.883** |
| ResNet-50(Pretrained) | ImageNet-1k | 0.109 | −0.059 | 0.109 | 0.243 | 0.233 | **0.617** |
| ViT(Pretrained) | ImageNet-100 | −0.033 | 0.253 | −0.055 | 0.626 | 0.516 | **0.802** |
| ViT(Pretrained) | ImageNet-1k | 0.030 | 0.364 | −0.061 | 0.152 | 0.576 | **0.848** |
| Average | | −0.135 | 0.128 | 0.119 | 0.252 | 0.119 | **0.661** |

### C.1.1 CDP, CIFAR-10, ImageNet-100 with three different temperatures

For CDP, CIFAR-10, and ImageNet-100, we train the encoders of ResNet-18/50 from scratch following the setups in Supplementary B.2.1. We test three temperature parameters for each dataset. The results are shown below.

Table 4: Metrics evaluated on CDP dataset. For CDP dataset, we could not achieve the good performance based on $\mathcal{L}_{\text{SimCLR}}$, we also investigate the results when we use $\mathcal{L}_{\text{class}}$ during contrastive learning.

| Model | Training loss | Temperature | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X; Y)$ | $\hat{I}_{\text{class}}(X; Y)$ |
|---|---|---|---|---|---|---|---|---|---|
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.1 | 42.64 | 0.196 | -0.941 | 0.776 | 0.056 | 5.960 | 3.179 |
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.2 | 46.27 | 0.227 | -1.074 | 0.743 | 0.052 | 5.387 | 3.060 |
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.3 | 49.90 | 0.218 | -1.041 | 0.747 | 0.017 | 4.938 | 2.957 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.1 | 44.45 | 0.184 | -0.685 | 0.834 | 0.005 | 5.663 | 3.374 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.2 | 50.01 | 0.226 | -0.784 | 0.820 | 0.051 | 5.107 | 3.547 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.3 | 46.80 | 0.214 | -0.723 | 0.829 | 0.000 | 4.498 | 2.936 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.1 | 99.15 | 0.357 | -0.810 | 0.978 | 0.869 | 0.890 | 5.970 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.2 | 99.26 | 0.404 | -0.989 | 0.988 | 0.897 | 0.890 | 5.970 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.3 | 99.13 | 0.388 | -0.995 | 0.990 | 0.912 | 0.946 | 5.956 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.1 | 98.60 | 0.396 | -0.689 | 0.978 | 0.909 | 1.298 | 5.966 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.2 | 97.21 | 0.402 | -0.806 | 0.987 | 0.934 | 0.910 | 5.976 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.3 | 93.04 | 0.423 | -0.849 | 0.989 | 0.937 | 0.950 | 5.857 |
| Pearson's $\rho$ with Acc. | | | | 0.977 | 0.058 | 0.956 | **0.992** | -0.988 | 0.990 |
| Kendall's $\tau_K$ with Acc. | | | | 0.545 | -0.061 | 0.485 | 0.333 | -0.727 | **0.545** |

Table 5: Metrics evaluated on CIFAR-10 dataset.

| Model | Temperature | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-18 | 0.1 | 90.03 | 0.380 | -2.735 | 0.399 | 0.287 | 8.063 | 2.717 |
| ResNet-18 | 0.3 | 91.11 | 0.449 | -3.147 | 0.321 | 0.293 | 7.912 | 2.874 |
| ResNet-18 | 0.5 | 90.97 | 0.427 | -2.839 | 0.427 | 0.452 | 7.730 | 2.756 |
| ResNet-50 | 0.1 | 92.06 | 0.403 | -2.351 | 0.458 | 0.227 | 8.117 | 2.806 |
| ResNet-50 | 0.3 | 92.97 | 0.562 | -2.950 | 0.328 | 0.224 | 7.954 | 2.902 |
| ResNet-50 | 0.5 | 93.01 | 0.467 | -2.432 | 0.467 | 0.267 | 7.879 | 2.803 |
| Pearson's $\rho$ with Acc. | | | 0.738 | 0.319 | 0.121 | -0.503 | -0.041 | **0.634** |
| Kendall's $\tau_K$ with Acc. | | | 0.600 | 0.067 | 0.333 | -0.467 | -0.067 | **0.467** |

Table 6: Metrics evaluated on ImageNet-100 dataset.

| Model | Temperature | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-18 | 0.1 | 72.60 | 0.291 | -2.265 | 0.497 | 0.329 | 8.364 | 3.394 |
| ResNet-18 | 0.2 | 76.42 | 0.352 | -2.593 | 0.438 | 0.375 | 8.375 | 3.907 |
| ResNet-18 | 0.3 | 75.66 | 0.315 | -2.268 | 0.515 | 0.405 | 8.313 | 3.857 |
| ResNet-50 | 0.1 | 74.08 | 0.038 | -0.270 | 0.941 | 0.272 | 8.412 | 3.967 |
| ResNet-50 | 0.2 | 75.52 | 0.037 | -0.277 | 0.943 | 0.332 | 8.347 | 4.186 |
| ResNet-50 | 0.3 | 77.80 | 0.056 | -0.408 | 0.914 | 0.338 | 8.403 | 4.263 |
| Pearson's $\rho$ with Acc. | | | -0.165 | 0.197 | 0.214 | 0.410 | 0.085 | **0.805** |
| Kendall's $\tau_K$ with Acc. | | | 0.200 | -0.333 | -0.067 | **0.467** | 0.067 | **0.467** |

Table 7: Theoretical upper bounds of the supervised loss for CDP dataset. All the bounds are determined based on the same variables, including the batch size, the number of class, and the contrastive loss. Since we fix the batch size and the number of classes, only the contrastive loss affects the bounds. Thus, all bounds have the same correlation coefficient of $\rho = -0.409$ and $\tau_K = -0.182$. We follow the official implementation codes of (Bao et al., 2022).

| Model | Training loss | Temperature | Acc. (%) | (Arora et al., 2019) | (Nozawa and Sato, 2021) | (Ash et al., 2021) | (Bao et al., 2022) |
|---|---|---|---|---|---|---|---|
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.1 | 42.64 | -399.448 | 1.931 | -911.233 | 0.807 |
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.2 | 46.27 | 4830.149 | 5.277 | 11018.672 | 2.447 |
| ResNet-18 | $\mathcal{L}_{\text{SimCLR}}$ | 0.3 | 49.90 | 7315.527 | 6.867 | 16688.383 | 3.226 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.1 | 44.45 | 213.227 | 2.323 | 486.419 | 0.999 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.2 | 50.01 | 5250.332 | 5.546 | 11977.204 | 2.579 |
| ResNet-50 | $\mathcal{L}_{\text{SimCLR}}$ | 0.3 | 46.80 | 7553.165 | 7.019 | 17230.492 | 3.301 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.1 | 99.15 | -159.413 | 2.084 | -363.658 | 0.883 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.2 | 99.26 | 1089.677 | 2.883 | 2485.801 | 1.274 |
| ResNet-18 | $\mathcal{L}_{\text{class}}$ | 0.3 | 99.13 | 3778.281 | 4.604 | 8619.119 | 2.117 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.1 | 98.60 | -167.034 | 2.079 | -381.042 | 0.880 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.2 | 97.21 | 1133.557 | 2.911 | 2585.902 | 1.288 |
| ResNet-50 | $\mathcal{L}_{\text{class}}$ | 0.3 | 93.04 | 4061.839 | 4.785 | 9265.980 | 2.206 |

### C.1.2 Evaluations over pre-trained encoders: ImageNet-100 and ImageNet-1k

We additionally test a variety of pre-trained models loaded from (Goyal et al., 2021; Khosla et al., 2020; Wightman, 2019). We inspect 16 pre-trained ResNet-50 models and 14 pre-trained ViT models. All models are pre-trained by ImageNet-1k dataset. We load the pre-trained models and evaluate the linear accuracy and the metrics. The results are shown below.

Table 8: Metrics evaluated on ImageNet-100 dataset using pre-trained ResNet-50 models.

| Algorithm | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|
| SupCon (Khosla et al., 2020) | 94.40 | 0.107 | -2.600 | 0.489 | 0.439 | 7.889 | 6.100 |
| Supervised pretrained | 93.00 | 0.701 | -3.173 | 0.380 | 0.403 | 7.598 | 5.816 |
| SwAV (Caron et al., 2020) | 92.52 | 0.296 | -1.659 | 0.636 | 0.282 | 8.544 | 5.560 |
| DeepCluster-v2 (Caron et al., 2020) | 92.38 | 0.244 | -1.308 | 0.709 | 0.254 | 8.544 | 5.560 |
| DINO (Caron et al., 2021) | 92.22 | 0.433 | -1.829 | 0.592 | 0.277 | 8.443 | 5.539 |
| Barlow Twins (Zbontar et al., 2021) | 90.80 | 0.477 | -2.415 | 0.458 | 0.316 | 8.528 | 5.513 |
| PIRL (Misra and Maaten, 2020) | 90.58 | 0.388 | -3.387 | 0.361 | 0.452 | 8.584 | 5.480 |
| SeLa-v2 (Caron et al., 2020) | 89.50 | 0.208 | -1.098 | 0.752 | 0.302 | 6.020 | 5.039 |
| SimCLR (Chen et al., 2020a) | 89.40 | 0.519 | -3.032 | 0.336 | 0.425 | 8.669 | 5.546 |
| MoCo-v2 (Chen et al., 2020b) | 87.54 | 0.321 | -2.820 | 0.497 | 0.413 | 8.592 | 5.490 |
| NPID++ (Misra and Maaten, 2020) | 79.60 | 0.745 | -2.637 | 0.423 | 0.303 | 8.190 | 4.792 |
| MoCo (He et al., 2020) | 76.94 | 0.701 | -3.174 | 0.380 | 0.403 | 8.338 | 4.904 |
| NPID (Wu et al., 2018) | 76.68 | 0.745 | -2.637 | 0.423 | 0.201 | 8.039 | 4.188 |
| ClusterFit (Yan et al., 2020) | 75.66 | 0.706 | -3.019 | 0.321 | 0.199 | 8.016 | 4.155 |
| RotNet (Gidaris et al., 2018) | 66.90 | 0.625 | -1.927 | 0.561 | 0.166 | 7.020 | 2.916 |
| Jigsaw (Noroozi and Favaro, 2016) | 56.74 | 0.220 | -0.486 | 0.888 | 0.076 | 6.339 | 2.510 |
| Pearson's $\rho$ with Acc. | | -0.286 | -0.265 | -0.227 | 0.722 | 0.510 | **0.967** |
| Kendall's $\tau_K$ with Acc. | | -0.293 | -0.008 | 0.092 | 0.410 | 0.233 | **0.883** |

Table 9: Metrics evaluated on ImageNet-1k dataset using pre-trained ResNet-50 models.

| Algorithm | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|
| SupCon (Khosla et al., 2020) | 78.72 | 0.697 | -2.560 | 0.479 | 0.302 | 8.722 | 7.783 |
| Supervised pretrained | 74.11 | 0.704 | -3.169 | 0.369 | 0.373 | 8.378 | 6.761 |
| SwAV (Caron et al., 2020) | 74.78 | 0.298 | -1.637 | 0.634 | 0.228 | 9.428 | 6.214 |
| DeepCluster-v2 (Caron et al., 2020) | 73.65 | 0.247 | -1.284 | 0.708 | 0.177 | 9.416 | 6.232 |
| DINO (Caron et al., 2021) | 74.22 | 0.434 | -1.802 | 0.590 | 0.225 | 9.313 | 6.133 |
| Barlow Twins (Zbontar et al., 2021) | 72.82 | 0.485 | -2.394 | 0.454 | 0.240 | 9.407 | 6.157 |
| PIRL (Misra and Maaten, 2020) | 70.51 | 0.400 | -3.378 | 0.345 | 0.375 | 9.481 | 6.247 |
| SeLa-v2 (Caron et al., 2020) | 69.66 | 0.209 | -1.064 | 0.756 | 0.218 | 7.354 | 5.774 |
| SimCLR (Chen et al., 2020a) | 69.12 | 0.536 | -2.991 | 0.329 | 0.397 | 9.580 | 6.277 |
| MoCo-v2 (Chen et al., 2020b) | 63.89 | 0.333 | -2.801 | 0.480 | 0.399 | 9.499 | 6.221 |
| NPID++ (Misra and Maaten, 2020) | 56.60 | 0.845 | -2.634 | 0.335 | 0.289 | 9.009 | 4.692 |
| MoCo (He et al., 2020) | 47.052 | 0.704 | -3.169 | 0.369 | 0.373 | 9.155 | 4.907 |
| NPID (Wu et al., 2018) | 52.70 | 0.761 | -2.634 | 0.417 | 0.192 | 8.821 | 3.836 |
| ClusterFit (Yan et al., 2020) | 48.81 | 0.710 | -3.004 | 0.313 | 0.171 | 8.773 | 3.915 |
| RotNet (Gidaris et al., 2018) | 41.54 | 0.627 | -1.913 | 0.553 | 0.143 | 7.696 | 2.802 |
| Jigsaw (Noroozi and Favaro, 2016) | 30.85 | 0.221 | -0.479 | 0.888 | 0.091 | 7.155 | 2.583 |
| Pearson's $\rho$ with Acc. | | -0.175 | -0.157 | -0.132 | 0.451 | 0.535 | **0.943** |
| Kendall's $\tau_K$ with Acc. | | -0.109 | 0.059 | 0.109 | 0.243 | 0.233 | **0.617** |

Table 10: Metrics evaluated on ImageNet-100 dataset using pre-trained ViT models.

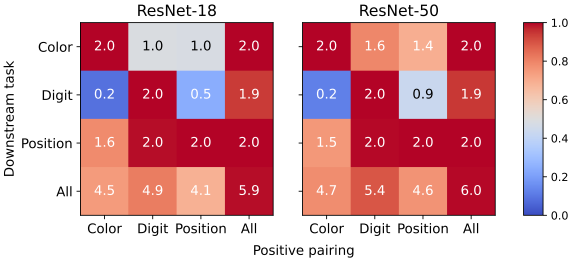
| Algorithm | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|
| Swin-B (Liu et al., 2021) | 96.20 | 0.787 | -3.663 | 0.502 | 0.559 | 8.073 | 6.222 |
| Supervised pretrained (ViT-B/16) (Dosovitskiy et al., 2021) | 95.36 | 0.565 | -3.843 | 0.450 | 0.538 | 8.252 | 5.977 |
| PiT-B (Heo et al., 2021) | 94.62 | 0.880 | -3.694 | 0.497 | 0.520 | 7.895 | 6.398 |
| DeiT (ViT-B/16) (Touvron et al., 2021a) | 94.30 | 0.833 | -3.761 | 0.507 | 0.499 | 7.799 | 6.287 |
| CaiT (XXS-36/16) (Touvron et al., 2021b) | 93.90 | 0.644 | -3.745 | 0.566 | 0.414 | 7.492 | 5.795 |
| PiT-S (Heo et al., 2021) | 93.76 | 0.820 | -3.763 | 0.491 | 0.448 | 7.664 | 6.151 |
| DeiT (ViT-S/16) (Touvron et al., 2021a) | 93.42 | 0.789 | -3.774 | 0.513 | 0.436 | 7.435 | 6.021 |
| CaiT (XXS-24/16) (Touvron et al., 2021b) | 93.28 | 0.662 | -3.784 | 0.532 | 0.379 | 7.488 | 5.690 |
| MoCo(v3) (ViT-B/16) (Chen et al., 2021b) | 93.12 | 0.130 | -1.275 | 0.796 | 0.390 | 8.594 | 5.654 |
| DINO (ViT-B/16) (Caron et al., 2021) | 92.84 | 0.408 | -3.610 | 0.475 | 0.510 | 8.454 | 5.675 |
| Supervised pretrained (ViT-S/16) (Dosovitskiy et al., 2021) | 92.70 | 0.886 | -3.482 | 0.505 | 0.528 | 6.863 | 5.515 |
| DeiT (ViT-T/16) (Touvron et al., 2021a) | 90.12 | 0.797 | -3.813 | 0.471 | 0.336 | 7.186 | 5.365 |
| Supervised pretrained (ViT-T/16) (Dosovitskiy et al., 2021) | 80.14 | 1.047 | -3.211 | 0.438 | 0.303 | 4.988 | 3.814 |
| DINO (ViT-S/16) (Caron et al., 2021) | 76.54 | 0.295 | -0.728 | 0.818 | 0.182 | 6.868 | 3.525 |
| Pearson's $\rho$ with Acc. | | 0.102 | -0.623 | -0.395 | 0.856 | 0.721 | **0.974** |
| Kendall's $\tau_K$ with Acc. | | -0.033 | 0.253 | -0.055 | 0.626 | 0.516 | **0.802** |

Table 11: Metrics evaluated on ImageNet-1k dataset using pre-trained ViT models. Because of the computational budget, we exclude the two largest models.

| Algorithm | Acc. (%) | Alignment ↓ | Uniformity ↓ | Tolerance ↑ | Linear CKA ↑ | $\hat{I}_{\text{SimCLR}}(X;Y)$ | $\hat{I}_{\text{class}}(X;Y)$ |
|---|---|---|---|---|---|---|---|
| Supervised pretrained (ViT-B/16) (Dosovitskiy et al., 2021) | 78.93 | 0.563 | -3.889 | 0.432 | 0.365 | 9.199 | 7.208 |
| DeiT (ViT-B/16) (Touvron et al., 2021a) | 78.34 | 0.842 | -3.834 | 0.482 | 0.234 | 8.679 | 8.009 |
| PiT-S (Heo et al., 2021) | 76.81 | 0.820 | -3.833 | 0.472 | 0.198 | 8.513 | 7.543 |
| CaiT (XXS-36/16) (Touvron et al., 2021b) | 75.67 | 0.637 | -3.840 | 0.550 | 0.228 | 8.373 | 6.795 |
| DeiT (ViT-S/16) (Touvron et al., 2021a) | 75.59 | 0.789 | -3.852 | 0.498 | 0.209 | 8.278 | 7.280 |
| MoCo(v3) (ViT-B/16) (Chen et al., 2021b) | 75.51 | 0.130 | -1.297 | 0.792 | 0.268 | 9.524 | 6.658 |
| CaiT (XXS-24/16) (Touvron et al., 2021b) | 74.09 | 0.661 | -3.864 | 0.516 | 0.205 | 8.315 | 6.547 |
| DINO (ViT-B/16) (Caron et al., 2021) | 73.28 | 0.411 | -3.646 | 0.465 | 0.375 | 9.367 | 6.598 |
| Supervised pretrained (ViT-S/16) (Dosovitskiy et al., 2021) | 72.85 | 0.889 | -3.506 | 0.494 | 0.428 | 7.572 | 6.233 |
| DeiT (ViT-T/16) (Touvron et al., 2021a) | 68.67 | 0.791 | -3.872 | 0.462 | 0.197 | 7.874 | 5.883 |
| Supervised pretrained (ViT-T/16) (Dosovitskiy et al., 2021) | 53.01 | 1.044 | -3.203 | 0.437 | 0.267 | 5.474 | 3.741 |
| DINO (ViT-S/16) (Caron et al., 2021) | 51.11 | 0.157 | -0.702 | 0.881 | 0.193 | 7.426 | 3.316 |
| Pearson's $\rho$ with Acc. | | 0.077 | -0.561 | -0.392 | 0.203 | 0.783 | **0.977** |
| Kendall's $\tau_K$ with Acc. | | -0.030 | -0.364 | -0.061 | 0.152 | 0.576 | **0.848** |

### C.2 Post-training MI estimation results of Figure 5

In addition to the linear evaluation performance, we also estimate MI after training and the results are shown in Figure 8. For each of the three individual tasks $C_{\text{color}}$, $C_{\text{digit}}$, and $C_{\text{position}}$, the true MI is 2 bits. For $C_{\text{all}}$, the true MI is 6 bits. For an easier understanding, we show the colors in Figure 8 after normalizing the estimated values by the true MI. As we have already found in Section 3.2, $\hat{I}_{\text{class}}(h_X; h_Y)$ is well-aligned with the linear accuracy and the post-training MI estimation values show similar patterns as in Figure 5.



ResNet-18

| | Color | Digit | Position | All |
|---|---|---|---|---|
| Color | 2.0 | 1.0 | 1.0 | 2.0 |
| Digit | 0.2 | 2.0 | 0.5 | 1.9 |
| Position | 1.6 | 2.0 | 2.0 | 2.0 |
| All | 4.5 | 4.9 | 4.1 | 5.9 |

ResNet-50

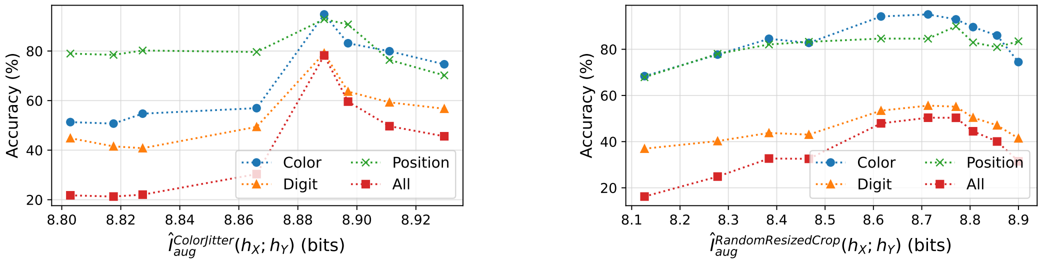
| | Color | Digit | Position | All |
|---|---|---|---|---|
| Color | 2.0 | 1.6 | 1.4 | 2.0 |
| Digit | 0.2 | 2.0 | 0.9 | 1.9 |
| Position | 1.5 | 2.0 | 2.0 | 2.0 |
| All | 4.7 | 5.4 | 4.6 | 6.0 |

Figure 8: Post-training MI estimation results for the experiment cases in Figure 5.

### C.3 Additional results of Section 3.3

We can expand our results to the case of commonly used $\mathcal{T}_{\text{aug}}$. Following Tian et al. (2020b), we utilize the color jittering and random resized crop augmentation and vary their strength parameters.

The results are provided in Figure 9. Considering that color jittering is not related to digit task nor position task, the results in Figure 9(a) indicate that the peak in the middle might not be relevant to InfoMin. Similar results can be found for random resized crop. Considering that random resized crop might be less relevant to the color task than to the digit task or position task, the results in Figure 9(b) indicate that the peak in the middle might not be relevant to InfoMin either. Based on our results, aligning the positive pairing method $\mathcal{T}$ and the downstream task $C$ is not possible, and also it might not be always helpful.



(a) Color jittering

(b) Random resized crop

Figure 9: Linear evaluation performance when (a) color jittering or (b) random resized crop is used for training.

# D Two empirical investigations related to the discussion section

## D.1 Training loss for effective contrastive learning does not need to correspond to the downstream task's MI

Because we have observed in Section 3.2 that $\hat{I}_{\text{class}}(h_X; h_Y)$ is the most effective metric for downstream task's linear evaluation performance, it is reasonable to ask if the corresponding loss $\mathcal{L}_{\text{class}}$ in Figure 2(b) can learn a better representation and achieve a better performance. Our experimental results are summarized in Table 12. We can see that the supervised loss $\mathcal{L}_{\text{class}}$ is outperformed by carefully designed unsupervised losses for two out of three cases. Even though $\hat{I}_{\text{class}}(h_X; h_Y)$ is a superior performance metric, the corresponding $\mathcal{L}_{\text{class}}$ is not necessarily a superior loss for learning representation. Furthermore, it is surprising to note that $\mathcal{L}_{\text{class}}$ is a supervised loss while the compared losses are unsupervised losses. Despite using the exact task information for the training, $\mathcal{L}_{\text{class}}$ performs worse than the carefully designed unsupervised learning methods.

A possible explanation can be related to the fact that $\mathcal{L}_{\text{class}}$ utilizes the minimum amount of information that is related to the task. While a high performing network must have its representation express the downstream-task information very well as we have shown in Section 3.2, the *training* of such a network requires additional learning signals on top of the basic downstream-task information. This explanation is also supported by the well known supervised loss $\mathcal{L}_{\text{SupCon}}$ that is proposed in (Khosla et al., 2020). Even though not shown in Table 12, the popular supervised loss $\mathcal{L}_{\text{SupCon}}$ easily outperforms the $\mathcal{L}_{\text{class}}$. In general, $\mathcal{L}_{\text{SupCon}}$ outperforms the unsupervised losses as well. $\mathcal{L}_{\text{SupCon}}$ is a supervised loss just like $\mathcal{L}_{\text{class}}$, but it experiments with known unsupervised augmentations and choose the high-performing augmentations to be used in addition to the class information.

Overall, we can conclude the followings for learning representation. (1) Using downstream-task information only (supervised) can be outperformed by a careful use of well-designed learning signals (unsupervised). (2) When supervised learning is allowed, both downstream-task information (i.e., class label) and well-designed learning signals (e.g., high-performance augmentations) should be used together to achieve the best performance.

Additionally, we would like to make it clear how our result is different from the work of Tschannen et al. (2019). It has been already pointed out by Tschannen et al. (2019) that MI alone might not be sufficient for learning effective representations for downstream tasks. The analysis method in the work, however, was not rigorous in that only a particular choice of augmentation and the corresponding joint distribution $p_{\text{aug}}(x, y)$ were studied. Without addressing exactly what information is shared by $p_{\text{aug}}(x, y)$, the analysis can be quite misleading. Furthermore, only $\mathcal{L}_{\text{aug}}$ was considered as the training objective in the work. As we have shown in Section 3.2, any analysis based on $\mathcal{L}_{\text{aug}}$ can be misleading because the information corresponding to the $p_{\text{aug}}(x, y)$ might not be sufficiently relevant to the downstream-task information anyway. In our work, we have considered $\mathcal{L}_{\text{class}}$ that is definitely related to the desired downstream-task information. While we also conclude that MI is not sufficient for a successful representation learning, our result is different and broadens the results in Tschannen et al. (2019) because we have developed and applied rigorous methods for analyzing MI in contrastive learning.

Table 12: Comparison of linear evaluation performance for a set of loss functions. Performance with $*$ indicates values reported in the existing works. Despite the superiority of $\mathcal{L}_{\text{class}}$ as a metric, generally it does not outperform the best known unsupervised losses.

| Loss | $\mathcal{L}_{\text{class}}$ | $\mathcal{L}_{\text{SimCLR}}$ | $\mathcal{L}_{\text{aug,best-known}}$ |
|---|---|---|---|
| CIFAR-10 | 91.0 | 91.0 | **94.1**$^*$ (SWD (Chen et al., 2021a)) |
| ImageNet-100 | **87.4** | 76.4 | 84.5$^*$ (MoCo-v2+MoCHi (Kalantidis et al., 2020)) |
| ImagNet-1k | 75.2 | 69.1$^*$ (Chen et al., 2020a) | **76.4**$^*$ (HCA (Xu et al., 2020)) |

## D.2 Negative sampling for effective contrastive learning does not need to follow the marginal distribution

For the contrastive learning to be equivalent to an MI maximization, the negative term (the denominator in Eq. (1)) normalized by $(2K - 1)$ needs to be an asymptotic estimation of the partition

Table 13: The effect of negative sampling dataset $\mathcal{D}^-$. Linear evaluation performance can be affected by choosing negative samples from a related or an unrelated dataset. (a) CIFAR-5A: For contrastive learning of CIFAR-5A dataset, the best performance is achieved by choosing the negative samples from CIFAR-5B dataset (i.e., not from CIFAR-5A dataset). (b) CIFAR-50A: For contrastive learning of CIFAR-50A dataset, the best performance is achieved by choosing the negative samples from CIFAR-50B dataset (i.e., not from CIFAR-50A dataset).

(a) $\mathcal{D} =$ CIFAR-5A

| $\mathcal{D}^-$ | CIFAR-5A (Baseline: InfoNCE loss) | CIFAR-5B | PACS-C | PACS-A | PACS-P | PACS-S | Uniform random |
|---|---|---|---|---|---|---|---|
| Accuracy (%) | 85.70 | **87.62** | 83.14 | 81.98 | 81.14 | 80.86 | 79.80 |

(b) $\mathcal{D} =$ CIFAR-50A

| $\mathcal{D}^-$ | CIFAR-50A (Baseline: InfoNCE loss) | CIFAR-50B | PACS-C | PACS-A | PACS-P | PACS-S | Uniform random |
|---|---|---|---|---|---|---|---|
| Accuracy (%) | 59.56 | **60.34** | 49.52 | 51.16 | 50.44 | 43.40 | 33.92 |

function $Z(y)(= \mathbb{E}_{p(y)}[e^{f(x,y)}])$ (Poole et al., 2019). This requirement can be fulfilled by drawing the negative samples with a uniform distribution over the entire training dataset. In practice, the negative samples in Eq. (1) are chosen as the samples in the mini-batch, primarily for the computational efficiency.

In contrast to the viewpoint of MI maximization, the viewpoint of *Noise Contrastive Estimation* (NCE) in (Gutmann and Hyvärinen, 2010) does not require the negative samples to be drawn from the marginal distribution. Instead, the negative samples can be drawn from any reasonable distribution including random noise such as Gaussian noise. Interestingly, both viewpoints were addressed in the original CPC work (Oord et al., 2018), but the relationship between the two viewpoints was not clarified. Here, we provide an experiment to show that the negative samples do not need to be drawn from the marginal distribution. In fact, we can enhance the performance of contrastive learning by carefully manipulating the negative sampling.

Before proceeding, we define four new datasets. CIFAR-5A and CIFAR-5B are disjoint datasets created from CIFAR-10. CIFAR-5A contains all the examples of the first five classes of CIFAR-10 and CIFAR-5B contains all the examples of the last five classes of CIFAR-10. CIFAR-50A and CIFAR-50B are created in a similar way from CIFAR-100 (first fifty classes of CIFAR-100 and last fifty classes of CIFAR-100).

The experimental results are shown in Table 13. The positive pairs are always drawn from the original dataset $\mathcal{D}$ (CIFAR-5A or CIFAR-50A), but the negative samples are drawn from the negative sampling dataset $\mathcal{D}^-$. As expected, performance degradation can be observed when $\mathcal{D}^-$ is one of PACS-(cartoon, art, photo, and sketch) (Li et al., 2017) or uniform random noise (Figure 10). When $\mathcal{D}^-$ is CIFAR-5B, however, the performance is improved by 1.92%. The same observations can be made for CIFAR-50A, with the improvement of 0.77%. The experiment results indicate that we can improve the linear evaluation performance by carefully choosing $\mathcal{D}^-$ for negative sampling. In our experiments, the performance was enhanced by choosing a dataset whose distribution slightly diverges from the true marginal distribution (CIFAR-5B and CIFAR-50B are not the marginals but at least they come from the same source datasets of CIFAR-10 and CIFAR-100).

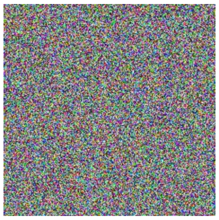


Figure 10: An example of uniform random noise image.