SynCo-v2: An Empirical Study of Training Self-Supervised Vision Transformers with Synthetic Hard Negatives

Nikolaos Giakoumoglou Andreas Floros Kleanthis Marios Papadopoulos Tania Stathaki
Imperial College London

Code: https://github.com/giakoumoglou/synco-v2

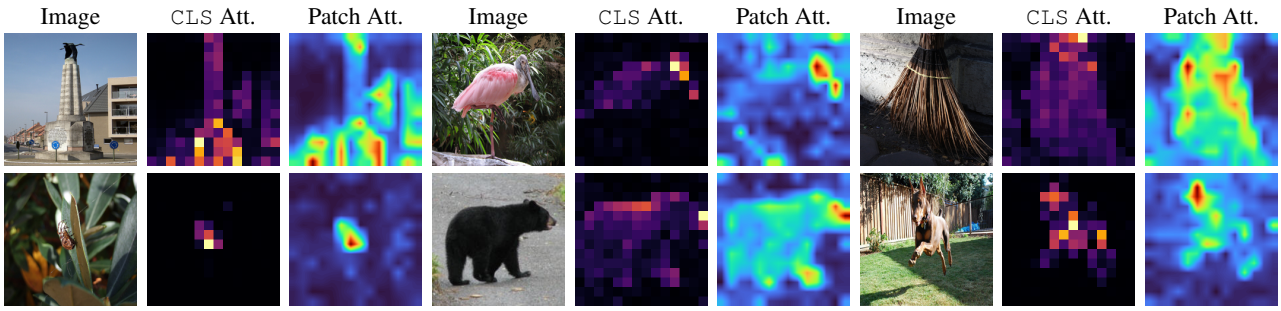


Figure 1. **Vision Transformer (DeiT-Small/16) attention visualization of SynCo-v2.** For each image set, we show the input image, CLS token attention, and patch attention maps from our method. Our approach with memory and synthetic hard negatives produces focused attention on semantically important regions with clear object boundaries and fine-grained details.

Abstract

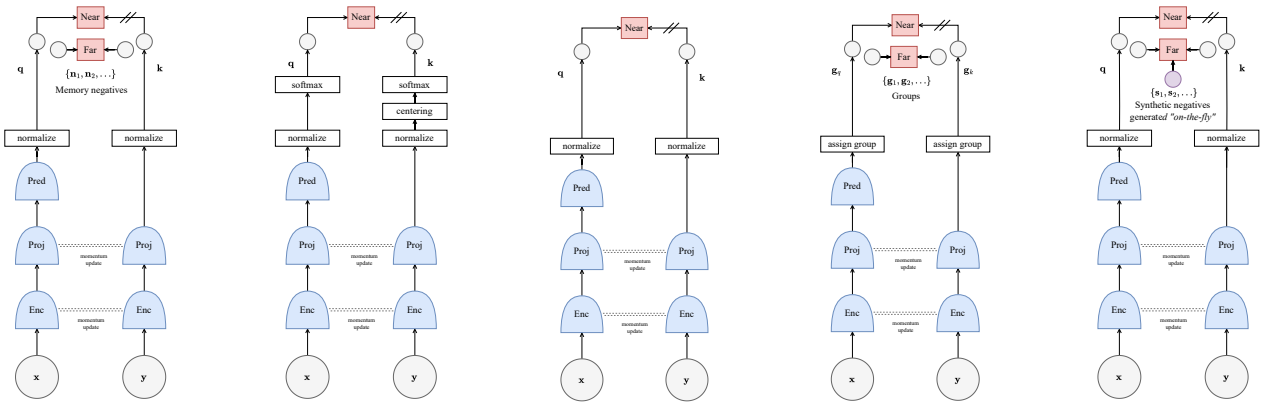
Self-supervised vision transformers are powerful and have achieved impressive results, including generalizing on unseen data and emergent abilities. Current joint embedding architectures can be formulated as either using no negatives (embedding alignment) or using negative samples (contrastive learning). We introduce SynCo-v2, a method that integrates synthetic hard negatives into unsupervised pretraining to improve representation quality. Our approach demonstrates consistent improvements on: ***(i)*** *ImageNet linear evaluation accuracy (improvement up to* ***+6.9%*** *compared to without synthetic negatives),* ***(ii)*** *transfer learning performance across downstream datasets, and* ***(iii)*** *emergent properties where self-supervised vision transformer features contain explicit information about the semantic segmentation of an image and serve as excellent k-NN classifiers (improvement up to* ***+11.3%****). Notably, SynCo-v2 achieves these benefits through simple modifications to existing contrastive frameworks, without requiring specialized techniques like centering, sharpening, or multi-crop training used by DINO [12]. These findings motivate reconsidering contrastive learning as a simpler yet powerful alternative to dominant generative and self-distillation approaches.* [1]

[1]Correspondence: nikos@imperial.ac.uk.

1. Introduction

Transformers [63] have transformed computer vision, establishing themselves as powerful alternatives to ConvNets [21, 43, 61]. Their adoption has been coupled with a training strategy inspired by NLP, *i.e.*, pretraining on large quantities of data and finetuning on the target dataset [21, 61]. As these models scale to billions of parameters and are trained on increasingly large and diverse datasets, they have demonstrated remarkable capabilities in both supervised and self-supervised learning paradigms, achieving state-of-the-art performance across numerous benchmarks [27, 30, 48].

Self-supervised learning has established itself as a powerful approach for visual representation learning, enabling models to extract meaningful patterns from vast amounts of unlabeled data [3, 7, 25, 42]. Self-supervised approaches for vision fall into three categories: **(i)** *pretext task* methods that solve auxiliary tasks such as rotation prediction [26, 47] or jigsaw puzzles [47]; **(ii)** *generative* methods that reconstruct or predict masked portions of inputs, such as MAE [30] inspired by masked language modeling [8, 20, 54, 55], and BEiT [4] following BERT-like pretraining [20, 41]; and **(iii)** *joint embedding architecture* methods that learn representations by comparing different views of data in a shared embedding space [12, 13, 28, 29, 56]. This work focuses on training transformers with joint embedding architectures, unlike prior works using generative methods [4, 30, 50].



(a) *Contrastive* (MoBY) (b) *Self-distillation* (DINO) (c) *Self-distillation* (BYOL) (d) *Clustering* (SMoG) (e) Ours (SynCo-v2)

Figure 2. **Self-supervised learning categories on vision transformers and this paper's contribution.** From left to right: (a) *contrastive learning* method [16, 65]; (b, c) *self-distillation* [12, 28]; (d) *clustering-based* [49]; and (e) SynCo-v2 (ours). Our method introduces synthetic hard negatives generated "*on-the-fly*" to improve contrastive learning methods for vision transformers. Shaded circles ◯ represent observed variables, dashed gray lines indicate the momentum update, // indicates a stop-gradient for backpropagation, white boxes represent functions, and red boxes represent loss functions. **Abbreviations legend:** Enc: encoder, Proj: projector, Pred: predictor.

The joint embedding methods adapted for vision transformers fall into three categories shown in Fig. 2, each using different "*tricks*" to avoid representational collapse: **(i)** *contrastive learning* methods embed different augmented views of the same image into a joint space, maximizing similarity between same-instance embeddings while minimizing similarity across instances [13, 16, 29, 65] (Fig. 2a); **(ii)** *self-distillation* (*teacher–student*) methods train a student to match a teacher's embeddings without negatives [5, 6, 12, 28] (Figs. 2b and 2c); and **(iii)** *clustering-based* methods employ clustering objectives [9–11, 49] (Fig. 2d).

Transformers trained with these joint embedding architecture methods exhibit emergent properties that go beyond standard classification accuracy, as their features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised transformers, nor with convolutional neural networks [12]. Self-distillation methods like DINO [12, 48, 58] and iBOT [71] demonstrate strong unsupervised semantic segmentation, with attention naturally aligning to object boundaries and meaningful regions without supervision.

Despite generative methods achieving higher accuracy [4, 30], joint embedding approaches remain important, achieving competitive accuracy and often surpassing generative methods in linear probing [6, 48]. Among these approaches, contrastive learning methods stand out for their simplicity and computational efficiency [16, 65], explicitly using negative samples to define representation boundaries [13, 29], yet they have received less attention recently.

We seek to answer the simple question:

Can simple modifications to negative sampling strategies in contrastive learning unlock stronger representations and emergent properties for vision transformers comparable to or exceeding those of self-distillation methods?

In this paper, we address this question by integrating synthetic hard negative generation in transformer-based contrastive learning, a strategy previously demonstrated effective for convolutional networks [24, 35] but *not* investigated for vision transformers. Instead of using complex architectures or training schemes (like multi-crop, centering, sharpening, *etc*., see Section 2), we adapt established synthetic negative generation approaches to transformer architectures, generating challenging samples "*on-the-fly*" through *six* complementary transformation strategies [24].

Through an extensive empirical evaluation, we demonstrate that integrating synthetic hard negatives into transformer self-supervised learning leads to three main improvements compared to training *without* synthetic hard negatives or *without* negatives: **(i)** increased top-1 accuracy on ImageNet linear evaluation (Tabs. 1 and 2), achieving 73.1% with DeiT and 75.4% with Swin; **(ii)** improved transfer learning performance across diverse downstream tasks (Tabs. 5 and 6); and **(iii)** strong emergent properties, where self-supervised vision transformer features contain explicit information about the semantic segmentation of an image (Section 4.3), producing precise attention maps that capture object boundaries (Fig. 4) and serve as excellent k-NN classifiers (Tabs. 1 and 2), achieving 71.0% top-1 accuracy with DeiT.

2. Related Work

Joint embedding architectures. *Joint embedding architecture* methods map augmented views into a shared embedding space while avoiding representational collapse through distinct mechanisms. *Contrastive learning* methods prevent collapse using large batch sizes [13] or momentum-encoded memory banks [15, 16, 29, 65] to provide sufficient negative samples. Alternative approaches formalize collapse avoidance via mutual information [31, 59, 62]. *Self-distillation* (*a.k.a. teacher-student distillation*) methods surprisingly avoid collapse without negatives. They use asymmetric architectures [14], momentum updates [12, 28], and stop-gradient operations [12, 14, 28]. Alternatively, they explicitly regularize feature covariance so representations do not collapse, *e.g.*, decorrelate features [5, 69], employ whitening [23], or manifold regularization [68]. Notably, DINO [12, 48, 58], which employs multiple techniques including centering, sharpening, momentum encoder, multi-crop training [11], and extended training, and iBOT [71], which integrates masked patch prediction, exhibit strong unsupervised segmentation. Unlike contrastive learning or self-distillation methods, I-JEPA [2] and V-JEPA [6] avoid collapse via their predictive structure and architectural asymmetry, not through negative samples or variance constraints. Finally, *clustering-based* approaches align embeddings with prototype assignments obtained via the Sinkhorn-Knopp algorithm [11] or via momentum grouping [49].

Contrastive learning. *Contrastive learning* methods treat instance discrimination as a pretext task, treating each image as its own class [13, 29]. The core principle involves bringing an anchor and a "*positive*" sample closer in the embedding space while pushing the anchor away from "*negative*" samples [36]. Training typically employs InfoNCE loss [62] or its variants [13, 22, 60, 67], maximizing mutual information between positive pairs while minimizing it for negatives. Negative samples are drawn from large batch sizes [13] or memory banks [15, 16, 29, 65]. The concept of challenging negative samples has been explored as a way to improve contrastive learning models. These samples, which lie close to the decision boundary, are crucial for refining the model's discriminative abilities [1, 57]. Various strategies leverage hard negatives through mixup-based interpolation between embeddings [35], debiased contrastive losses with theoretical analysis [57], importance reweighting schemes [67], and hardness-aware sampling from memory queues [60]. Systematic synthetic generation through transformation strategies has proven effective for convolutional networks [24]. Our method adapts synthetic hard negative generation to vision transformers by generating diverse, informative negatives "*on-the-fly*" rather than relying solely on batch size or memory bank capacity.

3. Methodology

In this section, we introduce our approach, named SynCo-v2. Our method builds upon existing contrastive learning frameworks (see Figure 2a) and aims to improve representation quality by generating informative negative samples (see Figure 2e). The overall framework of our method is illustrated analytically in Figure 3.

3.1. SynCo-v2

Like other joint embedding methods, SynCo-v2 also operates on the embedding pairs of distorted images. Specifically, given an image $\mathbf{x}$, and two distributions of image augmentation $\mathcal{T}_q, \mathcal{T}_k$, we create two augmented views of the same image using the transformations $t_q \sim \mathcal{T}_q$ and $t_k \sim \mathcal{T}_k$, *i.e*, $\mathbf{x}_q = t_q(x)$ and $\mathbf{x}_k = t_k(x)$.

Then, we use two encoders f_θ and f_ξ, two projectors g_θ and g_ξ, and a predictor h_θ with parameters θ and ξ to generate the corresponding embeddings $\mathbf{q}$ and $\mathbf{k}$, where $\mathbf{q} = h_\theta(g_\theta(f_\theta(\mathbf{x}_q)))$ and $\mathbf{k} = g_\xi(f_\xi(\mathbf{x}_k))$, and $\mathbf{q}, \mathbf{k} \in \mathbb{R}^d$ [28, 65]. We denote the *online* branch as f_θ, g_θ, and h_θ, and the *target* branch as f_ξ and g_ξ, with parameters θ and ξ, respectively. We assume that the outputs are ℓ_2-normalized.

We maintain a memory queue $\mathcal{Q} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K\}$ that consists of features from distinct images, serving as K negative samples [15, 16, 29, 65]. In here we set $K = 4096$. The negative samples $\{\mathbf{n}_i\}_{i=1}^K$ are embeddings generated from the target branch in previous steps and stored in the memory bank, which requires memory size $\mathcal{O}(K \cdot d)$, where d is the embedding dimension.

We use a momentum update such that only the parameter θ is updated through backpropagation, while the parameter ξ is maintained as the exponential moving average of θ:

$$\xi \leftarrow m \cdot \xi + (1 - m) \cdot \theta \tag{1}$$

where $m \in [0, 1]$ is the momentum coefficient that controls the update rate [28, 65]. This momentum mechanism ensures gradual evolution of f_ξ, improving the stability of negative samples across training iterations [29].

To generate synthetic hard negatives, we define the hardness of negative samples by their similarity to the query, measured through the logit values $\ell(\mathbf{n}_i) = \mathbf{q}^\top \cdot \mathbf{n}_i$. To identify the most challenging negatives, we order all negative features by decreasing similarity, *i.e.*, $\hat{\mathcal{Q}} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K\}$ where $\ell(\mathbf{n}_i) > \ell(\mathbf{n}_j)$ for all $i < j$. The top-N hardest negatives are then selected as $\hat{\mathcal{Q}}^N$ by truncating this ordered set. We define a general framework for synthetic negative generation where $\mathbf{s}_k^i$ represents the k-th synthetic negative from the i-th strategy. All synthetic negatives are ℓ_2-normalized to ensure consistency with the representation space geometry. Following [24, 35], we implement six distinct transformation strategies:

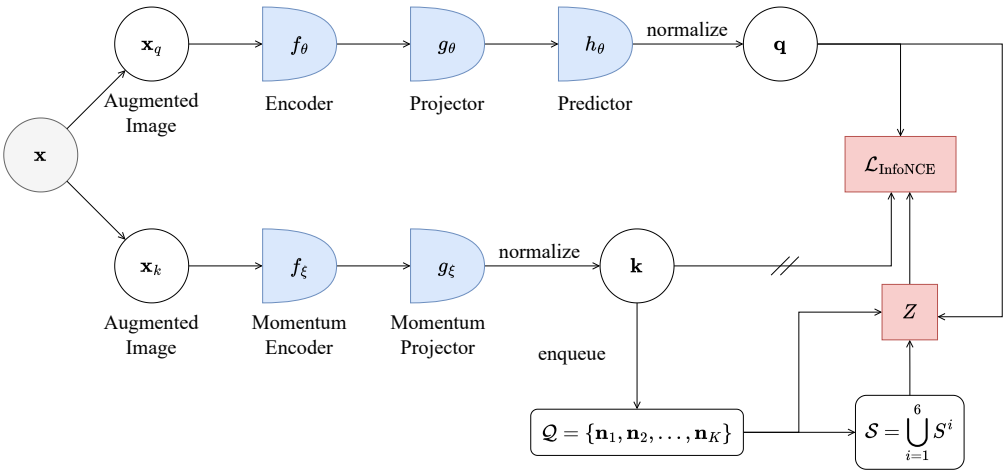


Figure 3. **SynCo-v2 architecture diagram.** SynCo-v2 extends MoBY [65] through the "*on-the-fly*" generation of synthetic hard negatives from the memory queue. Shaded circles ◯ represent observed variables, while empty circles ○ represent computed values, // indicates a stop-gradient for backpropagation, red boxes represent functions.

$$\mathbf{s}_k^i = \begin{cases} \alpha_k \cdot \mathbf{q} + (1-\alpha_k) \cdot \mathbf{n}_j, & i = 1 \\ \mathbf{n}_j + \beta_k \cdot (\mathbf{n}_j - \mathbf{q}), & i = 2 \\ \gamma_k \cdot \mathbf{n}_j + (1-\gamma_k) \cdot \mathbf{n}_l, & i = 3 \\ \mathbf{n}_j + \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I}), & i = 4 \\ \mathbf{n}_j + \delta \cdot \nabla_{\mathbf{n}_j} \text{sim}(\mathbf{q}, \mathbf{n}_j), & i = 5 \\ \mathbf{n}_j + \eta \cdot \text{sign}(\nabla_{\mathbf{n}_j} \text{sim}(\mathbf{q}, \mathbf{n}_j)), & i = 6 \end{cases} \tag{2}$$

where $\mathbf{n}_j, \mathbf{n}_l \in \hat{\mathcal{Q}}^N$ are selected hard negatives, and $\text{sim}(\mathbf{q}, \mathbf{n}_j) = \mathbf{q}^\top \cdot \mathbf{n}_j$ represents the cosine similarity function. **(i)** *Interpolated negatives* ($i = 1$) create synthetic examples between the query and hard negatives, where $\alpha_k \in (0, 0.5)$ controls the balance between query and negative contributions. **(ii)** *Extrapolated negatives* ($i = 2$) explore directions beyond hard negatives, where $\beta_k \in (1, 1.5)$ determines the extrapolation distance. **(iii)** *Mixup negatives* ($i = 3$) combine pairs of hard negatives with mixing coefficient $\gamma_k \in (0, 1)$. **(iv)** *Noise-injected negatives* ($i = 4$) add controlled stochasticity with Gaussian noise ($\sigma = 0.01$). **(v)** *Perturbed negatives* ($i = 5$) modify hard negatives using gradient-based perturbations with $\delta = 0.01$. **(vi)** *Adversarial negatives* ($i = 6$) apply sign-based perturbations with strength $\eta = 0.01$.

The complete set of synthetic hard negatives is formed as $\mathcal{S} = \bigcup_{i=1}^{6} S^i$, where $S^i = \{\mathbf{s}_1^i, \mathbf{s}_2^i, \ldots\}$ contains all $|S^i|$ synthetic negatives generated by the i-th strategy. These synthetic negatives require memory size $\mathcal{O}(|\mathcal{S}| \cdot d)$, where $|\mathcal{S}| = \sum_{i=1}^{6} |S^i| \ll K$. We augment the memory queue's negative samples with synthetically generated hard negatives by calculating the denominator Z that comprises contributions from both memory-based and synthetic negatives:

$$Z = \sum_{\mathbf{n} \in \mathcal{Q}} \exp(\mathbf{q}^\top \cdot \mathbf{n}/\tau) + \sum_{\mathbf{s} \in \mathcal{S}} \exp(\mathbf{q}^\top \cdot \mathbf{s}/\tau) \tag{3}$$

where τ is the temperature parameter. We set $\tau = 0.2$. Finally, we optimize the combined negative set using the InfoNCE loss function:

$$\mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q}, \mathcal{S}) = -\log \frac{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau)}{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau) + Z}. \tag{4}$$

Relation to MoBY. When *no* synthetic hard negatives are generated (*i.e.*, $\mathcal{S} = \emptyset$), our method reduces to the standard InfoNCE loss used by MoBY [65] and MoCo-v3 [16] for vision transformers (Figure 2a):

$$\mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q}) = -\log \frac{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau)}{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau) + \sum_{\mathbf{n} \in \mathcal{Q}} \exp(\mathbf{q}^\top \cdot \mathbf{n}/\tau)}. \tag{5}$$

Relation to BYOL. When we replace the InfoNCE loss with a mean squared error loss between the query $\mathbf{q}$ and key $\mathbf{k}$ representations, our method reduces to DINO [12] (Figure 2b) without "*tricks*" or to BYOL [28] (Figure 2c):

$$\mathcal{L}_{\text{MSE}}(\mathbf{q}, \mathbf{k}) = \frac{1}{2} \|\mathbf{q} - \mathbf{k}\|_2^2 \tag{6}$$

In this case, SynCo-v2 does not need negative samples at all, and uses only the momentum encoder and predictor head to avoid collapse, showing that our method can adapt to different learning approaches.

3.2. Implementation and Evaluation Protocols

Vision Transformer. We adopt DeiT-Small/16 [21, 61] or Swin-Tiny/7 [43] as the backbone f_θ. The projection (g_θ) and prediction (h_θ) heads are two-layer MLPs. Their hidden layers are 4096-dim with ReLU [45], and outputs are 256-dim without ReLU. All MLP layers use BN [33].

Implementation details. We pretrain on ImageNet ILSVRC-2012 [19] and ImageNet-100 [36] without labels. Following MoBY [65], we use AdamW [44] across 4 GPUs with batch size 512, base learning rate 0.03, and weight decay 10^{-4}. Training spans 300 epochs. The target-network EMA parameter m starts at $m_{\text{start}} = 0.99$ and increases linearly to 1. We adopt BYOL augmentations [28]. For synthetic negatives, we select the top $N = 256$ negatives from the memory queue and generate 128 synthetic hard negatives per anchor using six transformation strategies (Section 3.1). Finally, we apply asymmetric drop path rates [32] of 0.2 to the online encoder and 0.0 to the target encoder, as in [65].

Evaluation protocols. We follow standard self-supervised learning evaluation protocols to assess the quality of learned representations [13, 29, 70]. Three primary approaches are used: **(i)** *linear probing evaluation*, where a linear classifier is trained on frozen features while keeping the backbone network fixed; **(ii)** *full fine-tuning*, where all model parameters are updated on downstream tasks; and **(iii)** *k-NN evaluation*, where the model's learned features are used to predict labels using a k-nearest neighbors classifier.

4. Main Results

In this section, we present experimental results validating the effectiveness of SynCo-v2 for vision transformers, with implementation details in Sec. 7 and analytical results with robustness evaluation in Sec. 8 of the *supplementary*.

4.1. Linear Evaluation on ImageNet

We evaluate SynCo-v2 by training a linear classifier on top of the frozen representation following standard protocols [37, 38]. We report top-1, top-5 accuracies, and k-NN ($k = 10$) performance in Tables 1 and 2. On *DeiT-Small*, SynCo-v2 improves over MoBY (*no synthetic negatives*) by **+0.8%** and over BYOL (*no negatives at all*) by **+2.8%**, whereas the k-NN evaluation shows larger gains, with SynCo-v2 achieving **+6.7%** over MoBY and **+8.5%** over BYOL, demonstrating that synthetic hard negatives produce features particularly well-suited for nearest neighbor-based classification. On *Swin-Tiny*, SynCo-v2 gains **+0.7%** over MoBY and **+6.9%** over BYOL in linear evaluation, and achieves **+1.5%** and **+11.3%** improvements in k-NN, respectively.

Method	Epochs	Top-1	Top-5	k-NN
Supervised [61]	300	79.8	–	–
SimCLR [13]†	300	69.0	–	–
MoCo-v2 [15]‡	300	71.4	–	–
BYOL [28]†	300	71.0	–	–
SwAV [11]†	300	67.1	–	–
MoCo-v3 [16]†	300	72.5	–	–
DINO [12]‡	300	72.5	–	67.9
BeiT [4]§	300	15.7	–	–
CAE [17]§	300	51.8	–	–
MoBY [65]	300	72.8	–	–
BYOL [28]* (*repr.*)	300	70.3	91.0	62.5
MoBY [65]** (*repr.*)	300	72.3	88.3	64.3
SynCo-v2 (ours)	300	**73.1**	**91.4**	**71.0**

Table 1. **Linear and k-NN DeiT classification on ImageNet.** Results show top-1 and top-5 accuracy (in %) and k-NN accuracy for $k = 10$. All methods do *not* employ multi-crop augmentation [11] for a *fair* comparison. **Symbols:** † Adapted from [16]; ‡ Adapted from [12]; § Adapted from [17]; * SynCo-v2 without negatives; ** SynCo-v2 without synthetic negatives.

Method	Epochs	Top-1	Top-5	k-NN
Supervised [43]	300	81.3	–	–
SiMIM [66]	100	56.0	–	–
SMoG [49]	400	74.5	–	–
MoBY [65]	300	75.0	–	–
BYOL [28]* (*repr.*)	300	68.5	89.4	58.0
MoBY [65]** (*repr.*)	300	74.7	92.7	67.8
SynCo-v2 (ours)	300	**75.4**	**93.1**	**69.3**

Table 2. **Linear and k-NN Swin classification on ImageNet.**

4.2. Nearest Neighbor Retrieval

We evaluate our representations on image retrieval tasks to assess their effectiveness for matching and similarity search. Following the standard protocol, we consider the revisited [53] Oxford and Paris datasets [51]. We freeze the features and directly apply k-NN for retrieval. We report Mean Average Precision (mAP) for the Medium (M) and Hard (H) protocols. In Sec. 4.3 we compare SynCo-v2 with MoBY and BYOL. We observe that SynCo-v2 demonstrates strong retrieval performance, outperforming both BYOL and MoBY.

4.3. Discovering the Semantic Layout of Scenes

A remarkable property of self-supervised vision transformers, as shown by DINO [12, 48], is their ability to capture semantic scene structure without supervision. We evaluate this property through two complementary analyses: quantitative video segmentation performance (Sec. 4.3) and qualitative visualization of learned attention patterns (Fig. 4).

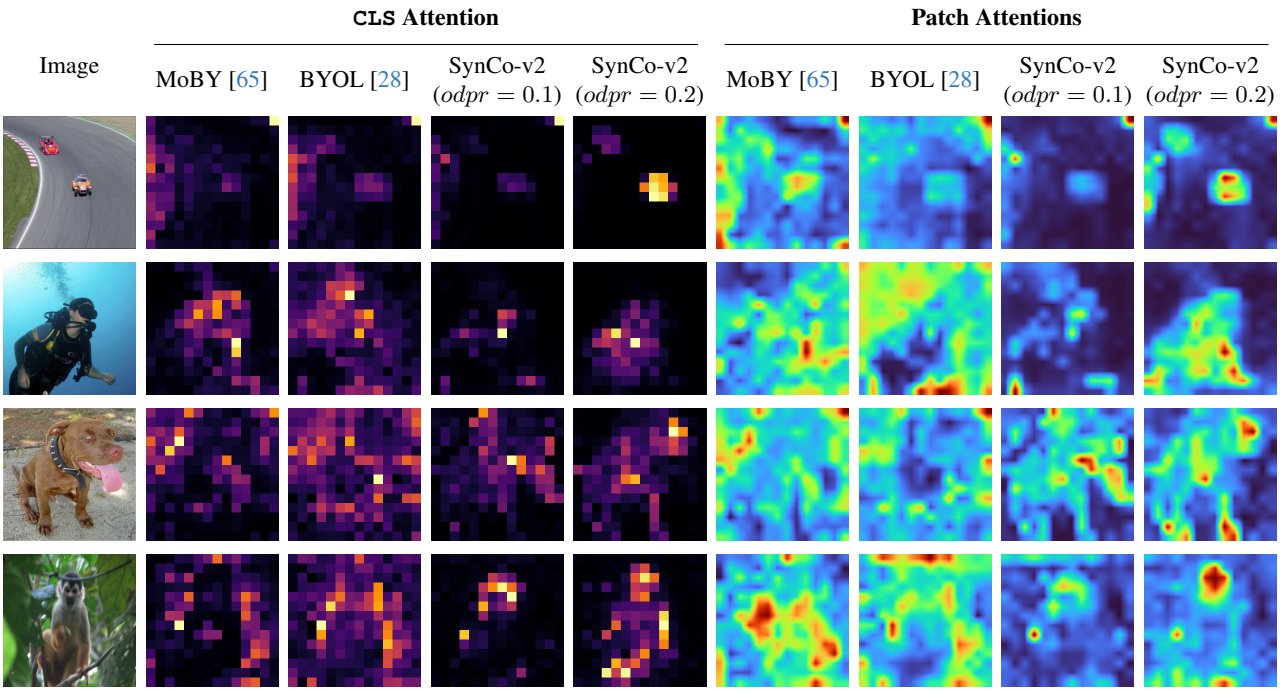


Figure 4. **DeiT-S/16 attention visualization across self-supervised methods.** We show CLS attention (**left group**) and patch attention (**right group**) for MoBY, BYOL, and our proposed SynCo-v2 method for $odpr = 0.1$ and $odpr = 0.2$.

Method	$\mathcal{R}$Ox		$\mathcal{R}$Par	
	M	H	M	H
DeiT-S				
BYOL [28]	23.80	5.40	52.20	20.48
MoBY [65]	32.44	6.83	61.92	25.18
SynCo-v2 (ours)	**39.99**	**11.60**	**66.83**	**29.28**
Swin-T				
BYOL [28]	24.06	4.12	49.73	18.73
MoBY [65]	32.44	7.33	61.46	24.43
SynCo-v2 (ours)	**34.28**	**8.01**	**63.93**	**27.33**

Table 3. **Image retrieval performance.** We report mAP on revisited Oxford ($\mathcal{R}$Ox) and Paris ($\mathcal{R}$Par) datasets.

Method	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
DeiT-S					
BYOL [28] (*repr.*)	41.3	41.5	40.9	41.1	33.6
MoBY [65] (*repr.*)	42.2	42.1	39.6	42.2	34.9
SynCo-v2 (ours)	**44.3**	**44.1**	**41.8**	**44.5**	**38.5**
Swin-T					
BYOL [28] (*repr.*)	34.4	37.9	30.3	31.0	13.7
MoBY [65] (*repr.*)	36.6	39.7	32.7	33.5	16.5
SynCo-v2 (ours)	**37.6**	**40.5**	**32.1**	**34.6**	**17.0**

Table 4. **DAVIS 2017 video object segmentation.** We report mean region similarity $\mathcal{J}_m$, mean contour-based accuracy $\mathcal{F}_m$, and their respective recall metrics $\mathcal{J}_r$ and $\mathcal{F}_r$. Image resolution is 480p.

Video instance segmentation. We evaluate the spatial coherence of learned representations on the DAVIS-2017 video instance segmentation benchmark [52], following [34]. We segment scenes using nearest-neighbor matching between consecutive frames without any training or finetuning, directly evaluating the frozen features. SynCo-v2 outperforms both MoBY and BYOL on this task (Sec. 4.3).

Visualizing attention mechanisms. Recent work [12, 48] demonstrated that vision transformers can segment objects and focus on semantically meaningful regions without explicit supervision. However, it remains unclear whether these properties are exclusive to self-distillation methods or emerge more generally in self-supervised vision transformers. Following [12], we extract attention weights from the last layer and visualize: **(i)** CLS attention, showing attention from the [CLS] token to patches, and **(ii)** patch attentions, showing self-attention patterns that capture object boundaries. As shown in Fig. 1, all methods segment foreground objects from backgrounds. However, SynCo-v2 produces significantly sharper attention maps, capturing fine-grained details like the bear's head and claws (Fig. 1), the horse's body structure (Fig. 1), and the dog's head and feet (Fig. 4), without employing multiple tricks as in DINO [12] (Sec. 2).

Method	Cifar_{10}	Cifar_{100}	Flwrs	Pets	Food_{101}	Cars	Cltch_{101}	DTD	Aircft	SUN_{397}	VOC
					DeiT-S						
BYOL [28] (*repr.*)	90.5	74.2	**87.7**	85.1	73.3	**49.2**	**99.1**	69.2	38.3	77.2	77.0
MoBY [65] (*repr.*)	88.9	73.0	56.8	80.8	69.7	24.2	93.2	62.0	17.4	72.9	74.2
SynCo-v2 (ours)	**92.1**	**79.7**	72.6	**86.1**	**75.0**	37.1	96.5	**70.0**	**39.2**	**78.5**	**78.8**
					Swin-T						
BYOL [28] (*repr.*)	88.6	72.2	83.8	83.0	73.7	34.7	99.0	**72.1**	28.9	77.9	78.5
MoBY [65] (*repr.*)	90.6	76.5	**90.3**	88.2	78.8	60.2	99.3	71.3	**43.6**	84.0	79.9
SynCo-v2 (ours)	**91.4**	**77.7**	89.5	**88.5**	**79.8**	**61.5**	**99.4**	**72.1**	43.3	**84.6**	**80.4**

Table 5. **Linear probing performance on various downstream classification datasets.** Results show top-1 accuracy (in %) with frozen weights except for the final fully-connected layer. See Sec. 7.1 of the *supplementary* for dataset details.

Method	Cifar_{10}	Cifar_{100}	Flwrs	Pets	Food_{101}
			DeiT-S		
BYOL [28] (*repr.*)	86.3	62.4	87.7	85.1	73.5
MoBY [65] (*repr.*)	75.2	80.3	66.0	82.3	71.0
SynCo-v2 (ours)	**96.8**	**83.1**	**88.3**	**87.2**	**85.7**
			Swin-T		
BYOL [28] (*repr.*)	89.2	64.9	83.8	83.0	74.0
MoBY [65] (*repr.*)	97.3	84.8	90.3	88.2	79.8
SynCo-v2 (ours)	**97.6**	**85.8**	**91.2**	**89.5**	**90.3**

Table 6. **End-to-end finetuning performance on various downstream classification tasks.** Results show top-1 accuracy (in %) with all parameters updated during training.

S^1	S^2	S^3	S^4	S^5	S^6	*DeiT-S*	*Swin-T*
✗	✗	✗	✗	✗	✗	69.2	70.9
✓	✗	✗	✗	✗	✗	69.5	71.2
✗	✓	✗	✗	✗	✗	69.4	71.1
✗	✗	✓	✗	✗	✗	69.6	71.3
✗	✗	✗	✓	✗	✗	69.4	71.1
✗	✗	✗	✗	✓	✗	69.3	71.0
✗	✗	✗	✗	✗	✓	69.3	71.0
✓	✓	✓	✓	✓	✓	**70.0**	**71.6**

Table 7. **Ablation study on synthetic negative strategies on ImageNet.** Each strategy generates 128 synthetic negatives. We pretrain for 100 epochs and report top-1 accuracy (%). We highlight the default hyperparameter.

4.4. Transfer Learning on Downstream Tasks

We evaluate whether the features learned on ImageNet with SynCo-v2 are generic and thus useful across image domains, or if they are ImageNet-specific. We perform linear evaluation and fine-tuning on the same set of classification tasks used in [37, 38]. We report results in Tables 5 and 6 for linear evaluation and fine-tuning, respectively. SynCo-v2 outperforms MoBY and BYOL on the majority of benchmarks. In linear probing, our method achieves superior performance on 8 out of 11 datasets with *DeiT-Small* and 9 out of 11 datasets with *Swin-Tiny*. In fine-tuning, SynCo-v2 outperforms both baselines on all 5 evaluated datasets with both architectures. SynCo-v2's representation can be effectively transferred to small images, *e.g.*, CIFAR-10 and CIFAR-100 [40], fine-grained recognition tasks, *e.g.*, Flowers-102 [46] and Stanford Cars [39], landscapes, *e.g.*, SUN397 [64], and textures, *e.g.*, DTD [18].

5. Ablation Study of SynCo-v2

We conduct ablation studies to analyze synthetic negative strategies, regularization techniques, and hyperparameters.

Synthetic hard negatives strategies. We perform ablation studies on combinations of synthetic negative transformation strategies. Table 7 shows that combining all six types (S^1–S^6) yields the highest performance. Without synthetic negatives, the baseline improves by **+0.8%** and **+0.7%** when all strategies are applied. While individual strategies vary in effectiveness (S^3 most impactful, then S^1), their combination provides complementary benefits exceeding the sum of individual contributions, validating that diverse synthetic negatives collectively improve representations.

Drop path regularization. We investigate the effect of asymmetric drop path on *online* and *target* encoders similar to [65]. Table 8 shows that higher drop path rates for the online encoder (0.2) with no dropout for the target encoder (0.0) yields optimal results, improving *DeiT-Small* performance by **+0.6%** and *Swin-Tiny* by **+0.7%** after 300 epochs compared to using 0.1/0.0 rates. This asymmetric configuration outperforms both no regularization and symmetric drop path rates, with the effectiveness likely stemming from encouraging the online encoder to learn more robust representations while maintaining stability in the target encoder.

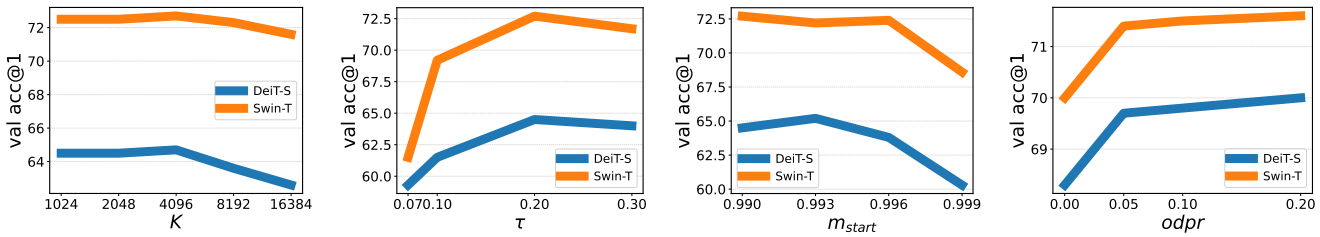


Figure 5. **Ablation studies on hyperparameters on ImageNet-100.** We pretrain for 100 epochs and report top-1 accuracy (%) using DeiT-S and Swin-T architectures. (**left**): queue size K; (**second from left**): temperature τ; (**second from right**): momentum m_{start}; (**right**): *online* drop path rate. **Default hyperparameters**: $K = 4096$, $\tau = 0.2$, $m_{\text{start}} = 0.99$, *online* drop path rate $= 0.2$.

***Online* dpr**	***Target* dpr**	**Epochs**	***DeiT-S***	***Swin-T***
0.0	0.0	100	68.3	70.0
0.1	0.1	100	68.1	69.8
0.05	0.0	100	69.7	71.4
0.1	0.0	100	69.8	71.5
0.2	0.0	100	**70.0**	**71.6**
0.1	0.0	300	72.5	74.7
0.2	0.0	300	**73.1**	**75.4**

Table 8. **Ablation study on the drop path rates on ImageNet.** We pretrain for 100/300 epochs and report top-1 accuracy (%). We highlight the default hyperparameter.

Tricks of MoCo-v3 and SynCo. We evaluate the necessity of various implementation *tricks* from MoCo-v3 [16] and SynCo [24]. Table 9 shows that fixed patch embeddings from MoCo-v3 are unnecessary when using our approach, while SynCo's cooldown strategy (disabling synthetic negatives in later epochs, specifically for the last 100 epochs) achieves **+0.9%** on *DeiT-Small* and **+1.3%** on *Swin-Tiny*. This cooldown approach has proven effective for both convolutional architectures [24] and vision transformers (ours). These findings demonstrate that our method reduces reliance on architecture-specific modifications, making it more broadly applicable across different vision transformer variants while simplifying implementation.

Other hyperparameters. We study the robustness of our approach across different contrastive hyperparameter settings to demonstrate seamless integration with existing frameworks. Figure 5 shows that performance remains stable across a wide range of queue sizes, temperatures, and momentum values using default hyperparameters from MoBY. These findings confirm that synthetic negatives can be readily adopted in existing contrastive learning pipelines *without* requiring architectural modifications, extensive hyperparameter re-tuning, or additional computational overhead during the hyperparameter search process.

MoCo-v3	**SynCo**	***DeiT-S***	***Swin-T***
✓	✗	72.0	73.6
✗	✗	72.2	74.1
✗	✓	**73.1**	**75.4**

Table 9. **Ablation study on applying MoCo-v3 and SynCo *tricks* on ImageNet.** We pretrain for 300 epochs and report top-1 accuracy (%). We highlight the default hyperparameter.

6. Conclusion

In this paper, we demonstrate that synthetic hard negatives significantly improve vision transformer representations in self-supervised contrastive learning. We show that emergent semantic segmentation properties—previously considered exclusive to self-distillation methods like DINO [12]—naturally arise in contrastive learning and are further strengthened through synthetic negative generation. Our method, SynCo-v2, achieves five key improvements over standard contrastive baselines: **(i)** improved ImageNet linear accuracy, **(ii)** strong k-NN performance indicating high-quality features, **(iii)** improved downstream task performance across diverse settings, **(iv)** sharper attention maps with better object boundary alignment, and **(v)** strong video object segmentation despite no video training. These gains are achieved without DINO's complex procedures (centering, sharpening, multi-crop, extended schedules); under identical training regimes and without extra *tricks*, our approach consistently outperforms baselines. We refer readers to Sec. 10 of the *supplementary* for discussion on scope, fair comparison—including DINO—and broader gains, and to Sec. 11 for limitations such as training with larger architectures and computational budget. Overall, our results challenge the prevailing focus on self-distillation and generative approaches, showing that contrastive learning with high-quality negatives remains a simple yet powerful alternative that integrates cleanly with any InfoNCE-based method [62], generalizes across architectures, and incurs minimal overhead. We hope this work encourages renewed interest in the potential of contrastive learning.

Acknowledgments

We acknowledge the computational resources and support provided by the Imperial College Research Computing Service (http://doi.org/10.14469/hpc/2232), which enabled our experiments.

References

[1] Adnan Ali, Jinlong Li, Huanhuan Chen, and Ali Kashif Bashir. From overfitting to robustness: Quantity, quality, and variety oriented negative sample selection in graph contrastive learning, 2024. 3

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture, 2023. 3

[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] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. 1, 2, 5

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022. 2, 3

[6] Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. 2, 3

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. 1

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features, 2019. 2

[10] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data, 2019.

[11] 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. 2, 3, 5

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 1, 2, 3, 4, 5, 6, 8

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 1, 2, 3, 5

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 3

[15] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020. 3, 5

[16] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021. 2, 3, 4, 5, 8

[17] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning, 2023. 5

[18] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 7

[19] 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. 5

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. 1

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is

worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 5

[22] 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. 3

[23] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning, 2021. 3

[24] Nikolaos Giakoumoglou and Tania Stathaki. Synco: Synthetic hard negatives for contrastive visual representation learning, 2025. 2, 3, 8

[25] Nikolaos Giakoumoglou, Tania Stathaki, and Athanasios Gkelias. A review on discriminative self-supervised learning methods in computer vision, 2025. 1

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. 1

[27] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild, 2021. 1

[28] 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. 1, 2, 3, 4, 5, 6, 7

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. 1, 2, 3, 5

[30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 1, 2

[31] 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. 3

[32] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. 5

[33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 5

[34] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *Advances in Neural Information Processing Systems*, pages 19545–19560. Curran Associates, Inc., 2020. 6

[35] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020. 2, 3

[36] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021. 3, 5

[37] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning, 2019. 5, 7

[38] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019. 5, 7

[39] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 7

[40] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 7

[41] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. 1

[42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 1, 5

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[45] Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. pages 807–814, 2010. 5

[46] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 722–729, 2008. 7

[47] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2016. 1

[48] 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. 1, 2, 3, 5, 6

[49] Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu. Unsupervised visual representation learning by synchronous momentum grouping, 2022. 2, 3, 5

[50] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers, 2022. 1

[51] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 5

[52] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018. 6

[53] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking, 2018. 5

[54] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. 1

[55] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[57] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. 3

[58] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. 2, 3

[59] 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. 3

[60] 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. 3

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021. 1, 5

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

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1

[64] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 7

[65] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers, 2021. 2, 3, 4, 5, 6, 7

[66] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022. 5

[67] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022. 3

[68] Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. Learning efficient coding of natural images with maximum manifold capacity representations, 2023. 3

[69] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. 3

[70] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016. 5

[71] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022. 2, 3