

# Boosting Contrastive Self-Supervised Learning with False Negative Cancellation

Tri Huynh\*  
University of Chicago

Simon Kornblith  
Google Research

Matthew R. Walter  
TTI-Chicago

Michael Maire  
University of Chicago

Maryam Khademi  
Google

## Abstract

Self-supervised representation learning has witnessed significant leaps fueled by recent progress in Contrastive learning, which seeks to learn transformations that embed positive input pairs nearby, while pushing negative pairs far apart. While positive pairs can be generated reliably (e.g., as different views of the same image), it is difficult to accurately establish negative pairs, defined as samples from different images regardless of their semantic content or visual features. A fundamental problem in contrastive learning is mitigating the effects of false negatives. Contrasting false negatives induces two critical issues in representation learning: discarding semantic information and slow convergence. In this paper, we study this problem in detail and propose novel approaches to mitigate the effects of false negatives. The proposed methods exhibit consistent and significant improvements over existing contrastive learning-based models. We achieve new state-of-the-art performance on ImageNet evaluations, achieving 5.8% absolute improvement in top-1 accuracy over the previous state-of-the-art when finetuning with 1% labels, as well as transferring to downstream tasks.

## 1. Introduction

With the universal adaptability of deep neural networks, representation learning has become the backbone of most modern AI agents, in which good pretrained representations have proven essential to improving performance on downstream tasks [16, 20, 49, 27]. While conventional approaches use labeled data to pretrain visual representations, there has been a recent surge in self-supervised representation learning [19, 15, 36, 45, 38, 8, 50, 30]. In fact, self-supervised visual representation learning has been closing the gap with, and in some cases even surpassing its supervised counterpart [9, 22, 11, 10]. Notably, most state-of-the-art self-supervised visual representation learning methods are converging around, and fueled by, the central concept of

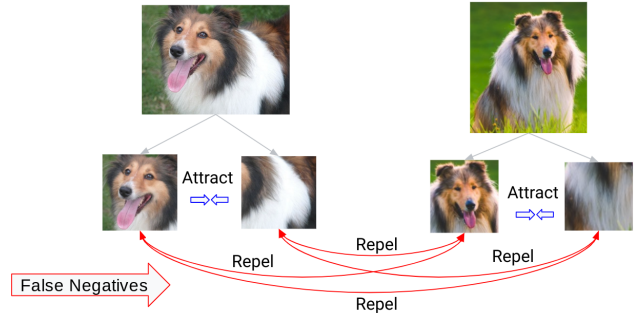


Figure 1. False negatives in contrastive learning. Without knowledge of labels, automatically selected negative pairs could actually belong to the same semantic category, creating false negatives.

contrastive learning [44, 23, 24, 42, 34, 22, 9].

In contrastive learning, the embedding space is governed by two opposing forces, the attraction of positive pairs and repulsion of negative pairs, effectively actualized through the contrastive loss. Without labels, recent breakthroughs in self-supervised visual representation learning rely on the instance discrimination task in which positive pairs are defined as different views of the same image, while negative pairs are formed by sampling views from different images, regardless of their semantic information [22, 9, 34]. Figure 1 illustrates this process. Positive pairs generated from different views of the same image are generally accurate since they are likely to carry similar semantic content or visual features. However, the creation of valid negative pairs is far more difficult. The common approach of defining negative pairs as samples from different images ignores their semantic content. For example, two images of a dog are considered a negative pair, as demonstrated in Figure 1.

By contrasting these undesirable negative pairs, the architecture is encouraged to discard their common features in the learned embedding, which are indeed the common semantic content, e.g., dog features in the previous example. We define those undesirable negative samples as false negatives, i.e., negative pairs from the same semantic category. Besides disregarding the semantic information, false negatives also hinder the convergence of contrastive learning-based objectives due to the appearance of contradicting objectives. For instance, as shown in Figure 1, the dog’s head

\*Work done during an internship at Google. Correspondence to Tri Huynh <trihuynh@uchicago.edu>, Maryam Khademi <maryamkhademi@google.com>.

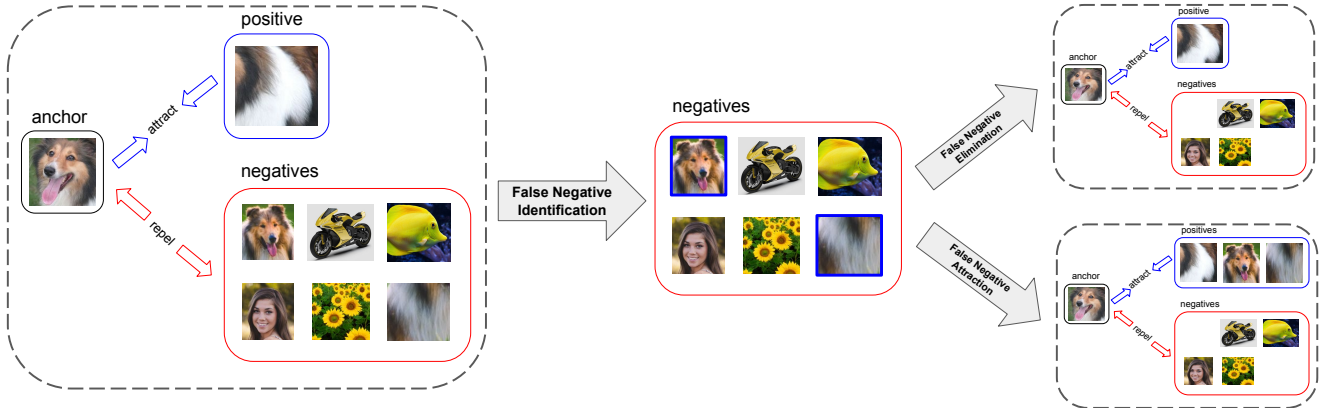


Figure 2. Overview of the proposed framework. **Left:** Original definition of the anchor, positive, and negative samples in contrastive learning. **Middle:** Identification of false negatives (blue). **Right:** false negative cancellation strategies, *i.e.* elimination and attraction.

on the left is attracted to its fur (positive pair), but is repelled from similar fur of another dog image on the right (negative pair), creating contradicting objectives.

While recent efforts focus on improved architectures [9, 10, 22] and data augmentation [9, 43], relatively little work considers the effects of negative samples, especially that of false negatives. Most existing methods focus on mining hard negatives [39, 25], or most recently, increasing positive samples to counter-balance the negatives [12]. However, there has been little effort to identify false negatives.

False negatives remain a fundamental problem in contrastive self-supervised learning. Without labels, this problem is very hard to adequately resolve, as it boils down to a chicken-and-egg problem, where we want to learn good semantic representations, but may need certain semantic information to start with. Nevertheless, in this paper, we attempt to study this problem in detail and propose novel ideas to overcome its limitations, as overviewed in Figure 2. Particularly, the contributions of the paper are as follows:

- We propose simple yet effective strategies to find potential false negatives in contrastive learning. Without any labels, the proposed method effectively finds false negatives with  $\sim 40\%$  accuracy among 1000 human-defined semantic categories on ImageNet [40]. Our work is the first to address this problem.
- We propose and study the effect of two different strategies to improve the contrastive loss based on the estimated false negatives, namely, false negative elimination and attraction.
- We show that the proposed methods consistently and significantly improve over existing contrastive learning-based approaches across a wide range of settings, *e.g.*, with or without momentum contrast [22]. Further, we show that our methods are effectively complementary with the recent multi-crop augmentation strategy in terms of both accuracy and computational

efficiency: Adding false negative cancellation on top of multi-crop incurs negligible computational overhead, while doubling the improvement (4% in ImageNet linear evaluation).

- The improved model establishes new state-of-the-art results for contrastive learning-based methods across all evaluations on ImageNet, as well as transferring to downstream tasks. Notably, the model achieves 5.8% absolute improvement of Top-1 accuracy in the semi-supervised setting with 1% labels on ImageNet.

## 2. Related Work

Early work on self-supervised learning employs proxy tasks to guide the learned embeddings, such as predicting the angle of a rotated image [19], the relative location of patches [15], or organizing shuffled patches to recover the original image much like solving a jigsaw puzzle [36]. Other proxy tasks include recovering an image from a corrupted version [45], predicting part of the image from context [38, 8] or generating one view of an image from another, *e.g.*, split-brain auto-encoder [50] or colorization [31, 51, 30]. While these approaches have been effective, the proxy tasks are fairly heuristic and lack generality.

Other works use clustering-based methods for self-supervised representation learning [1, 3, 47]. Caron et al. [6] iteratively improve the learned representations by clustering samples (*e.g.*, k-means) and using these clusters as pseudo-labels. They then train the network to classify samples based on these pseudo-labels. Our attraction-based approach to false negative cancellation is similar to clustering methods, as they both attempt to group visually connected samples. However, the formulation and context differs.

We take a contrastive learning-based approach to self-supervised representation learning. Earlier work in this area includes CPC [44, 23], Deep InfoMax [24, 2], and contrastive multiview coding [42]. Recognizing that contrastive

loss requires a large set of negative samples, PIRL [34] maintains a memory bank of previous representation of all images, which limits scalability. MoCo v1 [22] addresses this problem by maintaining a momentum encoder and a limited queue of previous samples. SimCLR v1 [9] eschews a momentum encoder in favor of a large batch size, and proposes updates to the projection head and data augmentation.

Realizing the important role of negative samples in contrastive learning, a few recent methods have investigated ways to improve negative sampling. These include increasing the sampling of hard negatives (*i.e.*, those that are close to the anchor) [39, 25], and using more positive samples to counter-balance the effects of undesirable negatives [12].

Inspired by contrastive learning, recent methods seek to predict one positive view from another. However, they are not categorized as contrastive learning as they have different formulations at their core and do not contrast against negative samples, a defining element of contrastive learning. SwAV [7] is an online clustering-based method that employs swapping prediction of different views from an image, while BYOL [21] employs a momentum encoder as a prediction target for another view from the main encoder.

### 3. Method

#### 3.1. Contrastive Learning

The goal of contrastive learning is to learn a transformation that brings positive pairs “nearby” in an embedding space while pushing negative pairs apart. This is done by minimizing a contrastive loss that, for each *anchor* image  $i$ , measures the (negative) similarity between its embedding  $z_i$  and that of its positive match  $z_j$  relative to the similarity between the anchor embedding of  $k \in \{1, \dots, M\}$  negative matches:

$$l_i = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^M \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}, \quad (1)$$

where  $\text{sim}(u, v)$  is a similarity function, *e.g.*, the  $L_2$  normalized cosine similarity  $\text{sim}(u, v) = u^T v / \|u\| \|v\|$ , and  $\tau$  is a temperature parameter.

Without known correspondence, self-supervised methods commonly define positive pairs as different augmentations of the same image and negative pairs as samples from different images. Consider a batch of  $N$  images, each augmented to form  $N$  positive pairs for a total of  $2N$  images. Methods like SimCLR use samples from the same batch as negative pairs (*i.e.*,  $M = 2N$  examples for each anchor  $i$ ), while MoCo uses samples from a momentum encoder to avoid the use of a large batch size.

#### 3.2. False Negative Cancellation

Regardless of whether the negative samples are drawn from the same batch or the output of a momentum encoder,

they are samples from different images that may or may not possess similar semantic content or visual features. Consequently, it is possible that some samples  $k$  have the same semantic content as the anchor  $i$ , and are thus false negatives. As discussed earlier, false negatives give rise to two critical problems in contrastive learning: they discard semantic information and slow convergence.

Our method seeks to mitigate these problems. In particular, we propose a strategy to identify false negatives and an approach that uses these false negatives to improve contrastive learning. The following discussion describes how we might incorporate knowledge of false negatives in contrastive learning. We discuss the process of identifying false negatives in Section 3.2.3.

Supposing we can find the false negatives, we propose two strategies that use them to improve contrastive learning.

##### 3.2.1 False Negative Elimination

The simplest strategy for mitigating the effects of false negatives is to not contrast against them. This amounts to the following modification to the contrastive objective (1):

$$l_i^{\text{elim}} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i, k \notin \mathbb{F}_i]} \exp(\text{sim}(z_i, z_k)/\tau)}, \quad (2)$$

where  $\mathbb{F}_i$  is the set of the detected false negatives with respect to an anchor  $i$ .

##### 3.2.2 False Negative Attraction

While eliminating false negatives alleviates the undesirable effects of contrasting against them, it ignores information available in what are actually true positives. Minimizing the original contrastive loss (Eqn. 1) only seeks to attract an anchor to different views of the same image. Including true positives drawn from different images would increase the diversity of the training data and, in turn, has the potential to improve the quality of the learned embeddings. Indeed, Khosla et al. [26] show that supervised contrastive learning (*i.e.*, where an anchor is attracted to samples having the same semantic label) can be more effective than the traditional supervised cross-entropy loss. Thus, we propose to treat the false negatives that have been identified as true positives and attract the anchor to this set. This yields the following expression for the contrastive loss:

$$l_i^{\text{att}} = -\frac{1}{1 + |\mathbb{F}_i|} \left( \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} + \sum_{f \in \mathbb{F}_i} \log \frac{\exp(\text{sim}(z_i, z_f)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \right) \quad (3)$$

We note that compared to simply ignoring the detected false negatives (2), the attraction strategy is more sensitive

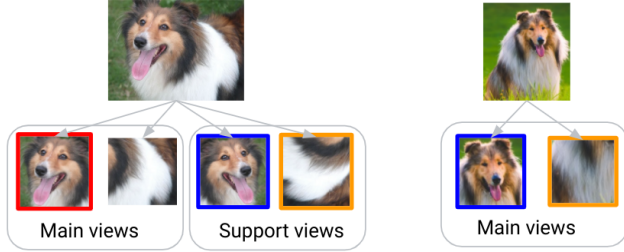


Figure 3. Finding false negatives with the support views. Negative samples (main views, right) may not have as reliable similarity with the anchor itself (red) as they do with other augmented views of the same image (support views). For instance, the dog’s face in the support view (left, blue) is closer to the negative sample (right, blue) in terms of the facial orientation than the anchor (red).

to the quality of the candidate false negatives. Encouraging embeddings that erroneously attract false positives can degrade the quality of the learned representations.

### 3.2.3 Finding False Negatives

Unfortunately, the process of identifying false negatives is fundamentally difficult, amounting to a chicken-and-egg problem—without labels, the learned semantic information could be used to establish valid and invalid correspondences, yet the correctness of these embeddings depends on the ability to identify correspondences. We propose a new approach to identify false negatives based on the following observations:

- *False negatives are samples from different images with the same semantic content, therefore they should hold certain similarity (e.g., dog features).*
- *A false negative may not be as similar to the anchor as it is to other augmentations of the same image, as each augmentation only holds a specific view of the object.*

The above observations mean that we may be able to approximate a false negative with more augmented views of the anchor. As an example, consider Figure 3 where we treat the picture of the dog’s head on the left (“main views,” in red) as the anchor image. The support views on the left are other augmented views generated from the same image and serve as positive matches. The picture of the dog’s head on the far right (“main view,” in blue) is not an augmented version of the anchor. Consequently, while it is similar to the anchor image, it would thus be treated as a negative match by contemporary self-supervised methods (*i.e.*, it is a false negative). However, we see that this image is more similar to the augmented view of the anchor (“support view,” in blue) than it is to the anchor with respect to the orientation of the dog’s face. Similarly, the head and fur of the dog are expected to be a positive pair. While the false negative fur on the far right (orange) could look different than the head

anchor, it should be more similar to the fur in the support views (orange).

Motivated by these observations, we propose a strategy for identifying candidate false negatives that follows as:

1. For each anchor  $i$ , generate a support set  $\mathbb{S}_i = \{z_i^s\}$  that contains other support views from the same image besides the two main views.
2. Compute similarity scores,  $\text{score}_{m,i}^s = \text{sim}(z_m, z_i^s)$ , between a negative sample  $z_m$  and each sample  $z_i^s$  in the support set.
3. Aggregate the computed scores for each negative sample,  $\text{score}_{m,i} = \text{aggregate}_{s \in \mathbb{S}}(\text{score}_{m,i}^s)$ .
4. Define a set of potential false negatives  $\mathbb{F}_i$  as the negative samples that are most similar to the support set based on the aggregated scores,  $\mathbb{F}_i = \text{best}(\text{score}_i)$ , where  $\text{score}_i = \{\text{score}_{m,i} | m\}$  is the set of scores for each negative sample with respect to anchor  $i$ .

For each element in the above procedure, there are several considerations one can make, including the choice of the similarity function, the strategy for aggregating scores, and the manner in which the most similar samples are defined. In this work, we investigate the following options:

**Similarity Function** We use the cosine similarity function, since it is used in the contrastive loss during pretraining.

**Aggregation Strategy** We consider both mean aggregation,  $\text{score}_{m,i} = \frac{1}{|\mathbb{S}|} \sum_{s=1}^{|\mathbb{S}|} \text{sim}(z_m, z_i^s)$ , and max aggregation,  $\text{score}_{m,i} = \max_{s \in \{1, \dots, |\mathbb{S}|\}} \text{sim}(z_m, z_i^s)$ , and discuss their effects in Section 4.

**Screening Strategy** We consider two choices for the most similar samples, one that considers the top- $k$  matches,  $\text{best}(\text{score}_i) = \{z_m | \text{score}_{m,i} \in \text{top}(\text{score}_i, k)\}$ , and one that considers those above a threshold  $t$ ,  $\text{best}(\text{score}_i) = \{z_m | \text{score}_{m,i} > t\}$ . A top- $k$  strategy may be preferred given information about the approximate number of false negatives, while thresholding may be better suited when a dynamic adaptation is expected. We also consider a strategy that combines top- $k$  and and thresholding,  $\text{best}(\text{score}_i) = \{z_m | \text{score}_{m,i} \in \text{top}(\text{score}_i, k) \ \& \ \text{score}_{m,i} > t\}$ .

## 4. Experimental Results

### 4.1. Ablation Studies

We use the same configuration as SimCLR v2 for pre-training and evaluation. The base encoder is ResNet-50 with a 3-layer MLP projection head. Data augmentation includes random crops, color distortion, and Gaussian blur. For each experiment, we pretrain for 100 epochs on the ImageNet ILSVRC-2012 training set, then freeze the encoder



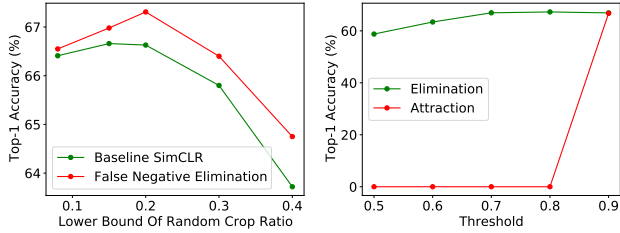


Figure 4. A comparison of top-1 accuracy (left) between false negative elimination and SimCLR; and (right) top-1 accuracy across filtering thresholds in false negative cancellation.

and train a linear classifier on top, which is then evaluated on the ImageNet evaluation set. We pretrain on 128 Cloud TPUs with a batch size of 4096. We use the LARS optimizer with a learning rate of 6.4 and a cosine decay schedule, and a weight decay of  $1 \times 10^{-4}$ .

#### 4.1.1 False Negative Cancellation Strategies

We evaluate the effects of various approaches to false negative mitigation, including the choice of cancellation strategy, aggregation score, and screening strategies, and draw the following conclusions.

**False negative elimination consistently improves contrastive learning across crop sizes, and the gap is higher for bigger crops.** Figure 4 (left) demonstrates that the inclusion of false negative elimination yields top-1 accuracy that is strictly better than that of the SimCLR baseline across the full range of crop ratios. We postulate that the bigger gap for larger crop sizes is due to the increased chance of having common semantic content in big crops, which leads to a higher ratio of false negatives. It is also worth noting that in Figure 4 (left), we only eliminate a negligible number of two potential false negatives among 8190 (batch size 4096) negative samples for each anchor, but it could affect top-1 accuracy by as much as 1%. This supports the significant effect of false negatives in contrastive learning.

**Having a support set helps in finding false negatives regardless of the cancellation strategy, with greater benefits with the attraction strategy.** Figure 5 contrasts the

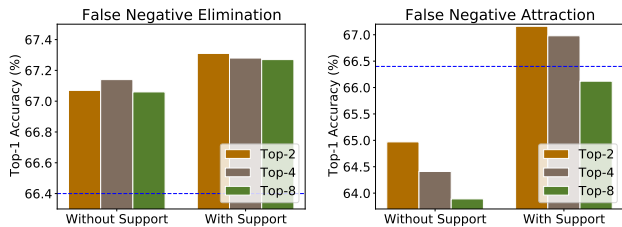


Figure 5. False negative cancellation with and without support set across top- $k$  choices for different mitigation strategies. The dashed line denotes the performance of the SimCLR baseline. The results use mean aggregation in scoring potential false negatives.

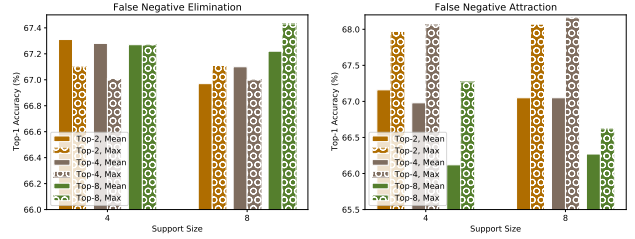


Figure 6. False negative cancellation with mean and max aggregation across support sizes and top- $k$  for the false negative (left) elimination and (right) attraction strategies.

top-1 accuracy when we compare negative samples to a support set (Section 3.2.3) to the case in which we only compare negative samples to the anchor itself. The use of a support set results in larger performance gains when using false negative attraction ( $\sim 2\%$ ) compared to the false negative elimination strategy ( $\sim 0.2\%$ ). Further, while the elimination strategy improves performance relative to the SimCLR baseline whether or not a support set is used, attracting false negatives found without support set actually hurts performance (Figure 5, right). This likely results from the fact that embeddings learned with attraction strategy are more sensitive to invalid false negatives (discussed next), justifying the use of a support set to reliably find false negatives.

**The attraction strategy is much more sensitive to the quality of the found false negatives compared to the elimination strategy.** This property is consistently confirmed through Figure 5 (right) as previously discussed, and Figure 4 (right), where the attraction strategy only works with more reliable false negatives, those having very high similarity scores, while the elimination method is not very sensitive to the thresholds.

**Max aggregation significantly and consistently outperforms mean aggregation for the attraction strategy, while the gains are less pronounced with false negative elimination.** Figure 6 demonstrates that max aggregation outperforms mean aggregation for all support sizes and top- $k$  values in the attraction strategy, with a gap in some cases greater than 1%. This may be due to the fact that false negatives are similar to a strict subset of the support set, in which case considering all elements as in mean aggregation corrupts the similarity score. The difference is more pronounced for the attraction strategy, which is more sensitive to invalid false negatives.

**Filtering by top- $k$  tends to perform better than by a threshold, while a combination of both provides the best balance.** As seen in Figure 7, the best choice of top- $k$  is better than the best threshold. A strategy that combines the two approaches achieves greater accuracy, with the exception of false negative attraction at top-4, for which there is a negligible degradation in performance.

**False negative attraction is superior to elimination**

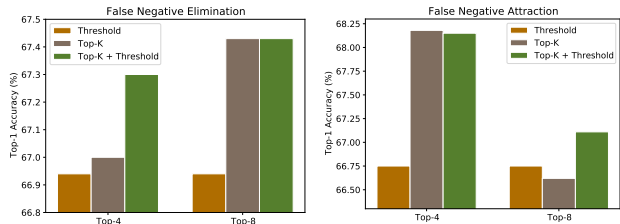


Figure 7. A comparison of top- $k$  and threshold-based filtering for false negative (left) elimination and (right) attraction strategies.

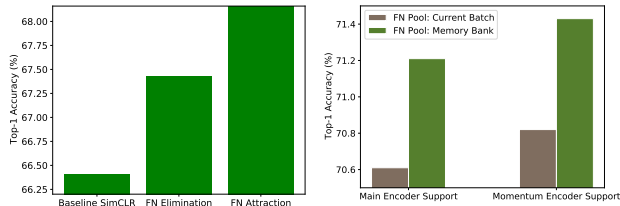


Figure 8. A comparison of top-1 accuracy (left) for different false negative cancellation strategies relative to the SimCLR baseline and (right) the use of the primary and momentum encoders.

SimCLR	FN Elimination / Gain	FN Attraction / Gain
66.41	67.43 / +1.02	68.16 / +1.75

Table 1. Top-1 accuracy improvement of false negative cancellation strategies over the SimCLR baseline.

**when the detected false negatives are valid.** As shown in Figure 8 (left) and Table 1, false negative elimination improves upon SimCLR by 1.02%, while false negative attraction results in a 1.75% improvement. These results use max aggregation, a support size of eight, top-4 for false negative attraction, and top-8 filtering for elimination.

#### 4.1.2 With Multi-crop

Caron et al. [7] propose a multi-crop data augmentation strategy that increases the number of positive views attracted to each anchor image, improving the quality of the learned embeddings. Multi-crop is closely related and complementary to false negative attraction in multiple facets, from principle to computational efficiency. While multi-crop attracts more positive samples, false negative attraction tries to attract samples that would otherwise be erroneously treated as negatives. The positive samples should be more reliable than the false negatives we attempt to find; however, multi-crop lacks semantic feature diversity, as it never attracts samples from different images. Because of these characteristics, multi-crop and false negative attraction offer complementary advantages. Furthermore, they can share computational overhead by using a common support set. Thus far, we have only used the support set to find false negatives, but they can also be used as additional positive views for multi-crop. In doing so, we may be able to

Method	Accuracy (Diff.)	Time (Diff.)
SimCLR	66.41	2.63
SimCLR + Multi-crop	68.50 (+2.09)	7.40 (+4.77)
SimCLR + Multi-crop + FN Att.	70.42 (+1.92)	7.50 (+0.10)

Table 2. Complementary performance and computational efficiency of multi-crop and false negative attraction.

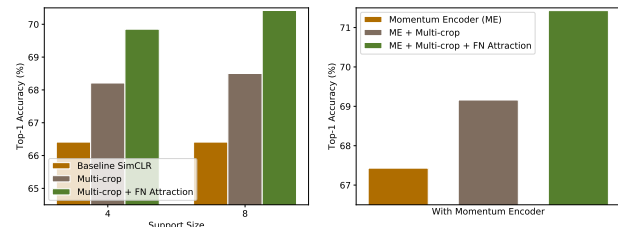


Figure 9. False negative cancellation with (left) multi-crop and (right) momentum encoder.

double the performance without noticeable overhead if the respective improvements are complementary.

Figure 9 and Table 2 demonstrate the advantages of using false negative cancellation together with multi-crop data augmentation. Adding multi-crop improves the performance of the SimCLR baseline by 2.09%, while incurring an additional 4.77 hours of computational overhead. Further adding false negative attraction on top of multi-crop yields a 1.92% absolute improvement in accuracy (i.e., similar to the 2.09% gain provided by multi-crop), while incurring only 0.1 hours of computation time.

#### 4.1.3 With Momentum Encoders

Thus far, we have identified false negatives in the current batch from a single encoder. However, other methods may store negative samples in a memory bank, or encoded by a momentum encoder. Here, we investigate the behaviors of the proposed method in these settings. Momentum contrast [22] employs two encoders, the main encoder and momentum encoder, and a memory bank where negative samples are stored alongside the samples from the current batch. This offers more options for finding false negatives, i.e., whether to use the support set from the main encoder or the momentum encoder, or whether to find negatives from samples in the current batch or all samples in memory. Figure 8 (right) indicates that it is better to generate the support set using samples from the momentum encoder compared to the main encoder. Further, finding false negatives in the memory bank yields greater top-1 accuracy than drawing false negatives from the current batch.

Figure 9 and Table 3 show that our method works across different configurations, with or without the presence of either momentum contrast or multi-crop. Notably, the performance gain from false negative cancellation in the presence

	Baseline	+ FN Cancel	Diff.
SimCLR	66.41	68.16	+1.75
SimCLR + Multi-crop	68.50	70.42	+1.92
SimCLR + Multi-crop + Momentum	69.16	71.43	+2.27

Table 3. Top-1 accuracy improvement of false negative cancellation for different baselines.

Model	Epochs	Time (h)	Acc. (%)
SimCLR	100	2.63	66.41
SimCLR	1000	26.22	70.34
Improved Model	100	7.50	70.42

Table 4. Computational efficiency and accuracy.

of momentum contrast is even higher at 2.27%.

#### 4.1.4 With True Labels & False Negatives Accuracy

We now consider the effectiveness of our false negative detection and cancellation strategy relative to the ideal setting in which we have access to ground-truth labels, which provides an upper-bound on performance. Figure 10 presents the top-1 accuracy when using false negative cancellation combined with multi-crop data augmentation as well as the accuracy that results from false negative cancellation using ground-truth labels. As expected, cancelling false negatives helps substantially (8.18%) when true labels are used. However, we close half the gap by just using multi-crop and the false negatives our method finds, increasing top-1 accuracy by 4.01% over the SimCLR baseline.

To better understand the extent to which we are able to identify false negatives, Figure 10 (right) plots the accuracy of the false negative detections over 100 epochs of pretraining. We see that the false negative detection accuracy steadily increases, reaching approximately 40% accuracy by 100 epochs. Note that the false negatives accuracy is computed based on human-defined semantic labels, with 1000 categories in ImageNet. The chance of finding a false negative for an anchor at random is just 0.1%.

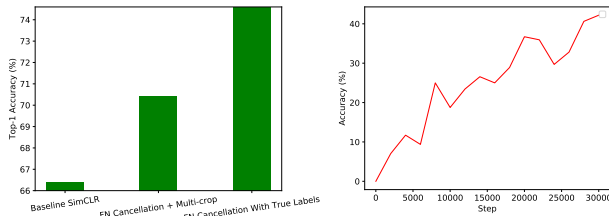


Figure 10. A visualization of (left) top-1 accuracy with false negative cancellation using detected vs. ground-truth labels and (right) the accuracy of false negative detection.

#### 4.1.5 Computational Efficiency

Table 4 compares the computational cost of our false negative cancellation strategy compared to the SimCLR baseline. As expected, we can see that for the same number of epochs, the process of detecting and incorporating false negatives incurs additional computational time. However, to achieve the same accuracy, SimCLR requires more than three times the amount of computation (26.22 h and 1000 epochs) than our framework (7.50 h and 100 epochs).

#### 4.2. Comparison with State-of-the-Art

We compare our improved model with false negative cancellation to other state-of-the-art methods on standard ImageNet evaluations and transfer to downstream tasks.

**Pretraining Settings** We use similar configurations as SimCLR v2. Specifically, we use ResNet-50 as the base encoder, with a 3-layer MLP projection head. We use a 65k memory buffer for the momentum encoder, with a momentum of 0.999. Following Tian et al. [43], we use random crops, color distortion, Gaussian blur, and RandAugment [14] for data augmentation. We also adopt multi-

Method	top-1	top-5
Supervised	76.5	
<i>Representation Learning</i>		
<i>Contrastive learning</i>		
MoCo v1 [22]	60.6	—
PIRL [34]	63.6	—
PCL [32]	65.9	—
SimCLR v1 [9]	69.3	89.0
MoCo v2 [11]	71.1	—
SimCLR v2 [10]	71.7	90.4
InfoMin [43]	73.0	91.1
FNC (ours)	<b>74.4</b>	<b>91.8</b>
<i>Others</i>		
BYOL [21]	74.3	91.6
SwAV [7]	<b>75.3</b>	—

Table 5. ImageNet linear evaluation.

Method	1% top-1	10% top-5	10% top-1	10% top-5
Supervised	25.4	56.4	48.4	80.4
<i>Semi-supervised</i>				
UDA [48]	—	68.8	—	88.5
FixMatch [41]	—	71.5	—	89.1
<i>Representation Learning</i>				
<i>Contrastive learning</i>				
PIRL [34]	30.7	60.4	57.2	83.8
PCL [32]	—	—	75.6	86.2
SimCLR v1 [9]	48.3	75.5	65.6	87.8
SimCLR v2 [10]	57.9	82.5	68.4	89.2
FNC (ours)	<b>63.7</b>	<b>85.3</b>	<b>71.1</b>	<b>90.2</b>
<i>Others</i>				
BYOL [21]	53.2	78.4	68.8	89.0
SwAV [7]	53.9	78.5	70.2	89.9

Table 6. ImageNet semi-supervised evaluation.

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
<i>Linear evaluation</i>												
SimCLR v1 [9]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
SimCLR v2 [10]	73.9	92.4	76.0	44.7	61.0	54.9	51.1	81.2	<b>76.5</b>	85.0	91.2	93.5
BYOL [21]	<b>75.3</b>	91.3	<b>78.4</b>	<b>57.2</b>	62.2	67.8	60.6	82.5	75.5	<b>90.4</b>	<b>94.2</b>	<b>96.1</b>
FNC (ours)	74.4	<b>93.0</b>	76.8	54.0	<b>63.2</b>	<b>68.8</b>	<b>61.3</b>	<b>83.0</b>	76.3	89.0	93.5	94.9
<i>Finetuned</i>												
SimCLR v1 [9]	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
SimCLR v2 [10]	88.2	97.5	86.0	74.9	<b>64.6</b>	91.8	87.6	84.1	74.7	89.9	92.3	97.2
BYOL [21]	<b>88.5</b>	<b>97.8</b>	86.1	<b>76.3</b>	63.7	91.6	88.1	<b>85.4</b>	<b>76.2</b>	<b>91.7</b>	<b>93.8</b>	97.0
FNC (ours)	88.3	97.7	<b>86.8</b>	<b>76.3</b>	64.2	<b>92.0</b>	<b>88.5</b>	84.7	76.0	90.9	93.6	<b>97.5</b>

Table 7. Transfer learning on classification task using ImageNet-pretrained ResNet models across 12 data sets.

Method	AP50
Supervised	81.3
MoCo v2 [11]	82.5
SwAV [7]	82.6
FNC (ours)	<b>82.8</b>

Table 8. Transfer learning on Pascal VOC object detection.

crop [7] with a support size of eight, which is shared with false negative cancellation. We use the attraction strategy with max aggregation for false negative cancellation, while the top- $k$  is set to 10 and a threshold of 0.7 is used for filtering the scores. We pretrain for 1000 epochs on 128 Cloud TPUs with a batch size of 4096. We use the LARS optimizer with a learning rate of 6.4, a cosine schedule for decaying the learning rate, and a weight decay of  $1 \times 10^{-4}$ .

#### 4.2.1 ImageNet Evaluation

Our evaluation on ImageNet follows a protocol similar to that of SimCLR v2. Specifically, we conduct linear evaluation, in which a linear classifier is trained on top of the frozen pretrained features, and consider semi-supervised settings, in which we finetune the network with 1% and 10% labels available. We use a batch size of 1024 with a 0.16 learning rate, while weight decay is removed. We finetune for 90 epochs in linear evaluation, 60 epochs for 1%, and 30 epochs for 10% labels in semi-supervised settings.

Table 5 presents the linear evaluation results. Our method achieves state-of-the-art performance for contrastive learning-based models with a top-1 accuracy of 74.4%, a 2.7% improvement over SimCLR v2 and a 1.4% boost from the previous best. Among all approaches, our method is second only to SwAV [7], a clustering-based method, and is even better than BYOL [21], a recent state-of-the-art method that only uses positive samples.

Table 6 presents the results in the semi-supervised setting, in which our proposed model not only exceeds other contrastive learning-based methods, but also achieves the best performance among all models across all measures. Notably, in the 1% labels setting, our method significantly

improves over the previous best to achieve 63.7% top-1 accuracy (a 5.8% absolute improvement).

#### 4.2.2 Transferring Features

**Image Classification** Following SimCLR, we perform the same evaluations on 12 classification datasets: Food [5], CIFAR10 [29], CIFAR100 [29], Birdsnap [4], SUN397 [46], Cars [28], Aircraft [33], VOC2007 [17], DTD [13], Pets [37], Caltech-101 [18], and Flowers [35]. We follow the same setup as in SimCLR. As Table 7 demonstrates, our approach achieves a significant improvement in performance among contrastive learning-based methods (i.e., SimCLR v1 and SimCLR v2) in both settings. In linear evaluation, our method outperforms SimCLR v1 on all 12 datasets and is better than SimCLR v2 on all but one dataset. In finetuning, the proposed model outperforms both SimCLR v1 and v2 on all but one dataset, and matches that of BYOL, with each being superior on about half of the datasets.

**Object Detection** To further evaluate the transferability of the learned embeddings, we finetune the model on PASCAL VOC object detection. We use similar settings as in MoCo [22], where we finetune on the VOC `trainval107+12` set using Faster R-CNN with a R50-C4 backbone, and evaluate on VOC `test2007`. We train for 34K iterations with batch size 16. The learning rate is set to 0.02, which is reduced by a factor of 10 after 20K and 28K iterations. As Table 8 reveals, our proposed method outperforms both MoCo v2, SwAV, and the supervised baseline.

## 5. Conclusion

In this work, we address a fundamental problem in contrastive self-supervised learning that has not been adequately studied, identifying false negatives, and propose strategies to utilize this ability to improve contrastive learning frameworks. In addition to bringing novel insights to this topic through in-depth experimental analysis, our proposed method significantly boosts existing models, and sets new performance standards for contrastive self-supervised learning methods.



## 6. Acknowledgement

We would like to express special thanks to Ting Chen and Mohammad Norouzi for helpful discussions regarding SimCLR and feedbacks on the project, Benjamin Louie for his dedicated helps during the internship, and Google Geo for the infrastructure and general support towards the project.

## References

- [1] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” in *Proc. Int’l Conf. on Learning Representations (ICLR)*, 2020. 2
- [2] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 15 535–15 545. 2
- [3] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer, “Cliquecnn: Deep unsupervised exemplar learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 3846–3854. 2
- [4] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, “Large-scale fine-grained visual categorization of birds,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 8
- [5] L. Bossard, M. Guillaumin, and L. V. Gool, “Food-101—Mining discriminative components with random forests,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2014. 8
- [6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2018. 2
- [7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 6, 7, 8
- [8] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2020. 1, 2
- [9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2020. 1, 2, 3, 7, 8
- [10] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 7, 8
- [11] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020. 1, 7, 8
- [12] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, “Debiased contrastive learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3
- [13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, “Describing textures in the wild,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 8
- [14] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical data augmentation with no separate search,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7
- [15] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proc. Int’l. Conf. on Computer Vision (ICCV)*, 2015. 1, 2
- [16] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “DeCAF: A deep convolutional activation feature for generic visual recognition,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2014, pp. 647–655. 1
- [17] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” *Int’l J. on Computer Vision*, vol. 88, no. 2, pp. 303—338, 2010. 8
- [18] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” in *Proceedings of the CVPR Workshop on Generative-Model Based Vision*, 2004. 8
- [19] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *Proc. Int’l Conf. on Learning Representations (ICLR)*, 2018. 1, 2
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587. 1

- [21] J.-B. Grill, F. Strub, F. Althché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020. 3, 7, 8
- [22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 6, 7, 8
- [23] O. J. Henaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, “Data-efficient image recognition with contrastive predictive coding,” in *Proc. Int’l Conf. on Learning Representations (ICLR)*, 2020. 1, 2
- [24] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *Proc. Int’l Conf. on Learning Representations (ICLR)*, 2019. 1, 2
- [25] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, “Hard negative mixing for contrastive learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3
- [26] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *arXiv preprint arXiv:2004.11362*, 2020. 3
- [27] D.-K. Kim and M. R. Walter, “Satellite image-based localization via learned embeddings,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, Singapore, May 2017. 1
- [28] J. Krause, M. S. J. Dengand, and L. Fei-Fei, “3D object representations for fine-grained categorization,” in *Proc. IEEE Int’l Conf. on Computer Vision Workshops*, 2013. 8
- [29] A. Krizhevsky, “Learning multiple layers of features from tiny images,” *Technical report, University of Toronto.*, 2009. 8
- [30] G. Larsson, M. Maire, and G. Shakhnarovich, “Colorization as a proxy task for visual understanding,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 840–849. 1, 2
- [31] G. Larsson, M. Maire, and G. Shakhnarovich, “Learning representations for automatic colorization,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2016. 2
- [32] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. H. Hoi, “Prototypical contrastive learning of unsupervised representations,” *arXiv preprint arXiv:2005.04966*, 2020. 7
- [33] S. Maji, E. Rahtu, J. Kannala, M. B. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” *arXiv preprint arXiv:1306.5151*, 2013. 8
- [34] I. Misra and L. van der Maaten, “Self-supervised learning of pretext-invariant representations,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3, 7
- [35] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *Proc. of the Indian Conf. on Computer Vision, Graphics and Image Processing*, 2008. 8
- [36] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2016. 1, 2
- [37] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, “Cats and dogs,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 8
- [38] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2
- [39] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, “Contrastive learning with hard negative samples,” *arXiv preprint arXiv:2010.04592*, 2020. 2, 3
- [40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115(3):211–252, 2015. 2
- [41] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *arXiv preprint arXiv:2001.07685*, 2020. 7
- [42] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2020. 1, 2

- [43] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good views for contrastive learning,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2020. [2](#), [7](#)
- [44] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2019. [1](#), [2](#)
- [45] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2008. [1](#), [2](#)
- [46] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “SUN Database: Large-scale scene recognition from abbey to zoo,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010. [8](#)
- [47] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proc. Int’l Conf. on Machine Learning (ICML)*, 2016, p. 478–487. [2](#)
- [48] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” *arXiv preprint arXiv:1904.12848*, 2019. [7](#)
- [49] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 818–833. [1](#)
- [50] R. Zhang, P. Isola, and A. A. Efros, “Split-brain autoencoders: Unsupervised learning by cross-channel prediction,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 645–654. [1](#), [2](#)
- [51] R. Zhang, P. Isola, and A. Efros, “Colorful image colorization,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2016. [2](#)