SynCo: Synthetic Hard Negatives in Contrastive Learning for Better Unsupervised Visual Representations

Nikolaos Giakoumoglou
Imperial College London
London, UK, SW7 2AZ
n.giakoumoglou23@imperial.ac.uk

Tania Stathaki
Imperial College London
London, UK, SW7 2AZ
t.stathaki@imperial.ac.uk

Abstract

*Contrastive learning has become a dominant approach in self-supervised visual representation learning. Hard negatives - samples closely resembling the anchor - are key to enhancing learned representations' discriminative power. However, efficiently leveraging hard negatives remains challenging. We introduce SynCo (**Syn**thetic **n**egatives in **Co**ntrastive learning), a novel approach that improves model performance by generating synthetic hard negatives on the representation space. Building on the MoCo framework, SynCo introduces six strategies for creating diverse synthetic hard negatives on-the-fly with minimal computational overhead. SynCo achieves faster training and better representation learning, reaching 67.9% top-1 accuracy on ImageNet ILSVRC-2012 linear evaluation after 200 pretraining epochs, surpassing MoCo's 67.5% using the same ResNet-50 encoder. It also transfers more effectively to detection tasks: on PASCAL VOC, it outperforms both the supervised baseline and MoCo with 82.5% AP; on COCO, it sets new benchmarks with 40.9% AP for bounding box detection and 35.5% AP for instance segmentation. Our synthetic hard negative generation approach significantly enhances visual representations learned through self-supervised contrastive learning. Code is available at* `https://github.com/giakoumoglou/synco`.

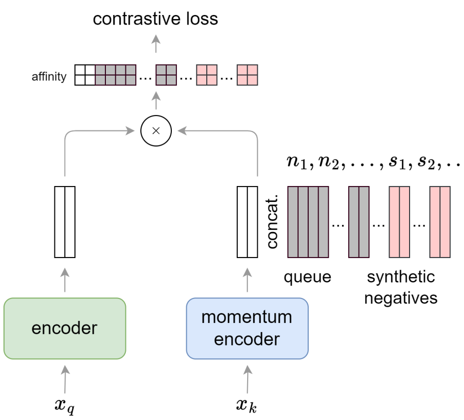


Figure 1. SynCo extends MoCo [9, 21] by introducing synthetic hard negatives generated on-the-fly from a memory queue. The process begins with two augmented views of an image, $\mathbf{x}_q$ and $\mathbf{x}_k$, processed by an encoder and a momentum encoder, respectively, producing feature vectors $\mathbf{q}$ and $\mathbf{k}$. The memory queue holds negative samples $\mathbf{n}_1, \mathbf{n}_2, \ldots$, which are concatenated with synthetic hard negatives $\mathbf{s}_1, \mathbf{s}_2, \ldots$ generated using the SynCo strategies. These combined negatives are used to compute the affinity matrix, which, together with the positive pair (query $\mathbf{q}$ and key $\mathbf{k}$), contributes to the InfoNCE loss calculation.

1. Introduction

Contrastive learning has emerged as a prominent approach in self-supervised learning, significantly advancing representation learning from unlabeled data. This technique, which discriminates between similar and dissimilar data pairs, has shown premise in visual representation tasks. Seminal works such as SimCLR [7] and MoCo [21] established instance discrimination as a pretext task. These methods generate multiple views of the same data point through augmentation, training the model to minimize the distance between positive pairs (augmented views of the same instance) while maximizing it for negative pairs (views of different instances).

Despite its effectiveness, instance discrimination faces challenges. A key limitation is the need for numerous negative samples, often leading to increased computational costs. For example, SimCLR requires large batch sizes for sufficient negatives [7]. While approaches like MoCo address some issues through dynamic queues and momentum encoders [9, 21], they still face challenges in selecting and maintaining high-quality hard negatives.

Recent studies have highlighted the importance of carefully crafted data augmentations in learning robust representations [3, 7, 9, 12, 24, 36, 41, 47]. These transformations likely provide more diverse, challenging copies of images, increasing the difficulty of the self-supervised task. This self-supervised task is a pretext problem (e.g., predicting image rotations [14] or solving jigsaw puzzles [34]) designed to induce learning of generalizable features without explicit labels. Moreover, techniques that combine data at the pixel level [53, 56] or at the feature level [45] have proven effective in helping models learn more resilient features, leading to improvements in both fully supervised and semi-supervised tasks.

The concept of challenging negative samples has been explored as a way to enhance contrastive learning models. These samples, which lie close to the decision boundary, are crucial for refining the model's discriminative abilities. Recent work like MoCHI [24] has shown improvements by incorporating harder negatives. Our work builds on this idea by proposing novel strategies for synthetic hard negative generation.

In this paper, we present SynCo (***Sy****nthetic* ***n****egatives in* ***Co****ntrastive learning*), a novel approach to contrastive learning that leverages synthetic hard negatives to enhance the learning process. Building on the foundations of MoCo, SynCo introduces six distinct strategies for generating synthetic hard negatives, each designed to provide diverse and challenging contrasts to the model. These strategies include: interpolated negatives; extrapolated negatives; mixup negatives; noise-injected negatives; perturbed negatives; and adversarial negatives. By incorporating these synthetic samples, SynCo aims to push the boundaries of contrastive learning, improving both the efficiency and effectiveness of the training process.

The main **contributions** of our work are as follows:

1. We introduce SynCo, a novel contrastive learning approach that improves representation learning by leveraging synthetic hard negatives. SynCo enhances the discriminative capabilities of models by generating challenging negatives on-the-fly from a memory queue, using six distinct strategies that target different aspects of the feature space to provide a comprehensive approach to hard negative generation. This process improves model performance without significant increases in computational demands, achieving faster training and better representation learning.
2. We empirically show improved downstream performance on ImageNet ILSVRC-2012 by incorporating synthetic hard negatives, demonstrating improvements in both linear evaluation and semi-supervised learning tasks.
3. We show that SynCo learns stronger representations by measuring their transfer learning capabilities COCO and PASCAL VOC detection, where it outperforms both the supervised baseline and MoCo.

The paper is structured as follows: Section 2 reviews related work; Section 3 explores hard negatives in contrastive learning; Section 4 introduces our synthetic hard negatives method; Section 5 presents experimental results; and Section 6 concludes the paper.

2. Related Work

2.1. Contrastive Learning

Recent contrastive learning methods focus on instance discrimination as a pretext task, treating each image as its own class. The core principle involves bringing an anchor and a "positive" sample closer in the representation space while pushing the anchor away from "negative" samples [7, 25]. Positive pairs are typically created through multiple views of each data point [41], using techniques such as color decomposition [40], random augmentation [7, 21], image patches [44], or student-teacher model representations [6, 15, 35]. The common training objective, based on InfoNCE [44] or its variants [7, 12, 43, 52], aims to maximize mutual information [2, 22], necessitating numerous negative pairs. While some approaches use large batch sizes [7] to address this, others like MoCo [9, 21], PIRL [32], and InstDis [49] employ memory structures. Recent advancements explore strategies such as invariance regularizers [33], dataset-derived positives [12], and unified contrastive formulas [39]. Some methods eliminate negative samples through asymmetric Siamese structures or normalization [6, 8, 15], while others prevent model collapse via redundancy reduction [54] or regularization [4]. Despite these innovations, the fine-grained nature of instance discrimination can lead to false-negative pairs [58].

2.2. Hard Negatives

Hard negatives are critical in contrastive learning as they improve the quality of visual representations by helping to define the representation space more effectively. These challenging yet relevant samples are harder to distinguish from the anchor point, enabling the model to better differentiate between similar features. The use of hard negatives involves selecting samples that are similar to positive samples but different enough to aid in learning distinctive features. Dynamic sampling of hard negatives during training prevents the model from easily minimizing the loss, enhancing its learning capabilities [7, 21]. Various approaches have been proposed to leverage hard negatives effectively. For instance, MoCo [21] utilizes a dynamic queue and momentum-based encoder updates to maintain fresh and challenging negatives throughout training. Other methods, such as SimCLR [7] and InfoMin [41], suggest adjusting the difficulty of negative samples by varying data augmentation techniques. This progressive increase in task difficulty benefits the training process. Building on these ideas, MoCHI [24] has explored integrating hard negative mixing into existing frameworks to further improve performance. By employing these methods, models become more adept at handling detailed and complex tasks, ensuring each negative sample significantly contributes to optimizing learning outcomes and boosting overall model effectiveness.

2.3. Synthetic Features

Synthetic feature generation is a widely used method to enhance deep learning models, especially in cases with limited labeled data. By adding synthetic features to the representation space, models improve in generalization and performance. Some methods have generated features for unseen classes using generative models [18, 38, 50], while others have integrated these into self-supervised and contrastive learning frameworks [29, 57]. Models that combine generation with representation spaces have also shown success in zero-shot learning [17]. In contrast, our approach directly generates synthetic hard negatives in contrastive learning, improving representation without additional generative models.

3. Preliminaries

3.1. Contrastive Learning

Contrastive learning seeks to differentiate between similar and dissimilar data pairs, often treated as a dictionary look-up where representations are optimized to align positively paired data through contrastive loss in the representation space [21]. Given an image x, and a distribution of image augmentation $\mathcal{T}$, we create two augmented views of the same image using the transformation $t_q, t_k \sim \mathcal{T}$, i.e., $x_q = t_q(x)$ and $x_k = t_k(x)$. Two encoders, f_q and f_k, namely the query and key encoders, generate the vectors $\mathbf{q} = f_q(x_q)$ and $\mathbf{k} = f_k(x_k)$ respectively. The learning objective minimizes a contrastive loss using the InfoNCE criterion [44]:

$$\mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q}) = -\log \frac{\exp(\mathbf{q}^T \cdot \mathbf{k}/\tau)}{\exp(\mathbf{q}^T \cdot \mathbf{k}/\tau) + \sum_{\mathbf{n} \in \mathcal{Q}} \exp(\mathbf{q}^T \cdot \mathbf{n}/\tau)} \tag{1}$$

Here, $\mathbf{k}$ is f_k's output from the same augmented image as $\mathbf{q}$, and $\mathcal{Q} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_K\}$ includes outputs from different images, representing negative samples of size K. The temperature parameter τ adjusts scaling for the ℓ_2-normalized vectors $\mathbf{q}$ and $\mathbf{k}$. The key encoder f_k can be updated in two ways. In the synchronized update approach, f_k is updated synchronously with f_q, maintaining identical weights throughout training [7]. Alternatively, a momentum update scheme can be employed, where f_k is updated using the equation: $\theta_k \leftarrow m \cdot \theta_k + (1 - m) \cdot \theta_q$ [21]. Here, θ_k and θ_q are the parameters of f_k and f_q respectively, and $m \in [0, 1]$ is the momentum coefficient. This momentum approach allows f_k to evolve more slowly, providing more consistent negative samples over time and potentially stabilizing the learning process. The memory bank $\mathcal{Q}$ can be defined in various ways, such as an external memory of all dataset images [32, 40, 49], a queue of recent batches [21], or the current minibatch [7].

The gradient of the contrastive loss in Equation (1) with respect to the query $\mathbf{q}$ is given by:

$$\frac{\partial \mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q})}{\partial \mathbf{q}} = -\frac{1}{\tau} \left((1 - p_k) \cdot \mathbf{k} - \sum_{\mathbf{n} \in \mathcal{Q}} p_n \cdot \mathbf{n} \right) \tag{2}$$

where

$$p_{z_i} = \frac{\exp(\mathbf{q}^T \cdot \mathbf{z_i}/\tau)}{\sum_{j \in Z} \exp(\mathbf{q}^T \cdot \mathbf{z_j}/\tau)} \tag{3}$$

with $\mathbf{z_i}$ being a member of the set $\mathcal{Q} \cup \{\mathbf{k}\}$. The positive and negative logits contribute to the loss similarly to a $(K + 1)$-way cross-entropy classification, with the key logit representing the query's latent class [1].

3.2. Understanding Hard Negatives

The effectiveness of contrastive learning approaches hinges critically on the utilization of hard negatives [1, 16, 23, 24, 31, 49]. Current approaches face significant challenges in efficiently leveraging these hard negatives. Sampling from within the same batch necessitates larger batch sizes [7, 10], potentially straining computational resources. Conversely, maintaining a memory bank containing representations of the entire dataset incurs substantial computational overhead in keeping the memory up-to-date [9, 21, 32, 49]. These limitations underscore the need for more efficient strategies to generate and utilize hard negatives in contrastive learning frameworks.

Hardness of negatives. The "hardness" of negative samples, defined by their similarity to positive samples in the representation space, determines how challenging they are for the model to differentiate, directly impacting the effectiveness of the contrastive learning process. Figure 2 illustrates the evolution of negative sample hardness during MoCo-v2 training on ImageNet-100. The plot depicts the top 1024 matching probabilities p_{z_i} across different training epochs. Initially, the distribution of these probabilities is relatively uniform. However, as training progresses, a clear trend emerges: fewer negatives contribute significantly to the loss function. This observation suggests that the model rapidly learns to distinguish most negatives, leaving only a small subset that remains challenging. Such a phenomenon underscores the importance of maintaining a diverse pool of hard negatives throughout the training process to sustain effective learning [24].

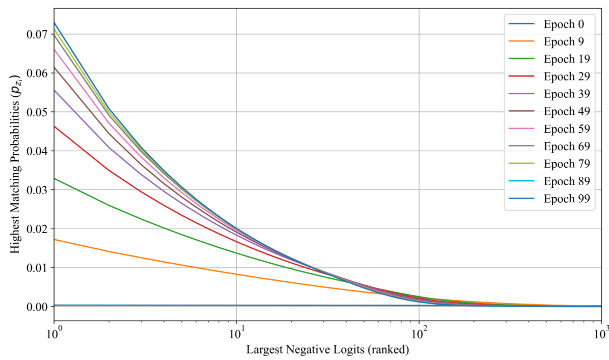


Figure 2. Histogram of the top 1024 matching probabilities p_{z_i}, $z_i \in \mathcal{Q}$ for MoCo-v2, over various epochs. Logits are organized in descending order, and each line indicates the mean matching probability across all queries [24].

Difficulty of the proxy task. The difficulty of the proxy task in contrastive learning, typically defined by the self-supervised objective, significantly influences the quality of learned representations. Figure 3 compares the proxy task performance of MoCo and MoCo-v2 on ImageNet-100, measured by the percentage of queries where the key ranks above all negatives. Notably, MoCo-v2, which employs more aggressive augmentations, exhibits lower proxy task performance compared to MoCo, indicating a more challenging learning objective. Paradoxically, this increased difficulty correlates with improved performance on downstream tasks such as linear classification [24]. This counterintuitive relationship between proxy task difficulty and downstream performance suggests that more challenging self-supervised objectives can lead to the learning of more robust and transferable representations, motivating the development of strategies to dynamically modulate task difficulty during training.

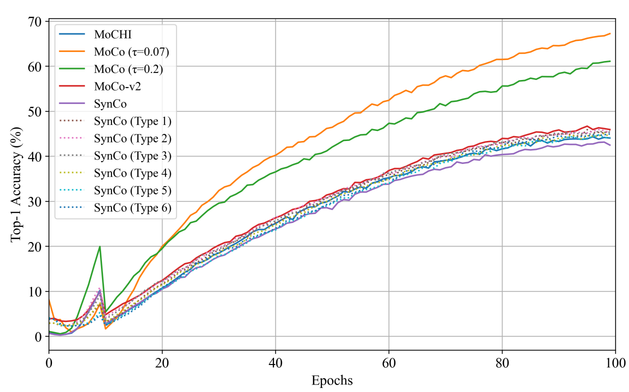


Figure 3. Performance comparison of MoCo, MoCo-v2, MoCHI, and SynCo (under various configurations) on ImageNet-100 in terms of accuracy on the proxy task (percentage of queries where the key is ranked higher than all negatives).

4. Synthetic Hard Negatives in Contrastive Learning

In this section, we present an approach for generating synthetic hard negatives in the representation space using six distinct strategies. We refer to our proposed approach as SynCo ("***Sy**nthetic **n**egatives in **Co**ntrastive learning*").

4.1. Generating Synthetic Hard Negatives

Let $\mathbf{q}$ represent the query image, $\mathbf{k}$ its corresponding key, and $\mathbf{n} \in \mathcal{Q}$ denote the negative features from a memory structure of size K. The loss associated with the query is computed using the logits $\ell(\mathbf{z_i}) = \mathbf{q}^T \cdot \mathbf{z_i}/\tau$, which are processed through a softmax function. We define $\hat{\mathcal{Q}} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K\}$ as the ordered set of all negative features, where $\ell(\mathbf{n}_i) > \ell(\mathbf{n}_j)$ for all $i < j$, implying that the negative features are sorted based on decreasing similarity to the query. The most challenging negatives are selected by truncating the ordered set $\hat{\mathcal{Q}}$, retaining only the first $N < K$ elements, denoted as $\hat{\mathcal{Q}}^N$.

Interpolated synthetic negatives (type 1). For each query $\mathbf{q}$, we propose to generate N_1 synthetic hard negative features by mixing the query $\mathbf{q}$ with a randomly chosen feature from the N hardest negatives in $\hat{\mathcal{Q}}^N$. Let $S^1 = \{\mathbf{s}_1^1, \mathbf{s}_2^1, \dots, \mathbf{s}_{N_1}^1\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^1 \in S^1$ would be given by:

$$\mathbf{s}_k^1 = \alpha_k \cdot \mathbf{q} + (1 - \alpha_k) \cdot \mathbf{n}_i, \quad \alpha_k \in (0, \alpha_{\max}) \tag{4}$$

where $\mathbf{n}_i \in \hat{\mathcal{Q}}^N$ and α_k is randomly sampled from a uniform distribution in the range $(0, \alpha_{\max})$. The resulting synthetic hard negatives are then normalized and added to the set of negative logits for the query. Interpolation creates a synthetic embedding that lies between the query and the negative in the representation space. We set $\alpha_{\max} = 0.5$ to guarantee that the contribution of the query is always less than that of the negative. This is similar to the hardest negatives (type 2) of MoCHI [24].

Extrapolated synthetic negatives (type 2). For each query $\mathbf{q}$, we propose to generate N_2 hard negative features by extrapolating beyond the query embedding in the direction of the hardest negative features. Similar to the interpolated method, we use a randomly chosen feature from the N hardest negatives in $\hat{\mathcal{Q}}^N$. Let $S^2 = \{\mathbf{s}_1^2, \mathbf{s}_2^2, \dots, \mathbf{s}_{N_2}^2\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^2 \in S^2$ would be given by:

$$\mathbf{s}_k^2 = \mathbf{n}_i + \beta_k \cdot (\mathbf{n}_i - \mathbf{q}), \quad \beta_k \in (1, \beta_{\max}) \tag{5}$$

where $\mathbf{n}_i \in \hat{\mathcal{Q}}^N$ and β_k is randomly sampled from a uniform distribution in the range $(1, \beta_{\max})$. These synthetic features are also normalized and used to enhance the negative logits. Extrapolation generates a synthetic embedding that lies beyond the query embedding in the direction of the hardest negative. We choose $\beta_{\text{max}} = 1.5$.

Mixup synthetic negatives (type 3). For each query $\mathbf{q}$, we propose to generate N_3 hard negative features by combining pairs of the N hardest existing negative features in $\hat{\mathcal{Q}}^N$. Let $S^3 = \{\mathbf{s}_1^3, \mathbf{s}_2^3, \dots, \mathbf{s}_{N_3}^3\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^3 \in S^3$ would be given by:

$$\mathbf{s}_k^3 = \gamma_k \cdot \mathbf{n}_i + (1 - \gamma_k) \cdot \mathbf{n}_j, \quad \gamma_k \in (0, 1) \tag{6}$$

where $\mathbf{n}_i, \mathbf{n}_j \in \hat{\mathcal{Q}}^N$ and γ_k is randomly sampled from a uniform distribution in the range $(0, 1)$. The resulting synthetic hard negatives are then normalized and added to the set of negative logits for the query. Mixup combines pairs of the hardest existing negative features to create a synthetic embedding that represents a blend of challenging cases. This is similar to the hard negatives (type 1) of MoCHI [24].

Noise-injected synthetic negatives (type 4). For each query $\mathbf{q}$, we propose to generate N_4 hard negative features by adding Gaussian noise to the hardest negative features. Using the top N hardest negatives $\hat{\mathcal{Q}}^N$, let $S^4 = \{\mathbf{s}_1^4, \mathbf{s}_2^4, \dots, \mathbf{s}_{N_4}^4\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^4 \in S^4$ would be given by:

$$\mathbf{s}_k^4 = \mathbf{n}_i + \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I}) \tag{7}$$

where $\mathbf{n}_i \in \hat{\mathcal{Q}}^N$ and $\mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I})$ represents Gaussian noise with standard deviation σ (where $\mathbf{I}$ is the identity matrix). The noisy negatives are normalized before being used in the loss calculation. Noise injection adds Gaussian noise to the hardest negative features, resulting in a synthetic embedding with added randomness.

Perturbed synthetic negatives (type 5). For each query $\mathbf{q}$, we propose to generate N_5 hard negative features by perturbing the embeddings of the hardest negative features. Given the top N hardest negatives $\hat{\mathcal{Q}}^N$, let $S^5 = \{\mathbf{s}_1^5, \mathbf{s}_2^5, \dots, \mathbf{s}_{N_5}^5\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^5 \in S^5$ would be given by:

$$\mathbf{s}_k^5 = \mathbf{n}_i + \delta \cdot \nabla_{\mathbf{n}_i} \text{sim}(\mathbf{q}, \mathbf{n}_i) \tag{8}$$

where $\mathbf{n}_i \in \hat{\mathcal{Q}}^N$ and $\text{sim}(\cdot, \cdot)$ is the similarity function and δ controls the perturbation magnitude. The perturbed embeddings are then normalized and added to the negative logits. Perturbation modifies the embeddings of the hardest negative features based on the gradient of the similarity function, creating synthetic negatives that are slightly adjusted to be more challenging for the model.

Adversarial synthetic negatives (type 6). For each query $\mathbf{q}$, we propose to generate N_6 hard negative features by applying adversarial perturbations to the hardest negative features to maximize their similarity to the query embeddings. Using the top N hardest negatives $\hat{\mathcal{Q}}^N$, let $S^6 = \{\mathbf{s}_1^6, \mathbf{s}_2^6, \dots, \mathbf{s}_{N_6}^6\}$ be the set of synthetic negatives to be generated. Then a synthetic negative feature $\mathbf{s}_k^6 \in S^6$ would be given by:

$$\mathbf{s}_k^6 = \mathbf{n}_i + \eta \cdot \text{sign}(\nabla_{\mathbf{n}_i} \text{sim}(\mathbf{q}, \mathbf{n}_i)) \tag{9}$$

where $\mathbf{n}_i \in \hat{\mathcal{Q}}^N$ and η controls the perturbation magnitude. The perturbed embeddings are normalized and added to the negative logits. Adversarial hard negatives apply adversarial perturbations to the hardest negative features, specifically altering them to maximize their similarity to the query embeddings, thereby producing the most challenging contrasts.

4.2. Integrating Synthetic Hard Negatives into the Contrastive Loss

The synthetic hard negatives generated are integrated into the contrastive learning process by modifying the InfoNCE loss. Let $\mathcal{S} = \bigcup_{i=1}^{6} S^i$ represent the concatenation of all synthetic hard negatives, where S^i is the set of synthetic negatives generated by the i-th strategy. This combined set of synthetic negatives augments the original negatives $\mathcal{Q}$, providing a more diverse and challenging set of contrasts for the query. The modified InfoNCE loss is given by:

$$\mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q}, \mathcal{S}) = -\log \frac{\exp(\mathbf{q}^T \cdot \mathbf{k}/\tau)}{Z + \sum_{\mathbf{s} \in \mathcal{S}} \exp(\mathbf{q}^T \cdot \mathbf{s}/\tau)} \tag{10}$$

where $Z = \exp(\mathbf{q}^T \cdot \mathbf{k}/\tau) + \sum_{\mathbf{n} \in \mathcal{Q}} \exp(\mathbf{q}^T \cdot \mathbf{n}/\tau)$, τ is the temperature parameter, $\mathcal{Q}$ is the set of original negatives, and $\mathcal{S}$ is the set of synthetic hard negatives. By incorporating both real and synthetic negatives, the model is exposed to a wider variety of challenging examples, which encourages learning more robust and generalizable representations. The overall computational overhead of SynCo is roughly equivalent to increasing the queue/memory by $\sum_{i=1}^{6} N_i \ll K$, along with the additional cost of generating the synthetic negatives.

4.3. Discussion on Synthetic Hard Negatives

The following explores the effects of incorporating synthetic hard negatives on the difficulty of the proxy task and how they influence the usage of the representation space.

Is the proxy task more difficult? Figure 3 depicts the proxy task performance for different configurations of SynCo. We observe that incorporating synthetic negatives leads to faster learning and improved performance. Each type of synthetic negative accelerates learning compared to the MoCo-v2 baseline, with the full SynCo configuration showing the most significant improvement (see Figure 5a) and the lowest final proxy task performance. This indicates that SynCo presents the most challenging proxy task. This is evidenced by $\max \ell(\mathbf{s}_k^i) > \max \ell(\mathbf{n}_j)$, where $\mathbf{s}_k^i \in S^i$ are synthetic negatives and $\mathbf{n}_j \in \tilde{\mathcal{Q}}_N$ are original negatives. Through SynCo, we modulate proxy task difficulty via synthetic negatives, pushing the model to learn more robust features.

Evaluating the usage of the representation space. To assess learned representations, we employ alignment and uniformity metrics proposed by [46]. These metrics provide insights into representation space utilization, with alignment quantifying the grouping of similar samples and uniformity measuring representation spread across the hypersphere. Figure 4 presents results for various models using features from the ImageNet-100 validation set. Our findings demonstrate that SynCo significantly enhances the uniformity of representations compared to MoCo-v2 and MoCHI, indicating its superior ability to utilize the representation space in the proxy task. Furthermore, the incorporation of synthetic negatives (types 1 to 6) leads to improved alignment. These results suggest that SynCo's approach to synthetic negative generation and contrastive learning yields more effective and well-distributed feature representations.

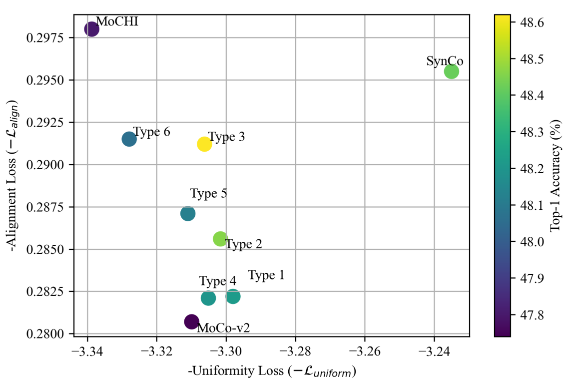


Figure 4. Performance comparison of MoCo-v2, MoCHI, and SynCo (under various configurations) on ImageNet-100 in terms of alignment and uniformity metrics. The x-axis and y-axis represent $-\mathcal{L}_{\text{uniform}}$ and $-\mathcal{L}_{\text{align}}$, respectively. The model with the highest performance is located in the upper-right corner of the chart. We use $K = 65$k.

5. Experiments

5.1. Implementation Details

We pretrain SynCo on ImageNet ILSVRC-2012 [11] and its smaller ImageNet-100 subset [40] for ablation studies using a ResNet-50 [19] encoder. Our implementation builds upon MoCo-v2 [9]. For training, unless stated otherwise, we use $K = 65$k ($K = 16$k for ImageNet-100). For SynCo, we also have a warm-up of 10 epochs, i.e. for the first epochs we do not synthesize hard negatives. We set SynCo's hyperparameters σ, δ, and η to 0.01. For hard negative generation, we use the top $N = 1024$ hardest negatives, with $N_1 = N_2 = N_3 = 256$ and $N_4 = N_5 = N_6 = 64$. For ImageNet linear evaluation, we train a linear classifier on frozen features for 100 epochs, using a batch size of 256 and a cosine learning rate schedule. Initial learning rates are set to 30.0 for ImageNet and 10.0 for ImageNet-100. To evaluate transfer learning, we apply SynCo to object detection tasks. For PASCAL VOC [13], we fine-tune a Faster R-CNN [37] on `trainval07+12` and test on `test2007`. For COCO [30], we use a Mask R-CNN [20], fine-tuning on `train2017` and evaluating on `val2017`. We employ Detectron2 [48] and report standard AP metrics, following [21] without additional hyperparameter tuning. Detailed implementation details are provided in the supplementary material.

5.2. Linear Evaluation on ImageNet

We evaluate the SynCo representation by training a linear classifier on top of the frozen features that were pretrained on ImageNet, following the procedure described in [7, 15, 26, 27, 44] (details in supplementary material). SynCo obtains 67.9% $\pm$ 0.16% top-1 accuracy and 88.0% $\pm$ 0.05% top-5 accuracy after 200 epoch pretraining (Table 1). Specifically, SynCo achieves a +0.4% top-1 accuracy improvement over MoCo-v2 and a +1.0% improvement over MoCHI. While MoCHI, which employs hard negatives, achieves lower performance than MoCo-v2, our method, which generates synthetic hard negatives, not only avoids this drop in performance but actually improves it. SynCo also surpasses the state-of-the-art methods SimCLR and SimSiam. When training for 800 epochs, SynCo obtains 70.6% top-1 accuracy (89.8% top-5 accuracy) (Table 2). This represents a +1.9% improvement in top-1 accuracy over MoCHI. These results show that SynCo achieves higher accuracy than its competitors with only a minor computational overhead for generating synthetic hard negatives, while also enabling faster training convergence.

5.3. Semi-supervised Training on ImageNet

We evaluate SynCo in a semi-supervised setting using 1% and 10% of labeled ImageNet data, following protocols from [7, 15, 27, 44, 55] (details in supplementary material). We use the same data splits as [7]. Results in Table 3 show SynCo's competitive performance. With 1% labeled data, SynCo achieves 50.8%$\pm$0.21% top-1 accuracy and 77.5%$\pm$ 0.12% top-5 accuracy. Using 10% labeled data, it reaches 66.6% $\pm$ 0.19% top-1 accuracy and 88.0% $\pm$ 0.10% top-5

Table 1. Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet ILSVRC-2012 with 200 epochs of pretraining using ResNet-50. Results for SynCo are averaged over 3 runs.

Method	Top-1	Top-5
Supervised	76.5	-
PIRL [32]	63.6	-
LA [58]	60.2	-
InfoMin [41]	70.1	89.4
SimSiam [8]	68.1	-
MoCo [21]	60.7	-
MoCo-v2 [9]	67.5	90.1
PCL-v2 [28]	67.6	-
MoCo-v2 + DCL [52]	67.6	-
SimCLR-v2 + DCL [52]	65.8	-
MoCHI [24]	66.9	-
SynCo (ours)	67.9	88.0

Table 2. Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet ILSVRC-2012 for models trained with extended epochs using ResNet-50. Results for SynCo are based on 1 run.

Method	Epochs	Top-1	Top-5
PIRL [32]	800	63.6	-
InfoMin [41]	800	73.0	91.1
BYOL [15]	1000	74.3	91.6
SimSiam [8]	800	68.1	-
SimCLR [7]	1000	69.3	-
BT [54]	1000	73.2	91.0
MoCo-v2 [9]	800	71.1	90.1
MoCHI [24]	800	68.7	-
SynCo (ours)	800	70.6	89.8

accuracy. We observe that despite a lower volume of labeled data, SynCo competes well with established semi-supervised methods like SimCLR, BYOL, and Barlow Twins.

5.4. Transferring to Detection

We evaluate the SynCo representation, pretrained for 200 epochs, by applying it to the detection task. We follow the protocol of [21]. Detailed configurations for object detection experiments are provided in in the supplementary material. Table 4 presents the results of object detection on PASCAL VOC and COCO datasets. SynCo displays consistent gains over both the supervised baseline and MoCo-v2. Specifically, SynCo, surpasses MoCo-v2 and performs on par with MoCHI in the PASCAL VOC detection task. On the more challenging COCO dataset, SynCo achieves new state-of-the-art results in both bounding box detection and instance segmentation. Additional results comparing SynCo and MoCo-v2 at different training epochs are presented in the supplementary material.

Table 3. Semi-supervised learning on ImageNet ILSVRC-2012 with 1% and 10% training examples using ResNet-50. Results for SynCo are averaged over 3 runs.

Method	Epochs	Top-1		Top-5	
		1%	10%	1%	10%
Supervised		25.4	56.4	48.4	80.4
InstDis [49]	200	-	-	39.2	77.4
PIRL [32]	800	30.7	60.4	57.2	83.8
SimCLR [7]	1000	48.3	65.6	75.5	87.8
BT [54]	1000	55.0	69.7	79.2	89.3
BYOL [15]	1000	53.2	68.8	78.4	89.0
SwAV [5]	800	53.9	70.2	78.5	89.9
SynCo (ours)	800	50.8	66.6	77.5	88.0

5.5. Ablation Study

We perform ablations studies on ImageNet-100. The results of our ablations are presented in Figure 5. Our findings consistently demonstrate that various SynCo configurations outperform the MoCo-v2 baseline. Additional ablation studies and analyses are presented in the supplementary material.

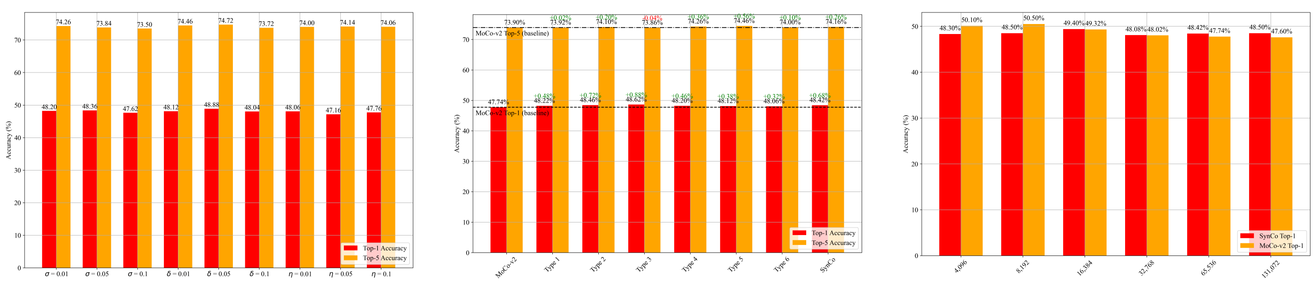
Ablation on type of hard negative. We evaluate the impact of each synthetic hard negative type on pretraining. For this, we select the top $N = 1024$ hardest negatives and generate $N_i = 256$, $i = 1, 2, \ldots, 6$ negatives. We train SynCo without hard negatives (equivalent to MoCo-v2) for 100 epochs and measure top-1 and top-5 accuracy. Subsequently, we train SynCo using each type of hard negative individually, and then using all six types in combination. The results of these ablations are presented in Figure 5a. We see that every SynCo configuration outperform the MoCo-v2 baseline.

Ablation on hyperparameters. We conducted ablations on the parameters σ, δ, and η of SynCo's type 4, type 5, and type 6 negatives, respectively. The results, presented in Figure 5b, show that varying these parameters does not lead to significant differences in performance. This suggests that SynCo is robust across a wide range of values for σ, δ, η.

Ablation on queue size. We investigate the effect of queue size Q on performance. We train SynCo and MoCo-v2 with reduced queue sizes. Our results, presented in Figure 5c, reveal that SynCo performs comparably to MoCo-v2 across various queue sizes. With smaller queues, SynCo underperforms compared to MoCo-v2. This can be attributed to the fact that the total generated synthetic negatives are too hard for the task and harm performance, a finding that is also observed in [24]. However, as the queue increases, SynCo performs on par with MoCo-v2. At the largest queue size tested, SynCo outperforms MoCo-v2.

Table 4. Results for object detection on PASCAL VOC and COCO. All models use a ResNet-50 backbone. We report AP, AP_{50}, and AP_{75}, which are standard COCO metrics. bb denotes bounding box detection, and msk denotes instance segmentation. The values in bold indicate the maximum of each column. Results for SynCo are averaged over 3 runs.

Method	VOC07+12 detection			COCO detection			COCO segmentation		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{msk}	AP^{msk}_{50}	AP^{msk}_{75}
Random init	16.8	35.9	13.0	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	53.5	81.3	58.8	38.2	58.2	41.6	33.3	54.7	35.2
InstDis [49]	-	-	-	37.4	57.6	40.6	34.1	54.6	36.4
PIRL [32]	55.5	81.0	61.3	38.5	57.6	41.2	34.0	54.6	36.2
InfoMin [41]	**57.6**	**82.7**	**64.6**	39.0	58.5	42.0	34.1	55.2	36.3
SwAV [5]	56.1	82.6	62.7	38.2	58.2	41.3	33.8	55.2	35.9
SimSiam [8]	57.0	82.4	63.7	39.2	59.3	42.1	34.4	56.0	36.7
BT [54]	56.8	82.6	63.4	39.2	59.0	42.5	34.3	56.0	36.5
MoCo [21]	55.5	81.5	61.3	38.5	58.3	41.6	33.6	54.8	35.6
MoCo-v2 [9]	57.0	82.4	63.6	39.0	58.6	41.9	34.2	55.4	36.2
MoCHI [24]	57.5	82.7	64.4	39.2	58.9	42.4	34.4	55.5	36.6
SynCo (ours)	57.2	82.6	63.9	**41.0**	**60.6**	**44.8**	**35.7**	**57.4**	**38.1**



(a) (b) (c)

Figure 5. Ablation studies on ImageNet-100 in terms of Top-1 and Top-5 accuracies (in %). (a) Performance comparison of different types of hard negatives. (b) Performance comparison of different values for σ, δ, and η on SynCo's type 4, type 5, and type 6 hard negatives, respectively. (c) Comparison of SynCo and MoCo-v2 across different queue sizes. For all ablations we use $K = 65$k.

6. Conclusion

This paper introduces SynCo, a novel approach to contrastive learning that leverages synthetic hard negatives to enhance visual representation learning. By generating diverse and challenging negative samples on-the-fly, SynCo addresses the limitations of existing methods in maintaining a pool of effective hard negatives throughout training. Our comprehensive experiments demonstrate that SynCo not only accelerates the learning process but also leads to more robust and transferable representations. The effectiveness of SynCo is evidenced by its superior performance across various benchmarks, including linear evaluation on ImageNet, semi-supervised learning tasks, and transfer learning to object detection on PASCAL VOC and COCO datasets.

While our experiments primarily utilized the MoCo framework, the proposed hard negative generation strategies are general and can be applied to any contrastive learning method that benefits from hard negatives, including SimCLR [7], CPC [44], PIRL [32], and others [12, 25, 42, 47, 51]. These methods, which employ the InfoNCE loss function (or its variants [7, 12]) and instance discrimination as the pretext task, can benefit from the enhanced hard negative generation strategies proposed by SynCo. The introduction of synthetic hard negatives can provide these methods with more challenging, informative contrasts, potentially leading to better feature representations. Also, SynCo's applicability extends beyond visual representation learning. The concept of hard negatives can be applied to various domains such as natural language processing, audio processing, and other areas where contrastive learning is relevant.

Acknowledgments

We would like to express our gratitude to Andreas Floros for his valuable feedback, particularly his assistance with equations, notations, and insightful discussions that greatly contributed to this work.

References

[1] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019. 3

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views, 2019. 2

[3] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023. 1

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924. Curran Associates, Inc., 2020. 7, 8

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 1, 2, 3, 6, 7, 8

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 2, 7, 8

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020. 1, 2, 3, 6, 7, 8

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021. 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[12] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, 2021. 1, 2, 8

[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2009. 6

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. 2

[15] 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, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 2, 6, 7

[16] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742, 2006. 3

[17] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2371–2381, 2021. 3

[18] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 6

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. 1, 2, 3, 6, 7, 8

[22] 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, 2019. 2

[23] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2018. 3

[24] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020. 1, 2, 3, 4, 5, 7, 8

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021. 2, 8

[26] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning, 2019. 6

[27] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019. 6

[28] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations, 2021. 7

[29] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo Liu, Yang Gao, and Jiebo Luo. Generating representative samples for few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13783–13792, 2021. 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 6

[31] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017. 3

[32] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019. 2, 3, 7, 8

[33] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms, 2020. 2

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2016. 2

[35] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2

[36] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning, 2021. 1

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 6

[38] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019. 3

[39] Chenxin Tao, Honghui Wang, Xizhou Zhu, Jiahua Dong, Shiji Song, Gao Huang, and Jifeng Dai. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework, 2022. 2

[40] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding, 2020. 2, 3, 6

[41] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems*, pages 6827–6839. Curran Associates, Inc., 2020. 1, 2, 7, 8

[42] Yonglong Tian, Olivier J. Henaff, and Aaron van den Oord. Divide and contrast: Self-supervised learning from uncurated data, 2021. 8

[43] 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?, 2022. 2

[44] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. 2, 3, 6, 8

[45] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. *arXiv preprint arXiv:1806.05236*, 2018. 2

[46] 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. 6

[47] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations, 2022. 1, 8

[48] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 6

[49] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018. 2, 3, 7, 8

[50] Yongqin Xian, Thomas Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5542–5551, 2018. 3

[51] Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning, 2021. 8

[52] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022. 2, 7

[53] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 2

[54] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. 2, 7, 8

[55] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020. 6

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[57] Yanzhao Zhang, Cho-Jui Hsieh, and Kai-Wei Wang. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. In *Advances in Neural Information Processing Systems*, pages 2983–2995, 2021. 3

[58] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings, 2019. 2, 7