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

Nikolaos Giakoumoglou[1*], **Andras Floros**[1*], **Kleanthis Marios Papadopoulos**[1*], **Tania Stathaki**[1]

[1]Imperial College London
[*]Core Team

We introduce SynCo-v2, a method that integrates synthetic hard negatives into unsupervised vision transformer pretraining to improve representation quality. Our approach is thoroughly benchmarked on ImageNet and transfer learning, image retrieval, copy detection, and image, video segmentation tasks. Notably, our proposed negatives give rise to *emergent properties*, where learned representations contain explicit information about the semantic content of an image and serve as excellent classifiers (up to +11.3% over baselines). SynCo-v2 achieves these benefits through simple modifications to existing contrastive frameworks and outperforms competing methods while being more resource efficient, *e.g.*, our ViT-B surpasses V-JEPA with ViT-L. Our findings motivate reconsidering contrastive learning as a simpler yet powerful alternative to dominant generative and self-distillation approaches.

Date: November 27, 2025
Correspondence: Nikolaos Giakoumoglou <nikos@imperial.ac.uk>
Code: https://github.com/giakoumoglou/synco-v2

IMPERIAL

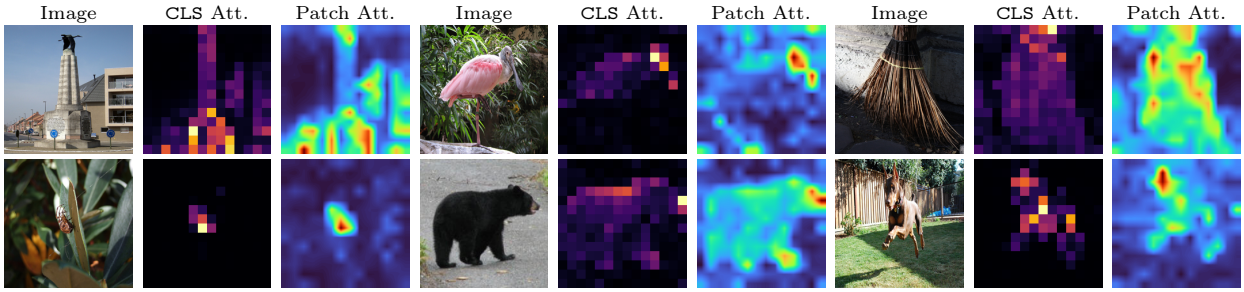
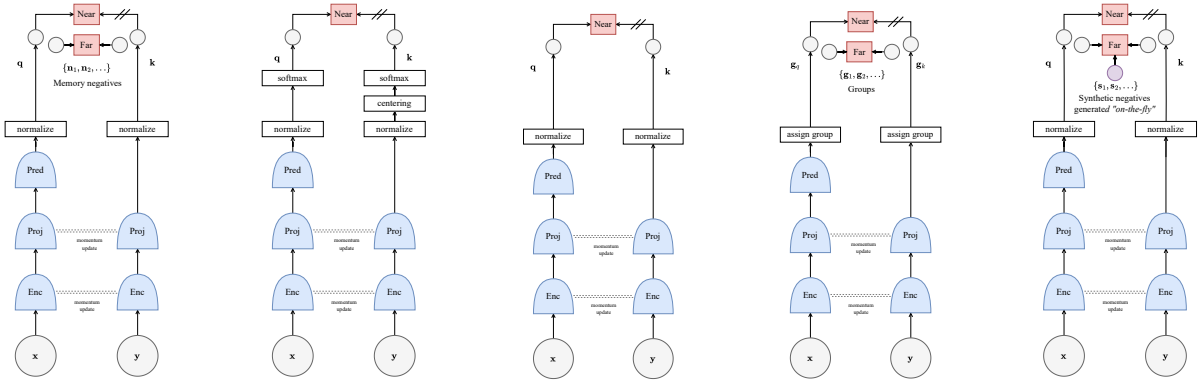


Figure 1 Vision Transformer (ViT-S/16) attention visualization of SynCo-v2. We show the inputs, CLS token attention, and patch attention maps from our method. Our approach with memory and synthetic hard negatives produces focused attention on semantically important regions with clear object boundaries and fine-grained details.

1 Introduction

Transformers (Vaswani et al., 2017) have revolutionized computer vision, establishing themselves as powerful alternatives to ConvNets (Dosovitskiy et al., 2021; Liu et al., 2021; Touvron et al., 2021). 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 (Dosovitskiy et al., 2021; Touvron et al., 2021). 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 (He et al., 2021; Oquab et al., 2023; Goyal et al., 2021).

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 (Bommasani et al., 2021; LeCun et al., 2015; Balestriero et al., 2023; Giakoumoglou et al., 2025). Self-supervised approaches for vision fall into three categories: **(i)** *pretext task* methods that solve auxiliary tasks such as rotation prediction



(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 (Xie et al., 2021a; Chen et al., 2021); (b, c) *self-distillation* (Caron et al., 2021; Grill et al., 2020); (d) *clustering-based* (Pang et al., 2022); 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.

(Gidaris et al., 2018; Noroozi and Favaro, 2016) or jigsaw puzzles (Noroozi and Favaro, 2016); **(ii)** *generative* methods that reconstruct or predict masked portions of inputs, such as MAE (He et al., 2021) inspired by masked language modeling (Radford and Narasimhan, 2018; Devlin et al., 2018; Radford et al., 2019; Brown et al., 2020), and BEiT (Bao et al., 2022) following BERT-like pretraining (Devlin et al., 2018; Lan et al., 2020); and **(iii)** *joint embedding architecture* methods that learn representations by comparing different views of data in a shared embedding space (He et al., 2020; Chen et al., 2020b; Grill et al., 2020; Caron et al., 2021; Radford et al., 2021). This work focuses on training transformers with joint embedding architectures, unlike prior works using generative methods (He et al., 2021; Bao et al., 2022; Peng et al., 2022).

The joint embedding methods adapted for vision transformers fall into three categories shown in Figure 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 (Chen et al., 2020b; He et al., 2020; Chen et al., 2021; Xie et al., 2021a) (Figure 2a); **(ii)** *self-distillation* (*teacher–student*) methods train a student to match a teacher's embeddings without negatives (Caron et al., 2021; Grill et al., 2020; Bardes et al., 2022, 2024b) (Figures 2b and 2c); and **(iii)** *clustering-based* methods employ clustering objectives (Caron et al., 2019a, 2020, 2019b; Pang et al., 2022) (Figure 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 (Caron et al., 2021). Self-distillation methods like DINO (Caron et al., 2021; Oquab et al., 2023; Siméoni et al., 2025) and iBOT (Zhou et al., 2022) demonstrate strong unsupervised semantic segmentation, with attention naturally aligning to object boundaries and meaningful regions without supervision. These emergent behaviors arise from extensive pretraining and architectural designs that promote structured attention patterns.

Despite generative methods achieving higher accuracy (He et al., 2021; Bao et al., 2022), joint embedding approaches remain important, achieving competitive accuracy and often surpassing generative methods in linear probing (Oquab et al., 2023; Bardes et al., 2024b). Among these approaches, contrastive learning methods stand out for their simplicity and computational efficiency (Chen et al., 2021; Xie et al., 2021a),

explicitly using negative samples to define representation boundaries (Chen et al., 2020b; He et al., 2020), yet they have received less attention recently.

We seek to answer the simple question:

> *Can simple modifications to negative sampling 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 (Giakoumoglou and Stathaki, 2025; Kalantidis et al., 2020) 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 extensive empirical evaluation, we demonstrate that integrating synthetic hard negatives into transformer-based self-supervised learning yields three key improvements over training *without* synthetic hard negatives or *without* negatives: **(i)** higher top-1 accuracy on ImageNet linear evaluation (Tables 1 and 2), achieving 73.1% and 77.1% with ViT-S/16 and ViT-B/16, and 75.4% and 78.0% with Swin-T and Swin-S, respectively; **(ii)** improved transfer learning performance across diverse downstream tasks (Tables 6 to 8); and **(iii)** strong emergent properties, where self-supervised vision transformer features encode explicit semantic segmentation information (Tables 3 to 5), producing precise attention maps that capture object boundaries (Figure 4) and serve as effective k-NN classifiers (Tables 1 and 2), achieving 73.3% top-1 accuracy with ViT-B/16.

2 Related Work

2.1 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 (Chen et al., 2020b) or momentum-encoded memory banks (He et al., 2020; Chen et al., 2020d, 2021; Xie et al., 2021a) to provide sufficient negative samples. Alternative approaches formalize collapse avoidance via mutual information (van den Oord et al., 2019; Hjelm et al., 2019; Tian et al., 2020). *Self-distillation* (*a.k.a. teacher-student distillation*) methods surprisingly avoid collapse without negatives. They use asymmetric architectures (Chen and He, 2020a), momentum updates (Grill et al., 2020; Caron et al., 2021), and stop-gradient operations (Grill et al., 2020; Caron et al., 2021; Chen and He, 2020a). Alternatively, they explicitly regularize feature covariance so representations do not collapse, *e.g.*, decorrelate features (Zbontar et al., 2021; Bardes et al., 2022), employ whitening (Ermolov et al., 2021), or manifold regularization (Yerxa et al., 2023). Notably, DINO (Caron et al., 2021; Oquab et al., 2023; Siméoni et al., 2025), which employs multiple techniques including centering, sharpening, momentum encoder, multi-crop training (Caron et al., 2020), and extended training, and iBOT (Zhou et al., 2022), which integrates masked patch prediction, exhibit strong unsupervised segmentation. Unlike contrastive learning or self-distillation methods, I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024b) 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 (Caron et al., 2020) or via momentum grouping (Pang et al., 2022).

2.2 Contrastive Learning

Contrastive learning methods treat instance discrimination as a pretext task, treating each image as its own class (Chen et al., 2020b; He et al., 2020). The core principle involves bringing an anchor and a "*positive*" sample closer in the embedding space while pushing the anchor away from "*negative*" samples (Khosla et al., 2021). Training typically employs InfoNCE loss (van den Oord et al., 2019) or its variants (Chen et al., 2020b; Dwibedi et al., 2021; Tomasev et al., 2022; Yeh et al., 2022), maximizing mutual information between positive

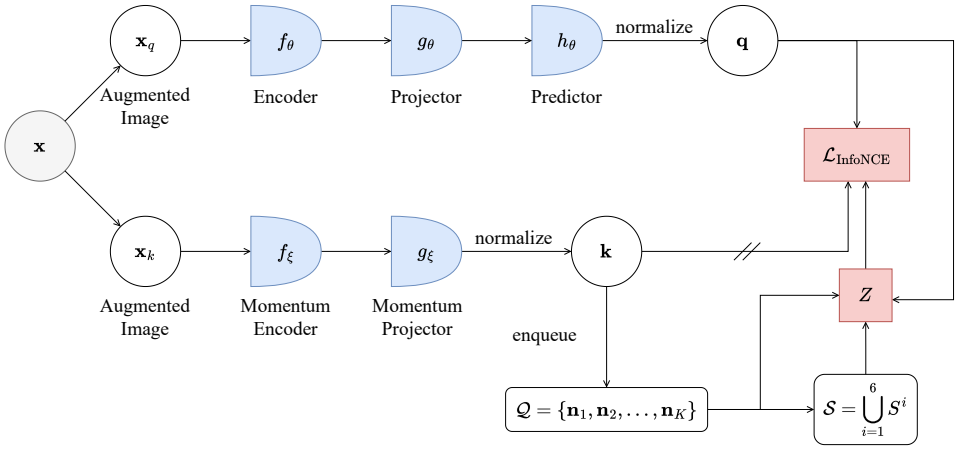


Figure 3 SynCo-v2 architecture diagram. SynCo-v2 extends MoBY (Xie et al., 2021a) through the "*on-the-fly*" generation of synthetic hard negatives from the memory queue. Shaded circles ◯ represent observed variables, while empty circles ◯ represent computed values, // indicates a stop-gradient for backpropagation, red boxes represent functions.

pairs while minimizing it for negatives. Negative samples are drawn from large batch sizes (Chen et al., 2020b) or memory banks (He et al., 2020; Chen et al., 2020d, 2021; Xie et al., 2021a). 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 (Robinson et al., 2021; Ali et al., 2024). Various strategies leverage hard negatives through mixup-based interpolation between embeddings (Kalantidis et al., 2020), debiased contrastive losses with theoretical analysis (Robinson et al., 2021), importance reweighting schemes (Yeh et al., 2022), and hardness-aware sampling from memory queues (Tomasev et al., 2022). Systematic synthetic generation through transformation strategies has proven effective for convolutional networks (Giakoumoglou and Stathaki, 2025). 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 Figures 2e and 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 $\mathbf{q}, \mathbf{k} \in \mathbb{R}^d$ (Xie et al., 2021a; Grill et al., 2020). 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 $\mathcal{Q} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_K\}$ that consists of features from distinct images, serving as K negative samples (He et al., 2020; Chen et al., 2020d, 2021; Xie et al., 2021a). 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: $\xi \leftarrow m \cdot \xi + (1 - m) \cdot \theta$, where

$m \in [0, 1]$ is the momentum coefficient that controls the update rate (Grill et al., 2020; Xie et al., 2021a). This momentum mechanism ensures gradual evolution of f_ξ, improving the stability of negative samples across training iterations (He et al., 2020).

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.*, $\hat{\mathcal{Q}} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_K\}$ 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 (Giakoumoglou and Stathaki, 2025; Kalantidis et al., 2020), we implement six distinct transformation strategies:

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

where $\mathbf{n}_j, \mathbf{n}_l \in \hat{\mathcal{Q}}^N$ are selected hard negatives, and $\mathrm{sim}(\mathbf{q}, \mathbf{n}_j) = \mathbf{q}^\top \cdot \mathbf{n}_j$ 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:

$$Z = \sum_{\mathbf{n} \in \mathcal{Q}} \exp(\mathbf{q}^\top \cdot \mathbf{n}/\tau) + \sum_{\mathbf{s} \in \mathcal{S}} \exp(\mathbf{q}^\top \cdot \mathbf{s}/\tau) \tag{2}$$

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

$$\mathcal{L}(\mathbf{q}, \mathbf{k}, \mathcal{Q}, \mathcal{S}) = -\log \frac{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau)}{\exp(\mathbf{q}^\top \cdot \mathbf{k}/\tau) + Z}. \tag{3}$$

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 (Xie et al., 2021a) and MoCo-v3 (Chen et al., 2021) for vision transformers (Figure 2a):

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

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 (Caron et al., 2021) (Figure 2b) without "*tricks*" or to BYOL (Grill et al., 2020) (Figure 2c):

$$\mathcal{L}_{\text{MSE}}(\mathbf{q}, \mathbf{k}) = \frac{1}{2}\|\mathbf{q} - \mathbf{k}\|_2^2. \tag{5}$$

3.2 Implementation and Evaluation Protocols

Architecture. We adopt ViT-S/16 (22M) and ViT-B/16 (86M) (Dosovitskiy et al., 2021; Touvron et al., 2021) or Swin-T (28M) and Swin-S (50M) (Liu et al., 2021) 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 (Nair and Hinton, 2010), and outputs are 256-dim without ReLU. All MLP layers use BN (Ioffe and Szegedy, 2015).

Implementation details. We pretrain on ImageNet ILSVRC-2012 (Deng et al., 2009) and ImageNet-100 (Khosla et al., 2021) without labels. Following MoBY (Xie et al., 2021a), we use AdamW (Loshchilov and Hutter, 2019) 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 (Grill et al., 2020). 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), totaling 768 synthetic negatives. Finally, we apply asymmetric drop path rates (Huang et al., 2016) of 0.2 to the online encoder and 0.0 to the target encoder, as in (Xie et al., 2021a). The code and models to reproduce our results is publicly available.

Evaluation protocols. We follow standard self-supervised learning evaluation protocols to assess the quality of learned representations (Zhang et al., 2016; He et al., 2020; Chen et al., 2020b). 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 Section A and more results in Section B.

4.1 Linear Evaluation on ImageNet

We evaluate SynCo-v2 by training a linear classifier on top of the frozen representation following standard protocols (Kornblith et al., 2019; Kolesnikov et al., 2019). We report top-1, top-5 accuracies, and k-NN ($k = 10$) performance in Tables 1 and 2. Notably, our ViT-S model surpasses I-JEPA ViT-B in linear evaluation, while our ViT-B outperforms I-JEPA, V-JEPA, and iBOT, even when these baselines employ larger models. Furthermore, SynCo-v2 surpasses MoBY (no synthetic negatives) and BYOL (no negatives at all) by a large margin across both linear and k-NN evaluations.

4.2 Nearest Neighbor Retrieval

We further evaluate our representations on landmark retrieval and copy detection tasks to assess their effectiveness for matching and similarity search.

Image retrieval. Following (Caron et al., 2021), we consider the revisited (Radenović et al., 2018) Oxford and Paris datasets (Philbin et al., 2008). We freeze the features and directly apply k-NN for retrieval. As shown in Table 3, SynCo-v2 demonstrates competitive retrieval performance, with our smaller models achieving results comparable to or exceeding those of larger architectures.

Table 1 Linear and k-NN ViT classification on ImageNet. Results show top-1 and top-5 accuracy and k-NN accuracy for models trained without multi-crop augmentation. **Symbols:** † adapted from (Chen et al., 2021); ‡ from (Caron et al., 2021); ◇ from (Zhou et al., 2022); § from (Chen et al., 2023a).

Method	Arch.	Ep.	Top-1	Top-5	k-NN
Supervised (Touvron et al., 2021)	ViT-S	300	79.8	–	–
Supervised (Touvron et al., 2021)	ViT-B	300	81.8	–	–
BYOL (Grill et al., 2020)†	ViT-S	300	71.0	–	–
SwAV (Caron et al., 2020)†	ViT-S	300	67.1	–	–
BYOL (Grill et al., 2020) (repr.)	ViT-S	300	70.3	91.0	62.5
MoBY (Xie et al., 2021a) (repr.)	ViT-S	300	72.3	88.3	64.3
MoBY (Xie et al., 2021a)	ViT-S	300	72.8	–	–
MoCo-v3 (Chen et al., 2021)†	ViT-S	300	72.5	–	–
MoCo-v3 (Chen et al., 2021)◇	ViT-B	300	76.7	–	–
DINO (Caron et al., 2021)‡	ViT-S	300	72.5	–	67.9
DINO (Caron et al., 2021)◇	ViT-B	200	76.0	–	71.2
iBOT (Zhou et al., 2022)	ViT-B	200	76.0	–	71.2
MAE (He et al., 2021)	ViT-B	1600	68.0	–	–
SimMIM (Xie et al., 2022)	ViT-B	800	56.7	–	–
BeiT (Bao et al., 2022)§	ViT-S	300	15.7	–	–
CAE (Chen et al., 2023a)§	ViT-S	300	51.8	–	–
CAE (Chen et al., 2023a)	ViT-B	1600	70.4	–	–
I-JEPA (Assran et al., 2023)	ViT-B	600	72.9	–	–
V-JEPA (Bardes et al., 2024b)	ViT-L	600	73.7	–	–
SynCo-v2 (ours)	ViT-S	300	73.1	91.4	71.0
SynCo-v2 (ours)	ViT-B	300	**77.1**	**94.4**	**73.3**

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

Method	Arch	Ep.	Top-1	Top-5	k-NN
Supervised (Liu et al., 2021)	Swin-T	300	81.3	–	–
Supervised (Liu et al., 2021)	Swin-S	300	83.0	–	–
BYOL (Grill et al., 2020) (repr.)	Swin-T	300	68.5	89.4	58.0
MoBY (Xie et al., 2021a)	Swin-T	300	75.0	–	–
MoBY (Xie et al., 2021a) (repr.)	Swin-T	300	74.7	92.7	67.8
SiMIM (Xie et al., 2022)	Swin-T	100	56.0	–	–
SMoG (Pang et al., 2022)	Swin-T	400	74.5	–	–
SynCo-v2 (ours)	Swin-T	300	75.4	93.1	69.3
SynCo-v2 (ours)	Swin-S	300	**78.0**	**95.6**	**71.9**

Copy detection. We evaluate performance on copy detection following (Caron et al., 2021) protocol, reporting mean average precision on the "strong" subset of the Copydays dataset (Douze et al., 2009). As shown in Table 4, SynCo-v2 surpass DINO (Caron et al., 2021) on this task.

4.3 Discovering the Semantic Layout of Scenes

A remarkable property of self-supervised vision transformers, as shown by DINO (Caron et al., 2021; Oquab et al., 2023), is their ability to capture semantic scene structure without supervision. We evaluate this property through two complementary analyses: quantitative video segmentation performance (Table 5) and qualitative visualization of learned attention patterns (Figure 4).

Table 3 Image retrieval performance. We report mAP on revisited Oxford ($\mathcal{R}$Ox) and Paris ($\mathcal{R}$Par) datasets. **Symbols**: $^{\diamond}$ trained for more epochs with multi-crop augmentation.

Method	Arch.	$\mathcal{R}$Ox		$\mathcal{R}$Par	
		M	H	M	H
Supervised	ViT-S	33.5	8.9	63.0	37.2
BYOL (Grill et al., 2020) (repr.)	ViT-S	23.8	5.4	52.2	20.5
BYOL (Grill et al., 2020) (repr.)	Swin-T	24.1	4.1	49.7	18.7
MoBY (Xie et al., 2021a) (repr.)	ViT-S	32.4	6.8	61.9	25.2
MoBY (Xie et al., 2021a) (repr.)	Swin-T	32.4	7.3	61.5	24.4
DINO (Caron et al., 2021)$^{\diamond}$	ViT-S	37.2	**13.7**	63.1	**34.4**
iBOT (Zhou et al., 2022)$^{\diamond}$	ViT-B	36.6	13.0	61.5	34.1
SynCo-v2 (ours)	ViT-S	**40.0**	12.6	**66.8**	31.3
SynCo-v2 (ours)	Swin-T	35.3	9.3	64.1	29.3

Table 4 Copy detection. We report the mAP performance in copy detection on Copydays "strong" subset (Douze et al., 2009). All models use resolution 224^2 and 1536 dimensions.

Method	Arch.	mAP
Supervised (Touvron et al., 2021)	ViT-B	76.4
DINO	ViT-B	81.7
SynCo-v2 (ours)	ViT-S	79.7
SynCo-v2 (ours)	ViT-B	**82.0**

Table 5 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.

Method	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
BYOL (Grill et al., 2020) (repr.)	ViT-S	41.3	41.5	40.9	41.1	33.6
BYOL (Grill et al., 2020) (repr.)	Swin-T	34.4	37.9	30.3	31.0	13.7
MoBY (Xie et al., 2021a) (repr.)	ViT-S	42.2	42.1	39.6	42.2	34.9
MoBY (Xie et al., 2021a) (repr.)	Swin-T	36.6	39.7	32.7	33.5	16.5
DINO (Caron et al., 2021)$^{\diamond}$	ViT-S	61.8	60.2	–	63.4	–
DINO (Caron et al., 2021)$^{\diamond}$	ViT-B	**62.3**	**60.7**	–	**63.9**	–
iBOT (Zhou et al., 2022)$^{\diamond}$	ViT-B	61.8	60.4	–	63.2	–
SynCo-v2 (ours)	ViT-S	44.3	44.1	41.8	44.5	38.5
SynCo-v2 (ours)	Swin-T	37.6	40.5	32.1	34.6	17.0

Video instance segmentation. We evaluate the spatial coherence of learned representations on the DAVIS-2017 video instance segmentation benchmark (Pont-Tuset et al., 2018), following (Jabri et al., 2020). We segment scenes using nearest-neighbor matching between consecutive frames without any training or finetuning, directly evaluating the frozen features. SynCo-v2 achieves strong performance on this task (Table 5).

Visualizing attention mechanisms. Recent work (Caron et al., 2021; Oquab et al., 2023) 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 (Caron et al., 2021), 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 Figure 1, all methods segment foreground objects from backgrounds. However, SynCo-v2 produces

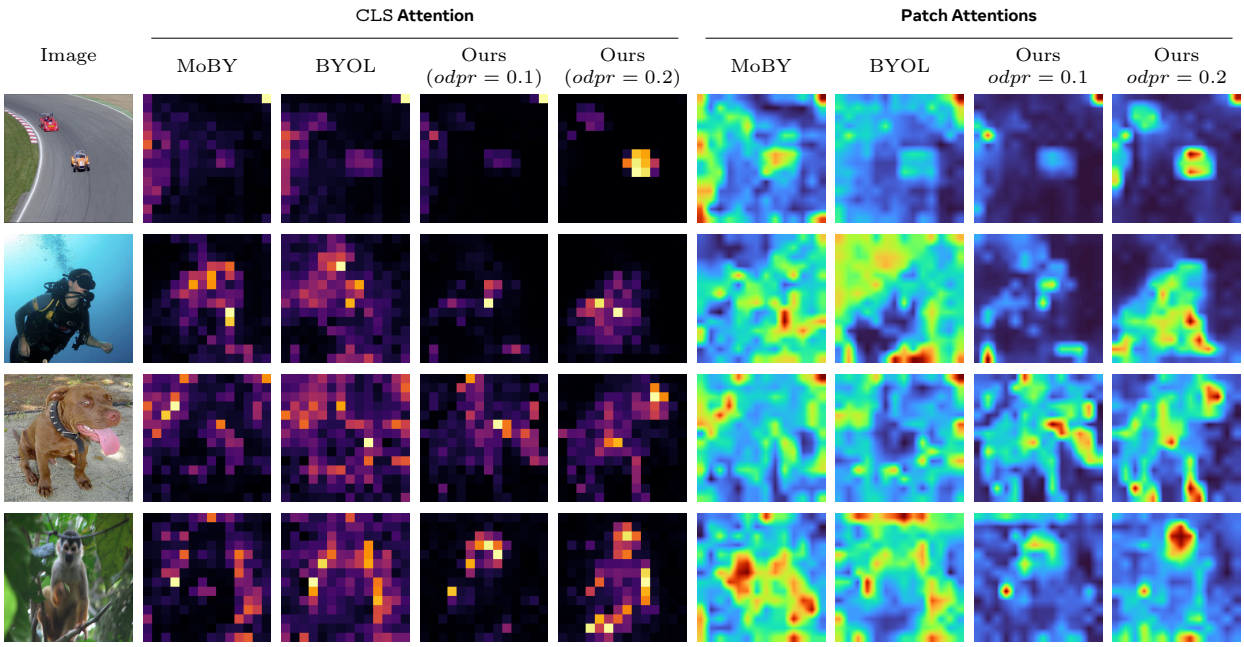


Figure 4 ViT-S/16 attention visualization across self-supervised methods. We show CLS attention (**left group**) and patch attention (**right group**) for MoBY, BYOL, and our proposed SynCo-v2 method for $odpr = 0.1$ and $odpr = 0.2$.

Table 6 Transfer learning performance on dense prediction tasks. Object detection (mAP^{bb}) and instance segmentation (mAP^{msk}) are evaluated on COCO while semantic segmentation performance is reported as mIoU on ADE20K.

Method	Arch.	$\mathbf{mAP^{bb}}$	$\mathbf{mAP^{msk}}$	mIoU
Supervised (Touvron et al., 2021)	ViT-S	46.2	40.1	44.5
Supervised (Liu et al., 2021)	Swin-T	48.1	41.7	44.5
iBOT (Zhou et al., 2022)	ViT-S	49.4	42.6	45.4
MoBY (Xie et al., 2021a)	Swin-T	48.1	41.5	44.1
SynCo-v2 (ours)	ViT-S	**49.9**	**42.8**	**46.0**
SynCo-v2 (ours)	Swin-T	48.9	42.2	45.2

significantly sharper attention maps, capturing fine-grained details like the bear's head and claws (Figure 1), the horse's body structure (Figure 1), and the dog's head and feet (Figure 4), without employing multiple tricks as in DINO (Caron et al., 2021) (Section 2).

4.4 Transfer Learning on Downstream Tasks

We evaluate SynCo-v2 transfer learning on dense prediction and classification tasks.

Detection and segmentation on COCO. Object detection and instance segmentation require jointly localizing objects and recognizing their categories. We evaluate on the COCO dataset (Lin et al., 2015) using Cascade Mask R-CNN (Cai and Vasconcelos, 2019), which simultaneously predicts bounding boxes and instance masks. As reported in Table 6, our method consistently improves over strong self-supervised baselines across both architectures.

Semantic segmentation on ADE20K. We further evaluate representation quality on semantic segmentation, a dense pixel-level classification task, using ADE20K (Zhou et al., 2018) with UPerNet (Xiao et al., 2018). As shown in Table 6, our method achieves the best mIoU across both transformer architectures.

Table 7 Linear probing performance on various downstream classification datasets. Results show top-1 accuracy (in %) with frozen weights except for the final fully-connected layer.

Method	Cifar_{10}	Cifar_{100}	Flowers	Pets	Food_{101}
	ViT-S				
BYOL (Grill et al., 2020) (repr.)	90.5	74.2	**87.7**	85.1	73.3
MoBY (Xie et al., 2021a) (repr.)	88.9	73.0	56.8	80.8	69.7
SynCo-v2 (ours)	**92.1**	**79.7**	72.6	**86.1**	**75.0**
	Swin-T				
BYOL (Grill et al., 2020) (repr.)	88.6	72.2	83.8	83.0	73.7
MoBY (Xie et al., 2021a) (repr.)	90.6	76.5	**90.3**	88.2	78.8
SynCo-v2 (ours)	**91.4**	**77.7**	89.5	**88.5**	**79.8**

Table 8 End-to-end finetuning performance on various downstream classification tasks. Results show top-1 accuracy (in %) with all parameters updated during training.

Method	Cifar_{10}	Cifar_{100}	Flowers	Pets	Food_{101}
	ViT-S				
BYOL (Grill et al., 2020) (repr.)	86.3	62.4	**87.7**	85.1	73.5
MoBY (Xie et al., 2021a) (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 (Grill et al., 2020) (repr.)	89.2	64.9	83.8	83.0	74.0
MoBY (Xie et al., 2021a) (repr.)	97.3	84.8	**90.3**	88.2	79.8
SynCo-v2 (ours)	**97.6**	**85.8**	91.2	**89.5**	**90.3**

Transfer learning. 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 (Kolesnikov et al., 2019; Kornblith et al., 2019). We report results in Tables 7 and 8 for linear evaluation and fine-tuning, respectively. In linear probing, our method achieves superior performance on 8 out of 11 datasets with ViT-Small and 9 out of 11 datasets with Swin-Tiny (*cf.* Table 18). In fine-tuning, SynCo-v2 outperforms both baselines on all 5 evaluated datasets with both architectures. SynCo-v2's representations can be effectively transferred to small images, *e.g.*, CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), fine-grained recognition tasks, *e.g.*, Flowers-102 (Nilsback and Zisserman, 2008) and Cars (Krause et al., 2013), landscapes, *e.g.*, SUN397 (Xiao et al., 2010), and textures, *e.g.*, DTD (Cimpoi et al., 2014). See Section B.4 for results on more datasets.

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 9 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 individual contributions, validating that diverse synthetic negatives improve representations.

Drop path regularization. We investigate the effect of asymmetric drop path on *online* and *target* encoders. Table 10 shows that higher drop path rates for the online encoder with no dropout for the target encoder yields optimal results, improving ViT-S performance by **+0.6%** and Swin-T by **+0.7%** after 300 epochs.

Table 9 ImageNet ablation study on synthetic negative strategies. Each strategy generates 128 synthetic negatives. We pretrain for 100 epochs and report top-1 accuracy (%). We highlight the default hyperparameter.

S^1	S^2	S^3	S^4	S^5	S^6	ViT-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 10 Ablation study on the drop path rates on ImageNet. We pretrain and report top-1 accuracy (%). We highlight the default hyperparameter.

Online dpr	Target dpr	Epochs	ViT-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 11 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.

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

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. Table 10 also shows how our method benefits from extended pretraining.

Tricks of MoCo-v3 and SynCo. We evaluate the necessity of implementation *tricks* from MoCo-v3 (Chen et al., 2021) and SynCo (Giakoumoglou and Stathaki, 2025). Table 11 shows that fixed patch embeddings from MoCo-v3 are unnecessary when using our approach, while SynCo's cooldown strategy (disabling synthetic negatives for the last 100 epochs) achieves **+0.9%** on ViT-S and **+1.3%** on Swin-T. This cooldown approach has proven effective for both convnets (Giakoumoglou and Stathaki, 2025) 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.

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

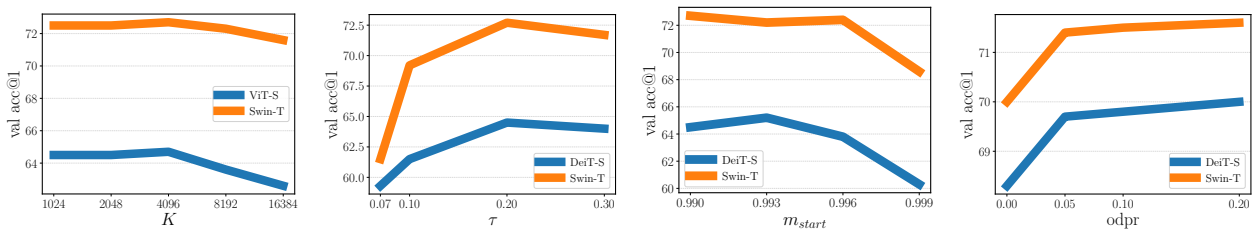


Figure 5 Ablation studies of SynCo-v2 on ImageNet-100. We pretrain for 100 epochs and report top-1 accuracy (%) using ViT-S and Swin-T architectures. (**left**): queue size K; (**second from left**): temperature τ; (**second from right**): momentum m_{start}; (**right**): *online* drop path rate (ImageNet here, see table 10). **Default hyperparameters**: $K = 4096$, $\tau = 0.2$, $m_{\text{start}} = 0.99$, *online* drop path rate $= 0.2$.

re-tuning, or additional computational overhead during the hyperparameter search process.

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 (Caron et al., 2021)—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 Section E for discussion on scope, fair comparison—including DINO—and broader gains. 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 (van den Oord et al., 2019), 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

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. https://arxiv.org/abs/2406.15044.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *European Conference on Computer Vision*, pages 484–501, 2020.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning, 2022.

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. https://arxiv.org/abs/2301.08243.

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.

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

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video, 2024a. https://arxiv.org/abs/2404.08471.

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*, 2024b.

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. https://crfm.stanford.edu/assets/report.pdf.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461. Springer International Publishing, 2014. doi: 10.1007/978-3-319-10599-4_29. https://doi.org/10.1007/978-3-319-10599-4_29.

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.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation, 2019. https://arxiv.org/abs/1906.09756.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

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

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

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

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

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference*

on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020a. https://proceedings.mlr.press/v119/chen20s.html.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020b.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners, 2020c.

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, 2023a.

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, 2023b. https://arxiv.org/abs/2202.03026.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020b. https://arxiv.org/abs/2011.10566.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020d.

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

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. doi: 10.1109/CVPR.2014.461.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020.

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. https://api.semanticscholar.org/CorpusID:57246310.

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

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

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.

Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584805. doi: 10.1145/1646396.1646421. https://doi.org/10.1145/1646396.1646421.

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.

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

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, September 2009. doi: 10.1007/s11263-009-0275-4. https://doi.org/10.1007/s11263-009-0275-4.

Alexandre Eymaël, Renaud Vandeghen, Anthony Cioppa, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Efficient image pre-training with siamese cropped masked autoencoders, 2024. https://arxiv.org/abs/2403.17823.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1): 59–70, 2007. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2005.09.012. https://www.sciencedirect.com/science/article/pii/S1077314206001688. Special issue on Generative Model Based Vision.

P F Felzenszwalb, R B Girshick, D McAllester, and D Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, September 2010. ISSN 0162-8828. doi: 10.1109/tpami.2009.167. http://dx.doi.org/10.1109/TPAMI.2009.167.

Nikolaos Giakoumoglou and Tania Stathaki. Synco: Synthetic hard negatives for contrastive visual representation learning, 2025. https://arxiv.org/abs/2410.02401.

Nikolaos Giakoumoglou, Tania Stathaki, and Athanasios Gkelias. A review on discriminative self-supervised learning methods in computer vision, 2025. https://arxiv.org/abs/2405.04969.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014.

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.

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.

Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders, 2023. https://arxiv.org/abs/2305.14344.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. https://arxiv.org/abs/2111.06377.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b.

Carlos Hinojosa, Shuming Liu, and Bernard Ghanem. Colormae: Exploring data-independent masking strategies in masked autoencoders, 2024. https://arxiv.org/abs/2407.13036.

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.

Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples, 2020. https://arxiv.org/abs/2010.12050.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. https://arxiv.org/abs/1603.09382.

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

Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19545–19560. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper_files/paper/2020/file/e2ef524fbf3d9fe611d5a8e90fefdc9c-Paper.pdf.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. https://arxiv.org/abs/2102.05918.

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

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

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

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

Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning, 2021.

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

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. doi: 10.1109/ICCVW.2013.77.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

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.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. http://dx.doi.org/10.1038/nature14539.

Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others, 2023. https://arxiv.org/abs/2212.04825.

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.

Jihao Liu, Xin Huang, Jinliang Zheng, Yu Liu, and Hongsheng Li. Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers, 2023a.

Yuan Liu, Songyang Zhang, Jiacheng Chen, Kai Chen, and Dahua Lin. Pixmim: Rethinking pixel reconstruction in masked image modeling, 2023b.

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.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution, 2022. https://arxiv.org/abs/2111.09883.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013.

Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. volume 27, pages 807–814, 06 2010.

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. doi: 10.1109/ICVGIP.2008.47.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2016. https://arxiv.org/abs/1603.09246.

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.

Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu. Unsupervised visual representation learning by synchronous momentum grouping, 2022.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

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

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. doi: 10.1109/CVPR.2008.4587635.

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. https://arxiv.org/abs/1704.00675.

Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking, 2018. https://arxiv.org/abs/1803.11285.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. https://api.semanticscholar.org/CorpusID:49313245.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. https://api.semanticscholar.org/CorpusID:160025533.

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. https://arxiv.org/abs/2103.00020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019.

Sucheng Ren, Fangyun Wei, Zheng Zhang, and Han Hu. Tinymim: An empirical study of distilling mim pre-trained models, 2023.

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

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823. IEEE, June 2015. doi: 10.1109/cvpr.2015.7298682. http://dx.doi.org/10.1109/CVPR.2015.7298682.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. http://dx.doi.org/10.1007/s11263-019-01228-7.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining, 2016. https://arxiv.org/abs/1604.03540.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels, 2021. https://arxiv.org/abs/2109.14279.

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. https://arxiv.org/abs/2508.10104.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*,

volume 29. Curran Associates, Inc., 2016. https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. volume 23, pages 828–841. IEEE, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6827–6839. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper_files/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

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.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021. https://arxiv.org/abs/2012.12877.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

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.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, 2019.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021. https://arxiv.org/abs/2102.12122.

Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M. Alvarez. Freesolo: Learning to segment objects without annotations, 2022. https://arxiv.org/abs/2202.12181.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut, 2023. https://arxiv.org/abs/2209.00383.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training, 2023.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021. https://arxiv.org/abs/2103.15808.

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

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. doi: 10.1109/CVPR.2010.5539970.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding, 2018. https://arxiv.org/abs/1807.10221.

Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers, 2021a.

Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning, 2021b. https://arxiv.org/abs/2011.10043.

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.

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

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

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

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018.

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

Appendix

A Implementation Details

We implement SynCo-v2 in PyTorch following the implementation of MoBY (Xie et al., 2021a). We re-implement **(i)** MoBY (Xie et al., 2021a) (contrastive method with memory negatives), and **(ii)** BYOL (Grill et al., 2020) (self-distillation method without negatives). All the three previous methods share the exact same architecture of the backbone f_θ, projector g_θ, and predictor h_θ.

A.1 Architecture

For the ViT architectures (Touvron et al., 2021), we use a patch size of 16×16 (`/16`). The input image is divided into non-overlapping patches that are passed through a linear layer to form embeddings. An extra learnable class token `[CLS]` is added to the sequence to aggregate information from all patches (Lan et al., 2020; Dosovitskiy et al., 2021), even though it is not attached to any label or supervision in our self-supervised setting. Refer to Vaswani et al. (2017) for details about Transformers and to Dosovitskiy et al. (2021) for its adaptation to images. ViT-Small consists of 12 transformer blocks with an embedding dimension of 384 and 6 attention heads, resulting in 22M parameters (denoted as ViT-S`/16` or ViT-S). ViT-Base extends this with 12 transformer blocks, an embedding dimension of 768 and 12 attention heads, resulting in 86M parameters (denoted as ViT-B`/16` or ViT-B). For the Swin architectures (Liu et al., 2021), we use a hierarchical design with shifted windows of size 7×7. Swin-Tiny features an embedding dimension of 96 that expands through the network stages (96, 192, 384, 768), with 2, 2, 6, and 2 blocks across the four stages respectively, resulting in 28M parameters (denoted as Swin-T). Swin-Small uses a similar hierarchical structure with an initial embedding dimension of 96 expanding through stages (96, 192, 384, 768), but with 2, 2, 18, and 2 blocks across the four stages, resulting in 50M parameters (denoted as Swin-S). Detailed specifications are provided in Table 12. The projection head g_θ consists of a 2-layer MLP with a hidden dimension of 4096, batch normalization (Ioffe and Szegedy, 2015), and ReLU activation (Nair and Hinton, 2010), followed by an output layer of dimension 256 without activation. The prediction head h_θ follows the same architecture. Both MLPs use ℓ_2-normalization on their outputs to ensure all representations lie on the unit hypersphere, which is standard practice in contrastive learning (Chen et al., 2020b; He et al., 2020; Grill et al., 2020). The projection head g_θ is attached to the output of the `[CLS]` token for ViT architectures.

Table 12 Transformer architectural specifications. We report the number of transformer blocks, embedding dimension, number of attention heads, total parameters, and throughput (images/sec) for the architectures used in this work. For Swin architectures, values in brackets represent the configuration across the four hierarchical stages.

	ViT-S`/16`	ViT-B`/16`	Swin-T	Swin-S
Blocks	12	12	[2,2,6,2]	[2,2,18,2]
Dim	384	768	[96,192,384,768]	[96,192,384,768]
Heads	6	12	[3,6,12,24]	[3,6,12,24]
Params	22M	86M	28M	50M
Throughput	940	292	755	436

A.2 Image Augmentations

We use the same augmentation as used in BYOL (Grill et al., 2020). We transform each input image with two sampled augmentations to produce two distorted versions of the input. The augmentation pipeline consists of random cropping, resizing to 224×224, randomly flipping the images horizontally, applying color distortion, optionally converting to grayscale, adding Gaussian blurring, and applying solarization. The detailed parameter settings for these augmentations are provided in Table 13.

Table 13 Parameters used to generate image augmentations. This table lists the augmentation probabilities and intensity settings used for pretraining.

Parameter	$\mathcal{T}_q$	$\mathcal{T}_k$
Random crop probability	1.0	1.0
Flip probability	0.5	0.5
Color jittering probability	0.8	0.8
Brightness adjustment max intensity	0.4	0.4
Contrast adjustment max intensity	0.4	0.4
Saturation adjustment max intensity	0.2	0.2
Hue adjustment max intensity	0.1	0.1
Color dropping probability	0.2	0.2
Gaussian blurring probability	1.0	0.1
Solarization probability	0.0	0.2

A.3 Datasets

Our experimental evaluation spans multiple visual recognition benchmarks with varying complexity and domain characteristics. The primary pretraining dataset is ImageNet ILSVRC-2012[1] (Deng et al., 2009), containing 1.28M training images across 1,000 object categories, while ImageNet-100 (Khosla et al., 2021) (a 100-class subset with approximately 128K training images[2]) is used for ablation studies. Transfer learning assessment encompasses eleven downstream classification tasks spanning different visual domains: CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) contain natural images at low resolution (32×32); Oxford Flowers-102 (Nilsback and Zisserman, 2008) features fine-grained flower species; Oxford-IIIT Pets (Parkhi et al., 2012) includes cat and dog breeds; Food-101 (Bossard et al., 2014) contains food dishes; Stanford Cars (Krause et al., 2013) includes fine-grained car models; Caltech-101 (Fei-Fei et al., 2007) covers diverse object categories; DTD (Cimpoi et al., 2014) focuses on texture recognition; FGVC Aircraft (Maji et al., 2013) contains fine-grained aircraft variants; SUN397 (Xiao et al., 2010) features scene recognition; and PASCAL VOC 2007 (Everingham et al., 2009) includes multi-label object classification. For dense prediction tasks, we evaluate on COCO (Lin et al., 2015) for object detection and instance segmentation, containing 118K training images with 80 object categories and pixel-level annotations, and ADE20K (Zhou et al., 2018) for semantic segmentation, featuring 20K training images with 150 semantic categories covering diverse indoor and outdoor scenes. Additionally, we assess representation quality on image retrieval using the revisited Oxford and Paris datasets (Radenović et al., 2018) for landmark retrieval, and Copydays (Douze et al., 2009) for copy detection, which contains 157 original images with various transformations to evaluate robustness to image manipulations. Detailed statistics are provided in Table 14.

A.4 Pretraining

We pretrain all models on ImageNet ILSVRC-2012 (Deng et al., 2009) for 300 epochs. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a base learning rate of 1.5×10^{-4} and a weight decay of 0.1. The training schedule begins with a warmup period during the first 40 epochs in which the learning rate linearly increases from zero to the base learning rate. Following this, the learning rate gradually decreases to zero following a cosine decay schedule without restarts. For the target network, the exponential moving average parameter m starts from $m_{\text{start}} = 0.99$ and is increased to one during training. Specifically, we set $m \triangleq 1 - (1 - m_{\text{start}}) \cdot \left(\cos\left(\frac{\pi k}{K}\right) + 1\right)/2$, with k the current training step and K the maximum number of training steps. We use a total batch size of 512, split across 4 NVIDIA L40S GPUs (128 per GPU) for the smallest models and scaled to 8 NVIDIA L40S GPUs (64 per GPU) for the largest models. We apply asymmetric drop path rates (Huang et al., 2016) of 0.2 for the *online* encoder and 0.0 for the *target* encoder.

[1] Available at: https://www.image-net.org/.
[2] The list of classes from ImageNet-100 is available at: https://github.com/HobbitLong/CMC/blob/master/imagenet100.txt

Table 14 Dataset characteristics. This table lists the datasets used for pretraining and/or evaluation.

Dataset	Classes	Train	Test
ImageNet (Deng et al., 2009)	1000	1,281,167	50,000
ImageNet-100 (Khosla et al., 2021)	100	128,116	5,000
CIFAR-10 (Krizhevsky, 2009)	10	50,000	10,000
CIFAR-100 (Krizhevsky, 2009)	100	50,000	10,000
Flowers-102 (Nilsback and Zisserman, 2008)	102	2,040	6,149
Oxford-IIIT Pets (Parkhi et al., 2012)	37	3,680	3,669
Food-101 (Bossard et al., 2014)	101	75,750	25,250
Stanford Cars (Krause et al., 2013)	196	8,144	8,041
Caltech-101 (Fei-Fei et al., 2007)	101	3,060	6,085
DTD (Cimpoi et al., 2014)	47	3,760	1,880
FGVC Aircraft (Maji et al., 2013)	100	6,667	3,333
SUN397 (Xiao et al., 2010)	397	19,850	19,850
PASCAL VOC 2007 (Everingham et al., 2009)	20	5,011	4,952
COCO (Lin et al., 2015)	80	118,287	5,000
ADE20K (Zhou et al., 2018)	150	20,210	2,000

A.4.1 SynCo-v2.

For synthetic negatives, we empirically select the top $N = 256$ hardest negatives from $\mathcal{Q}$ to form $\hat{\mathcal{Q}}^N$, and generate 128 synthetic hard negatives for each of the six transformation strategies, resulting in $|\mathcal{S}^i| = 128$ for each strategy i, totaling $128 \times 6 = 768$ synthetic negatives per training iteration. For Types 1–3 (interpolation, extrapolation, and mixup), the coefficients $(\alpha_k, \beta_k, \gamma_k)$ are sampled independently and uniformly at random for each synthetic negative at every training iteration (*i.e.*, per batch, not per epoch) from the ranges $\alpha_k \sim \mathcal{U}(0, 0.5)$, $\beta_k \sim \mathcal{U}(1, 1.5)$, and $\gamma_k \sim \mathcal{U}(0, 1)$. This stochastic sampling ensures diversity in the geometric exploration of the embedding space across training iterations. For Types 4–6 (noise injection, perturbed, and adversarial negatives), the scalar coefficients remain fixed throughout training with $\sigma = 0.01$, $\delta = 0.01$, and $\eta = 0.01$. We use a temperature $\tau = 0.2$ for the contrastive loss and a queue size $K = 4096$ for the memory bank $\mathcal{Q}$. We implement a warmup phase for the first 30 epochs and a cooldown period for the last 100 epochs, during which no synthetic negatives $\mathcal{S}$ are generated. This cooldown strategy stabilizes training and prevents potential over-fitting to synthetic samples in the final training stages (Giakoumoglou and Stathaki, 2025). These hyperparameters (α_k, β_k, γ_k ranges, and σ, δ, η values) were adopted from (Giakoumoglou and Stathaki, 2025), where they were *empirically* determined based on the intuition of each transformation strategy for convolutional networks. We keep these hyperparameters fixed throughout our experiments with vision transformers. Notably, Giakoumoglou and Stathaki (2025) reported *no* significant performance differences in ablations over σ, δ, and η, suggesting robustness to these specific values.

A.4.2 MoBY.

For reproduction of MoBY (Xie et al., 2021a), we follow the original implementation as in Section A.4 and as described in Xie et al. (2021a).

A.4.3 BYOL.

For reproduction of BYOL (Grill et al., 2020), we follow the original implementation with some adaptations. We use gradient accumulation of 8 steps resulting in an effective batch size of 4096, maintaining the same base learning rate of 1.5×10^{-4} and weight decay of 0.1. Unlike SynCo-v2 and MoBY, BYOL does not use drop path regularization (setting both *online* and *target* drop path to 0.0) or contrastive temperature, as it employs a mean squared error loss instead of InfoNCE.

A.5 Linear Probing

For linear evaluation, we follow the standard linear evaluation protocol of (He et al., 2020; Kornblith et al., 2019; Kolesnikov et al., 2019), which consists of training a linear classifier on top of the frozen features without updating the backbone network parameters or batch normalization statistics. The linear classifier operates on the [CLS] token representations for ViT-S/B and the final global average pooled features for Swin-T/S. We use random resized cropping with scale from $[0.08, 1]$ and horizontal flipping as data augmentation and train for 100 epochs with a 5-epoch linear warmup, weight decay of 0, and SGD optimizer with momentum of 0.9. For linear evaluation, we select the learning rate via grid search over $\{0.1, 0.5, 0.75, 1.0, 1.25, 1.5\}$ and report top-1 and top-5 accuracy on center crops during validation. We apply label smoothing with $\epsilon = 0.1$ for improved generalization.

A.6 Fine-tuning

For fine-tuning evaluations on transfer learning tasks, we initialize networks with pretrained weights and adapt all parameters during downstream task training, following established protocols (Kornblith et al., 2019; Kolesnikov et al., 2019; Dosovitskiy et al., 2021). We train for 100 epochs using SGD optimizer with momentum of 0.9, a base learning rate of 1×10^{-3}, and weight decay of 1×10^{-4}. The learning rate follows a cosine decay schedule with a 10-epoch linear warmup period. We apply standard data augmentation including random resized crops with scale $[0.08, 1.0]$ and random horizontal flipping during training. Additional regularization includes dropout with rate 0.1 in the classifier head and label smoothing with $\epsilon = 0.1$. We apply gradient clipping with a maximum norm of 1.0 to ensure stable training. We do *not* perform extensive hyperparameter search for downstream evaluations and instead maintain consistent configurations across all tasks.

A.7 Other Evaluations Protocols

We follow the established evaluation protocols from prior work to ensure fair comparison and reproducibility. For image retrieval on revisited Oxford and Paris datasets and copy detection on Copydays, we adopt the protocol from DINO (Caron et al., 2021), using frozen features with k-NN matching and reporting Mean Average Precision (mAP) for Medium (M) and Hard (H) evaluation protocols. For video object segmentation on DAVIS 2017, we follow the DINO protocol (Caron et al., 2021), using nearest-neighbor matching between consecutive frames without any training or finetuning, and report mean region similarity ($\mathcal{J}_m$), mean contour accuracy ($\mathcal{F}_m$), and their respective recall metrics ($\mathcal{J}_r$, $\mathcal{F}_r$) at 480p resolution. For object detection and instance segmentation on COCO, we adopt the transfer learning protocol from MoBY (Xie et al., 2021a), finetuning Cascade Mask R-CNN (Cai and Vasconcelos, 2019) with pretrained backbones and reporting bounding box mAP (mAP^{bb}) and mask mAP (mAP^{msk}). For semantic segmentation on ADE20K, we follow the MoBY protocol (Xie et al., 2021a), finetuning UPerNet (Xiao et al., 2018) with pretrained encoders and reporting mean Intersection over Union (mIoU). These protocols are detailed in the respective original papers and ensure our results are directly comparable to prior work.

B Extended Results

In this section, we provide comprehensive evaluations of SynCo-v2 on k-NN classification (Section B.1), robustness benchmarks (Section B.2), adversarial attacks (Section B.3), transfer learning on downstream classification tasks (Section B.4), and computational efficiency analysis (Section B.5).

B.1 k-NN Classification Evaluation

To thoroughly evaluate the quality of representations learned by SynCo-v2, we analyze k-nearest neighbors (k-NN) classification performance across multiple values of k on ImageNet ILSVRC-2012. As shown in Table 15, SynCo-v2 consistently outperforms both BYOL and MoBY across all evaluated neighborhood sizes for both ViT-S and Swin-T architectures. The performance advantage is most evident at smaller neighborhood sizes ($k = 10, 20$), where SynCo-v2 achieves improvements of **+6.5-8.0%** over baselines with ViT-S and **+1.4-9.4%** with Swin-T. This indicates that synthetic hard negatives help create a feature space where the closest neighbors are highly semantically relevant, enabling more accurate fine-grained discrimination.

Table 15 **k-NN classification on ImageNet ILSVRC-2012 across different neighborhood sizes.** Top-1 k-NN accuracy (%) for different values of k. Best results are in **bold**.

Method	$k = 10$	$k = 20$	$k = 100$	$k = 200$
	ViT-S			
BYOL (Grill et al., 2020) (*repr.*)	62.53	62.43	60.49	59.12
MoBY (Xie et al., 2021a) (*repr.*)	64.38	64.26	61.42	59.74
SynCo-v2 (ours)	**71.02**	**70.45**	**67.66**	**65.86**
	Swin-T			
BYOL (Grill et al., 2020) (*repr.*)	58.10	57.96	55.14	53.34
MoBY (Xie et al., 2021a) (*repr.*)	67.84	67.14	63.68	61.82
SynCo-v2 (ours)	**69.28**	**68.72**	**65.24**	**63.17**

B.2 Out-of-Distribution and Robustness Evaluation

To comprehensively assess the robustness and generalization capabilities of learned representations, we evaluate on multiple variants of ImageNet that test different aspects of model reliability. For robustness evaluation, we employ four datasets: **ImageNet-v2** (Recht et al., 2019), which comprises three test sets of 10,000 images each (matched frequency (MF), threshold 0.7 (T0.7), and top images (TI)) collected with a new data collection process to assess distribution shift; **ImageNet-C** (IN-C) (Hendrycks and Dietterich, 2019), which applies 15 different corruption types (noise, blur, weather, and digital distortions) at 5 severity levels to measure resilience to common corruptions; **ImageNet-A** (IN-A) (Hendrycks et al., 2021b), containing naturally occurring adversarial examples that are difficult for ImageNet-trained models; and **ImageNet-Watermark** (IN-W) (Li et al., 2023), which includes watermarked versions of ImageNet images to test robustness to artificial perturbations. For out-of-distribution detection and generalization, we use three datasets: **ImageNet-Sketch** (IN-S) (Wang et al., 2019), containing 50,000 black-and-white sketch drawings of ImageNet objects; **ImageNet-R** (IN-R) (Hendrycks et al., 2021a), consisting of 30,000 artistic renditions across various styles (paintings, cartoons, sculptures, graffiti, embroidery) to test domain shift robustness; and **ImageNet-O** (IN-O) (Hendrycks et al., 2021b), containing objects from classes not in ImageNet-1K to measure anomaly detection via FPR95 (false positive rate at 95% true positive rate, where lower is better). For all robustness benchmarks except ImageNet-O, we report top-1 accuracy (in %). For ImageNet-C, we report the mean accuracy across all corruption types and severity levels. For ImageNet-O, we follow the standard protocol (Hendrycks et al., 2021a) and report FPR95, where a lower value indicates better out-of-distribution detection capability. All evaluations use the linear classifiers trained on clean ImageNet, without any fine-tuning on the perturbed or out-of-distribution data. We use center crop preprocessing for all evaluations to ensure fair comparison across methods. As shown in Table 16, SynCo-v2 achieves better overall balance across diverse robustness benchmarks. These results suggest that training with synthetic hard negatives improves the robustness of learned representations to various distribution shifts and corruptions, without sacrificing clean accuracy.

B.3 Adversarial Robustness Evaluation

We evaluate the adversarial robustness of SynCo-v2 by testing against a comprehensive suite of adversarial attacks that represent diverse adversarial threat models. Following standard practices in adversarial machine learning (Madry et al., 2018), we assess model performance against both white-box and black-box attacks on the ImageNet validation set. All attacks are implemented using the torchattacks library (Kim, 2020)[3]. Our evaluation includes: **FGSM** (Fast Gradient Sign Method) (Goodfellow et al., 2014), a single-step gradient-based attack with $\epsilon = 8/255$; **PGD** (Projected Gradient Descent) (Madry et al., 2018), an iterative gradient-based attack with $\epsilon = 8/255$, step size $\alpha = 2/255$, and 10 iterations; **C&W** (Carlini & Wagner) (Carlini and Wagner, 2017), an optimization-based attack that minimizes ℓ_2 perturbation with confidence parameter $\kappa = 0$, 50 optimization steps, learning rate of 0.01, and initial constant $c = 10^{-4}$; **Square Attack** (Andriushchenko et al., 2020), a score-based black-box attack using ℓ_∞ norm with 1,000 queries; **AutoAttack** (Croce and Hein, 2020), an ensemble of attacks that adaptively combines multiple methods using ℓ_∞ norm; **TIFGSM**

[3]Available at: https://github.com/Harry24k/adversarial-attacks-pytorch.

Table 16 Robustness and out-of-distribution evaluation of self-supervised methods. Results show top-1 accuracy (in %) on various ImageNet variants using ViT-S and Swin-T backbones, except ImageNet-O where FPR95 is reported. Best results are in **bold**.

Method	Ep.	IN	Robustness						Out-Of-Distribution		
			MF	T0.7	TI	IN-C	IN-A	IN-W	IN-S	IN-R	IN-O
ViT-S											
BYOL (Grill et al., 2020) (repr.)	300	70.3	58.0	66.4	72.5	41.5	7.0	54.2	18.2	27.8	87.2
MoBY (Xie et al., 2021a) (repr.)	300	72.3	60.0	67.8	74.5	39.5	**16.4**	49.6	**24.1**	**32.3**	**100.0**
SynCo-v2 (ours)	300	**73.1**	**60.8**	**69.6**	**75.5**	**44.5**	10.2	**55.7**	22.3	31.7	**100.0**
Swin-T											
BYOL (Grill et al., 2020) (repr.)	300	68.5	56.5	64.7	70.6	40.4	6.9	52.8	17.7	27.1	**85.0**
MoBY (Xie et al., 2021a) (repr.)	300	74.7	62.9	71.4	77.3	44.8	7.5	**63.0**	22.9	35.2	83.1
SynCo-v2 (ours)	300	**75.4**	**64.0**	**72.4**	**78.0**	**46.0**	**8.6**	62.4	**23.1**	**35.8**	77.6

(Translation-Invariant FGSM) (Dong et al., 2019), which incorporates translation invariance with $\epsilon = 8/255$, $\alpha = 2/255$, and 10 steps; and **OnePixel** (Su et al., 2019), which modifies only a single pixel per image using differential evolution with 10 steps. All attacks are performed on the ImageNet validation set using the linear classifiers trained on clean ImageNet. We do not perform adversarial training and evaluate the natural robustness of representations learned through self-supervised pretraining. For gradient-based attacks (FGSM, PGD, C&W, TIFGSM), we use the default threat model with ℓ_∞ norm constraint of $\epsilon = 8/255$ for FGSM and PGD. For C&W, we use the ℓ_2 norm with confidence parameter $\kappa = 0$. For black-box attacks (Square, AutoAttack), we use the default query budgets. All evaluations use untargeted attacks and report top-1 accuracy (in %) on adversarial examples. As shown in Table 17, SynCo-v2 achieves better overall balance across diverse adversarial threats. These results suggest that synthetic hard negatives encourage the model to learn more robust decision boundaries that are less susceptible to adversarial perturbations, without requiring explicit adversarial training.

Table 17 Adversarial robustness evaluation of self-supervised methods. Results show top-1 accuracy (in %) under various attacks on ImageNet validation set using ViT-S and Swin-T architectures. Best results are in **bold**. **Abbreviations**: Cln: Clean, FG: FGSM, CW: C&W, Sq: Square, AA: AutoAttack, TI: TI-FGSM, 1Px: OnePixel.

Method	Ep.	Cln	FG	PGD	CW	Sq	AA	TI	1Px
ViT-S									
BYOL (Grill et al., 2020) (repr.)	300	70.3	25.0	3.0	29.5	**22.4**	2.7	11.8	68.9
MoBY (Xie et al., 2021a) (repr.)	300	72.3	**27.1**	14.8	34.0	20.4	**13.1**	17.0	70.4
SynCo-v2 (ours)	300	**73.1**	24.6	**15.8**	**37.**7	19.6	12.6	**18.**7	**71.4**
Swin-T									
BYOL (Grill et al., 2020) (repr.)	300	68.5	**24.4**	**2.9**	28.8	21.9	2.6	11.5	67.1
MoBY (Xie et al., 2021a) (repr.)	300	74.7	22.7	1.5	27.6	25.8	1.2	14.1	73.7
SynCo-v2 (ours)	300	**75.4**	22.3	2.1	**29.7**	**26.3**	**2.9**	**14.3**	**74.4**

B.4 Transfer Learning on Classification Tasks

We evaluate the transfer learning capability of SynCo-v2 by assessing how well the learned representations generalize to diverse downstream classification tasks beyond ImageNet. Following standard linear probing protocols (Chen et al., 2020b; Grill et al., 2020), we freeze the pretrained backbone weights and train only a single fully-connected classification layer on each target dataset. We evaluate on eleven diverse downstream datasets spanning various visual domains: **CIFAR-10** and **CIFAR-100** (Krizhevsky, 2009), containing 32×32 natural images with 10 and 100 classes respectively; **Oxford Flowers-102** (Nilsback and Zisserman, 2008), consisting of 102 flower categories; **Oxford-IIIT Pets** (Parkhi et al., 2012), containing 37 pet breeds; **Food-101** (Bossard et al., 2014), comprising 101 food categories; **Stanford Cars** (Krause et al., 2013), featuring 196

car models; **Caltech-101** (Fei-Fei et al., 2007), containing 101 object categories; **DTD** (Describable Textures Dataset) (Cimpoi et al., 2014), with 47 texture categories; **FGVC-Aircraft** (Maji et al., 2013), consisting of 100 aircraft variants; **SUN397** (Xiao et al., 2010), containing 397 scene categories; and **PASCAL VOC 2007** (Everingham et al., 2009), featuring 20 object classes for multi-label classification. These datasets cover fine-grained recognition (Flowers, Pets, Cars, Aircraft), scene understanding (SUN397), texture recognition (DTD), and general object classification across various scales and domains. As shown in Table 18, SynCo-v2 achieves competitive or superior performance across most downstream tasks compared to BYOL and MoBY baselines. These results suggest that training with synthetic hard negatives improves the generalization capability of learned representations to diverse visual domains, without requiring additional pretraining data or domain-specific adaptations.

Table 18 Linear probing performance on various downstream classification datasets. Results show top-1 accuracy (in %) with frozen weights except for the final fully-connected layer. SynCo-v2 demonstrates strong transferability across diverse visual domains, achieving best or competitive performance on most benchmarks. Best results are in **bold**. **Abbreviations**: C_{10}/C_{100}: CIFAR-10/100, Flw: Flowers-102, Fd_{101}: Food-101, Cal_{101}: Caltech-101, Air: FGVC-Aircraft, SUN: SUN397, VOC: PASCAL VOC.

Method	C_{10}	C_{100}	Flw	Pets	Fd_{101}	Cars	Cal_{101}	DTD	Air	SUN	VOC
ViT-S											
BYOL (Grill et al., 2020) (*repr.*)	90.5	74.2	**87.7**	85.1	73.3	49.2	**99.1**	69.2	38.3	77.2	77.0
MoBY (Xie et al., 2021a) (*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 (Grill et al., 2020) (*repr.*)	88.6	72.2	83.8	83.0	73.7	34.7	99.0	71.8	28.9	77.9	78.5
MoBY (Xie et al., 2021a) (*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**

B.5 Computational Efficiency Analysis

To assess practical applicability, we analyze computational requirements, including parameters, memory usage, and wall-clock training time.

Memory and parameter overhead. The memory overhead of storing 768 synthetic negatives is approximately 50 MB, negligible compared to the total footprint and much smaller than storing $K = 4096$ memory bank negatives. The method introduces no meaningful increase in parameter count, as the synthetic negatives are generated on the fly and do not require additional learnable parameters (*cf.* Table 19).

Table 19 Computational efficiency comparison. Model parameters, memory consumption, and training duration for all methods trained with batch size 512 across 4 GPUs on ImageNet ILSVRC-2012. Lower memory and training time are better (indicated by ↓).

Method	Total Params	Train. Params	Mem. ↓ (MB)	Time/Ep. ↓ (min)	100 Ep. ↓ (h)	300 Ep. ↓ (h)
ViT-S						
BYOL (Grill et al., 2020)	50.7M	26.3M	21,316	31.60	52.66	157.97
MoBY (Xie et al., 2021a)	50.7M	26.3M	21,325	32.71	54.52	163.55
SynCo-v2 (ours)	50.7M	26.3M	21,325	31.60	52.66	157.99
Swin-T						
BYOL (Grill et al., 2020)	65.6M	33.8M	29,167	39.72	66.19	198.58
MoBY (Xie et al., 2021a)	65.6M	33.8M	29,177	39.78	66.31	198.93
SynCo-v2 (ours)	65.6M	33.8M	29,176	38.75	64.58	193.75

Generation time and amortization. As shown in Table 20, Types 1–4 (geometric transformations) require less than 1 ms each to generate 128 negatives, while Types 5–6 (gradient-based perturbations) take about 11 ms each, remaining efficient overall. The total synthetic negative generation time is roughly 24 ms for all six strategies, which is amortized over the full iteration composed of two forward passes and a backward pass (typically 2−3× slower than a single forward pass).

Table 20 Average inference time for synthetic negative generation. The reported time (in ms) corresponds to generating $N_i = 128$ hard negatives per strategy, averaged over 100 runs on a single GPU. Standard deviations are reported as ±.

Component	Time (ms)
ViT-S forward pass	175.14 ± 4.40
Projector forward pass	0.22 ± 1.22
Type 1 (Interpolation)	0.51 ± 1.87
Type 2 (Extrapolation)	0.45 ± 1.87
Type 3 (Mixup)	0.65 ± 2.29
Type 4 (Noise Injection)	0.43 ± 1.55
Type 5 (Perturbed)	10.98 ± 1.94
Type 6 (Adversarial)	11.13 ± 1.99
Total synthetic negatives	~ 24

Overall training efficiency. As shown in Table 19, the overall runtime of SynCo-v2 is comparable to training without synthetic negatives, and the "on-the-fly" generation avoids pre-computed datasets or external augmentation models. Our performance gains and emergent properties far exceed the overhead, which is neglible (training time for 300 epochs almost similar to training without synthetic negatives), with SynCo-v2's small models competitive with larger, more expensive models such as I-JEPA (Assran et al., 2023), V-JEPA (Bardes et al., 2024b), and DINO (Caron et al., 2021).

C Extended Ablations

In this section, we provide extended ablation studies in tabular form for the hyperparameter sensitivity experiments on queue size K, temperature τ, and momentum m_{start} presented in Section 5, along with a detailed analysis of synthetic negative configurations.

C.1 Hyperparameter Sensitivity

We report full numerical results for the queue size, temperature, and momentum ablations discussed in Section 5. All experiments use ViT-S and Swin-T pretrained for 100 epochs on ImageNet-100 with default settings ($K = 4096$, $\tau = 0.2$, $m_{\text{start}} = 0.99$) unless the respective hyperparameter is varied. Tables 21 to 23 confirm that SynCo-v2 is robust across a wide range of hyperparameter settings.

Table 21 Queue size ablation. Top-1 accuracy (%) on ImageNet-100 for varying queue sizes K, with $\tau = 0.2$ and $m_{\text{start}} = 0.99$ (*cf.* Figure 5). We highlight the default configuration.

K	ViT-S	Swin-T
1024	64.5	72.5
2048	64.5	72.5
4096	**64.7**	**72.7**
8192	63.6	72.3
16384	62.6	71.6

Table 22 Temperature ablation. Top-1 accuracy (%) on ImageNet-100 for varying τ, with $K = 4096$ and $m_{\text{start}} = 0.99$ (*cf.* Figure 5). We highlight the default configuration.

τ	ViT-S	Swin-T
0.07	59.3	61.5
0.1	61.5	69.2
0.2	**64.7**	**72.7**
0.3	64.0	71.7

Table 23 Momentum ablation. Top-1 accuracy (%) on ImageNet-100 for varying m_{start}, with $K = 4096$ and $\tau = 0.2$ (*cf.* Figure 5). We highlight the default configuration.

m_{start}	ViT-S	Swin-T
0.99	64.7	**72.7**
0.993	**65.2**	72.2
0.996	63.8	72.4
0.999	60.3	68.6

Table 24 Synthetic negative configuration ablation on ImageNet-100. Top-1 accuracy (%) after 100 epochs of pretraining. We fix $K = 4096$, $\tau = 0.2$, $m_{\text{start}} = 0.99$ and vary the number of synthetic negatives per strategy. We highlight the default configuration.

N	$\lvert S^1\rvert$	$\lvert S^2\rvert$	$\lvert S^3\rvert$	$\lvert S^4\rvert$	$\lvert S^5\rvert$	$\lvert S^6\rvert$	ViT-S	Swin-T
0	0	0	0	0	0	0	63.0	71.9
256	64	64	64	32	32	32	63.3	72.2
256	128	128	128	64	64	64	63.1	72.3
256	128	128	128	128	128	128	64.7	72.7
256	256	256	256	64	64	64	63.4	72.4
256	256	256	256	128	128	128	63.4	72.5
256	256	256	256	256	256	256	63.3	72.5
512	64	64	64	32	32	32	62.0	72.5
512	128	128	128	64	64	64	62.7	71.8
512	128	128	128	128	128	128	62.1	72.4
512	256	256	256	64	64	64	62.1	72.6
512	256	256	256	128	128	128	63.6	71.9
512	256	256	256	256	256	256	64.3	**73.2**
1024	64	64	64	32	32	32	63.4	72.4
1024	128	128	128	64	64	64	62.7	72.4
1024	128	128	128	128	128	128	62.9	72.4
1024	256	256	256	64	64	64	64.5	72.9
1024	256	256	256	128	128	128	64.6	72.3
1024	256	256	256	256	256	256	63.0	72.3
1024	512	512	512	64	64	64	**64.8**	72.5
1024	512	512	512	128	128	128	63.1	72.6
1024	512	512	512	256	256	256	63.2	72.1

C.2 Synthetic Negative Configurations

We ablate the total number of synthetic negatives $|\mathcal{S}| = \sum_{i=1}^{6} |S^i|$ and their distribution across the six transformation strategies defined in Equation (1), where $|S^i|$ denotes the number of synthetic negatives generated by the i-th strategy. All experiments fix $K = 4096$, $\tau = 0.2$, and $m_{\text{start}} = 0.99$, using ViT-S and Swin-T pretrained for 100 epochs on ImageNet-100. We evaluate various configurations of $(|S^1|, |S^2|, |S^3|, |S^4|, |S^5|, |S^6|)$ across three scales of top-N hardest negatives selected from the queue $\hat{\mathcal{Q}}^N$, with $N \in \{256, 512, 1024\}$. Table 24 shows that SynCo-v2 maintains robust performance across a wide range of configurations. We adopt the default configuration $|S^i| = 128$ for $i = 1, \ldots, 6$ with total $|\mathcal{S}| = 768$ synthetic negatives, which achieves strong results on both architectures.

D Extended Related Work

In this section, we provide additional context on the broader self-supervised learning landscape (Section D.1), contrastive learning evolution (Section D.2), hard negative mining strategies (Section D.3), self-supervised vision transformer adaptations (Section D.4), and emergent semantic properties observed in transformer-based representations (Section D.5).

D.1 Self-supervised Learning

Self-supervised learning has evolved through multiple paradigms, each with distinct approaches to learning visual representations. Early pretext task methods (Gidaris et al., 2018; Noroozi and Favaro, 2016; Zhang et al., 2016) defined auxiliary supervised tasks to learn features, but were limited by task-specific biases. *Joint embedding methods* (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Caron et al., 2021) learn by

comparing different views of data, with different approaches to avoiding representational collapse. In contrast to joint embedding approaches that learn through comparison and distinction between different views or instances, an alternative major branch of self-supervised learning adopts a *generative* methodology. These generative techniques learn through reconstruction or prediction of input data components, with Masked Image Modeling (MIM) establishing itself as a particularly effective strategy. iGPT (Chen et al., 2020a) introduced the idea of treating images as sequential data for autoregressive prediction, which was subsequently followed by BEiT (Bao et al., 2022) and BEiT-v2 (Peng et al., 2022), transferring BERT-style (Devlin et al., 2018) masked prediction methodologies to computer vision. MAE (He et al., 2021) demonstrated that extensive masking of image patches (reaching 75%) establishes an effective self-supervised objective, while SimMIM (Xie et al., 2022) streamlined the methodology with lightweight prediction architectures. Subsequent improvements include: MaskFeat (Wei et al., 2023) which predicted HOG features rather than raw pixels, Context Autoencoder (Chen et al., 2023b) which exploited contextual relationships, and MSN (Assran et al., 2022) which integrated masking with siamese network architectures. Contemporary research has emphasized efficiency and effectiveness through methodologies including SiamMAE (Gupta et al., 2023), MixMAE (Liu et al., 2023a), PixMIM (Liu et al., 2023b), and TinyMIM (Ren et al., 2023). While recent developments include CropMAE (Eymaël et al., 2024) featuring efficient siamese cropped autoencoder designs and ColorMAE (Hinojosa et al., 2024) investigating data-independent masking approaches. These generative methodologies differ fundamentally from discriminative techniques by learning to predict or reconstruct absent information rather than contrasting different views or instances, providing a complementary pathway to self-supervised visual learning. While generative methods currently achieve higher top-1 accuracy on ImageNet (He et al., 2021), joint embedding approaches often surpass them in linear probing tasks and exhibit stronger emergent properties such as unsupervised semantic segmentation (Caron et al., 2021; Oquab et al., 2023).

D.2 Contrastive Learning

Contrastive learning has progressed from early instance discrimination methods (Wu et al., 2018) to sophisticated frameworks employing momentum encoders (He et al., 2020; Chen et al., 2020d), large batch sizes (Chen et al., 2020b), and advanced augmentation strategies (Chen et al., 2020c). Key innovations include the introduction of momentum-based memory banks to increase the number of negative samples (He et al., 2020), the demonstration that very large batch sizes can replace memory banks (Chen et al., 2020b), and the development of asymmetric network designs that avoid collapse without explicit negative samples (Grill et al., 2020; Chen and He, 2020b). More recent works have explored nearest-neighbor contrastive learning (Dwibedi et al., 2021), representation decorrelation (Zbontar et al., 2021), relative location prediction (Tomasev et al., 2022), and region-level contrastive learning (Xie et al., 2021b). Despite these advances, the role of negative sample quality versus quantity remains an active research question (Robinson et al., 2021; Ali et al., 2024), with ongoing debates about whether contrastive methods can match the performance of generative approaches without extensive computational resources.

D.3 Hard Negative Mining

Hard negative mining has a rich history in computer vision, originally developed for object detection (Felzenszwalb et al., 2010; Shrivastava et al., 2016) and later adapted for metric learning (Schroff et al., 2015; Sohn, 2016). In contrastive learning, various strategies have emerged to identify and leverage hard negatives: selecting negatives with high similarity to the anchor (Robinson et al., 2021), using nearest neighbors in the embedding space (Dwibedi et al., 2021), employing debiased sampling to address false negatives (Yeh et al., 2022), and generating synthetic hard negatives through mixing (Kalantidis et al., 2020) or adversarial perturbations (Ho and Vasconcelos, 2020). It is important to distinguish between methods that augment *positive* pairs versus those that generate hard *negatives*. Nearest-neighbor approaches like NNCLR (Dwibedi et al., 2021) and MSF (Koohpayegani et al., 2021) retrieve similar samples from a support set to enrich the positive set, helping models learn invariance to different views of the same semantic content. In contrast, hard negative generation methods like MoCHI (Kalantidis et al., 2020) create challenging negative samples to refine decision boundaries and improve discriminative power. These strategies are complementary: positive augmentation improves feature invariance (learning what to ignore), while hard negative generation improves feature discrimination (learning what distinguishes instances). SynCo (Giakoumoglou and Stathaki, 2025) systematically explored six synthetic negative generation strategies for convolutional networks, demonstrating

consistent improvements. Our work extends this approach to vision transformers, showing that the benefits of synthetic hard negatives transfer effectively to transformer architectures and enhance both discriminative performance and emergent semantic properties.

D.4 Self-supervised Vision Transformers

The adaptation of transformers to computer vision (Dosovitskiy et al., 2021; Touvron et al., 2021; Liu et al., 2021) has enabled new self-supervised learning approaches. DINO (Caron et al., 2021) demonstrated that self-distillation with vision transformers produces strong emergent properties in attention mechanisms, while MoCo-v3 (Chen et al., 2021) and MoBY (Xie et al., 2021a) showed that contrastive learning can be effectively adapted to transformers with appropriate modifications such as random patch projection and asymmetric architectures. Masked autoencoding methods like MAE (He et al., 2021), BEiT (Bao et al., 2022), and SimMIM (Xie et al., 2022) have achieved state-of-the-art results by leveraging transformers' ability to handle variable-length sequences and reconstruct masked patches. Recent works have combined multiple self-supervised objectives, such as iBOT (Zhou et al., 2022) which integrates masked image modeling with self-distillation, and I-JEPA (Assran et al., 2023) which predicts representations of masked regions in latent space. Scaling efforts have demonstrated that self-supervised vision transformers can be trained on billions of images to achieve remarkable zero-shot and few-shot capabilities, as shown by DINO-v2 (Oquab et al., 2023), SEER (Goyal et al., 2021), and V-JEPA (Bardes et al., 2024a). Our work demonstrates that simple contrastive learning with improved negative sampling can achieve competitive performance and emergent properties without requiring complex training procedures, multi-crop augmentation strategies (Caron et al., 2020), or massive computational scale, making it an accessible and effective alternative for practitioners with limited resources.

D.5 Emergent Properties

A particularly intriguing aspect of self-supervised vision transformers is their emergent semantic segmentation capabilities without explicit supervision. DINO (Caron et al., 2021) first observed that self-attention maps naturally correspond to object boundaries and semantically meaningful regions, enabling applications in unsupervised object discovery and weakly-supervised segmentation. DINO-v2 (Oquab et al., 2023) demonstrated that these properties strengthen with scale, achieving impressive zero-shot segmentation performance. Recent work has explored the mechanisms behind these emergent properties: TokenCut (Wang et al., 2023) leverages self-attention for unsupervised image segmentation, LOST (Siméoni et al., 2021) localizes objects in images without supervision using ViT features, and FreeSOLO (Wang et al., 2022) performs self-supervised instance segmentation. Our work contributes to this line of research by demonstrating that contrastive learning with synthetic hard negatives can elicit similar emergent properties, suggesting that these capabilities are not exclusive to self-distillation methods but can be enhanced through improved negative sampling strategies that encourage semantically meaningful attention patterns.

E Discussion

In this section, we discuss the computational efficiency and design principles of SynCo-v2, covering the intuition behind each synthetic negative strategy (Section E.1), comparison to DINO visualizations (Section E.2), and potential extensions (Section E.3).

E.1 Intuition of SynCo-v2

To address why these specific transformations improve representation learning, we provide intuition for each of the six synthetic negative generation strategies (Giakoumoglou and Stathaki, 2025). Type 1 (interpolation) creates samples between the query and hard negatives, increasing diversity and encouraging refined decision boundaries in semantically ambiguous regions. Type 2 (extrapolation) pushes beyond the query away from hard negatives, exploring extreme representation space regions and improving robustness. Type 3 (mixup) combines hard negative pairs, creating novel challenging samples that encourage generalized feature learning beyond the original distribution. Type 4 (noise injection) adds Gaussian perturbations, promoting invariance

to minor fluctuations and enhancing generalization. Type 5 (gradient-based perturbation) modifies embeddings using similarity gradients with variable magnitude, refining discriminatory power by directing toward higher confusion regions. Type 6 (adversarial perturbation) applies sign-based gradient perturbations with fixed magnitude, creating maximally challenging contrasts. The complementary nature of these strategies ensures comprehensive coverage of challenging regions, preventing overfitting to specific negative patterns while maintaining appropriate task difficulty. For detailed analysis, we refer readers to Giakoumoglou and Stathaki (2025).

To address the fundamental question of *why* synthetic hard negatives improve representation learning, we provide theoretical motivation for our six transformation strategies. Our approach draws from two complementary principles in representation learning: geometric exploration and adversarial robustness. Strategies 1–4 (interpolation, extrapolation, mixup, and noise injection) perform geometric exploration around hard negatives in the embedding space. Interpolated negatives (S^1) create samples between the query and hard negatives, effectively densifying the decision boundary region. Extrapolated negatives (S^2) explore directions beyond hard negatives, pushing the boundary further from the anchor. Mixup negatives (S^3) combine pairs of hard negatives to discover intermediate semantic concepts that might not exist in the original dataset. Noise-injected negatives (S^4) add controlled stochasticity to prevent overfitting to specific negative samples. In contrast, strategies 5–6 (perturbed and adversarial negatives) leverage adversarial training principles by generating maximally confusing samples using gradient information. These gradient-based methods identify directions in the embedding space where the model is most uncertain, creating negatives that explicitly challenge the current decision boundary. The key insight is that geometric and gradient-based methods provide complementary coverage of the hard negative space: geometric strategies ensure broad exploration of the embedding manifold, while adversarial strategies target model-specific weaknesses. This diversity of hard negatives collectively refines decision boundaries more effectively than any single strategy.

We acknowledge that while our design choices are largely empirical—as is often the case in deep learning—they are motivated by established principles from augmentation theory and adversarial training. Importantly, our approach remains relatively simple compared to alternatives like DINO (Caron et al., 2021), which requires multiple specialized techniques (centering, sharpening, multi-crop augmentation, and extended training schedules). Our method integrates seamlessly with any InfoNCE-based framework through a straightforward augmentation of the memory queue, requiring no architectural modifications or complex training procedures.

E.2 Comparison to DINO Visualizations

While our attention visualizations (Figures 6 to 10) demonstrate that SynCo-v2 produces sharper and more semantically meaningful attention maps compared to MoBY and BYOL, we acknowledge that they do not yet reach the exceptional quality of DINO (Caron et al., 2021) and DINO-v2 (Oquab et al., 2023) visualizations. However, this comparison must be contextualized: DINO employs **(i)** multi-crop augmentation (2 global crops at 224×224 and multiple local crops at 96×96), **(ii)** trains for significantly longer (800-1600 epochs vs. our 300 epochs), and **(iii)** uses various training *tricks* (centering, sharpening, temperature scaling) specifically designed to improve stability and attention quality. In contrast, SynCo-v2 achieves its attention properties through simple contrastive learning with synthetic hard negatives, using only standard two-crop augmentation at 224×224 resolution and a 300-epoch training schedule. Despite these differences in training setup, SynCo-v2 demonstrates strong emergent segmentation capabilities that significantly surpass baseline contrastive learning (Xie et al., 2021a; Chen et al., 2021) and self-distillation without negatives (Grill et al., 2020; Caron et al., 2021). This suggests that synthetic hard negatives provide a complementary mechanism for improving attention quality that could potentially be combined with DINO-style training procedures.

E.3 Potential Extensions

Several promising directions could extend our work. First, exploring large language models or vision-language models to generate semantically-aware synthetic negatives could further improve representation quality by creating negatives that are challenging at a semantic rather than purely embedding-space level. Second, adapting our approach for multimodal contrastive learning (*e.g.*, CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021)) could improve vision-language alignment by providing harder cross-modal negatives. Third, investigating the role of synthetic hard negatives in continual learning or domain adaptation settings could

reveal whether they help mitigate catastrophic forgetting or improve transfer across distribution shifts. Fourth, combining synthetic negatives with recent advances in momentum encoders (*e.g.*, exponential moving average variants (Oquab et al., 2023)) or projection head designs could yield additional improvements. Finally, extending our analysis to other transformer variants (*e.g.*, Swin-V2 (Liu et al., 2022), CvT (Wu et al., 2021), PVT (Wang et al., 2021)) would validate the generality of our findings across diverse architectural designs.

F Limitations

In this section, we discuss the limitations of our work, including constraints on architectural scale (Section F.1), training budget (Section F.2), and hyperparameter tuning (Section F.3).

F.1 Architectural Scale

Due to limited computational resources, our experiments were constrained to smaller architectures (*cf.* Section A.1). Evaluating our approach on larger models such as ViT-Large (304M parameters) and Swin-Large (197M parameters) would provide valuable insights into scalability. However, these architectures require more than 16 GPUs to maintain the respective batch sizes (even for batch size 512), exceeding our computational budget. We believe the current architectures sufficiently demonstrate the effectiveness of synthetic hard negatives for vision transformers, given that: **(i)** we show consistent improvements across two distinct transformer architectures (ViT and Swin), suggesting generalization across different design paradigms, **(ii)** two sizes per architecture, and **(iii)** the method achieves meaningful gains over strong baselines under the same computational constraints.

F.2 Training Budget

Similarly, our experiments were limited to 300-epoch (*cf.* Section A.4) training schedules on ImageNet at 224×224 resolution; due to computational constraints, we did not perform multiple runs to assess variability. Investigating longer training regimes (*e.g.*, 800–1600 epochs as in DINO (Caron et al., 2021) and DINO-v2 (Oquab et al., 2023)) and higher resolutions could reveal whether synthetic hard negatives provide additional benefits with extended training budgets.

F.3 Hyperparameter Tuning

We did *not* perform exhaustive hyperparameter tuning across all possible combinations of synthetic negative strategies and their parameters. Instead, we adopted most hyperparameters from SynCo (Giakoumoglou and Stathaki, 2025) and made minimal adaptations for vision transformers. Nevertheless, we provide comprehensive ablations on synthetic negative strategy combinations (Table 9), drop path regularization (Table 10), implementation tricks (Table 11), queue size (Table 21), temperature (Table 22), and momentum (Table 23). Given the substantial gains already achieved with these empirically set hyperparameters, we found *no compelling* need for exhaustive tuning. While a more systematic exploration of strategy combinations, the number of synthetic negatives per strategy, hard negative selection thresholds, and transformation parameters could potentially yield marginal improvements, the consistent gains achieved with our current configuration suggest that the method is robust to hyperparameter choices, making it practical for adoption without extensive tuning.

G Attention Visualization

We present visualizations for ViT-Small (Section G.1), using [CLS] token attention and patch self-attention maps, and for Swin-Tiny (Section G.2), using gradient-based Class Activation Mapping and saliency methods adapted to the hierarchical window-based architecture.

G.1 Vision Transformer

For ViT-Small, we extract and visualize attention maps using two complementary approaches: **(i)** [CLS] token attention, and **(ii)** patch self-attention. We extract attention weights from the last transformer layer following DINO visualization protocol (Caron et al., 2021). For [CLS] attention, we take the attention weights from the [CLS] token to all patch tokens, average across attention heads, reshape to the spatial grid (14×14 for patch size 16), and upsample to the original image resolution (224×224) using bilinear interpolation. For patch self-attention, we compute the average attention each patch receives from all other patches (excluding the [CLS] token), which naturally highlights salient regions and object boundaries. We apply thresholding to focus on the most attended regions. As shown in Figures 6 to 10, SynCo-v2 consistently produces sharper and more semantically meaningful attention patterns compared to both MoBY and BYOL.

G.2 Swin Transformer

For Swin-Tiny, direct attention map extraction is challenging due to the hierarchical window-based architecture where attention is computed within local windows rather than globally. Therefore, we employ two complementary gradient-based visualization methods to reveal which spatial regions most strongly influence the model's predictions: **(i)** Class Activation Mapping (CAM) (Selvaraju et al., 2019), and **(ii)** Gradient Saliency. For CAM visualization (Selvaraju et al., 2019), we extract the final feature map before the global average pooling layer and compute the weighted combination of feature maps using the gradients of the predicted class score with respect to the feature activations. For gradient saliency, we compute the absolute gradient of the predicted class score with respect to input pixels: Saliency $= \left|\frac{\partial y^c}{\partial \mathbf{x}}\right|$, which highlights regions with the largest influence on predictions. We average across color channels and normalize the saliency map. Both visualizations use the predicted class (rather than ground truth) to reflect what the model actually learned. As shown in Figures 11 to 15, despite the architectural differences between ViT and Swin, we observe consistent patterns where SynCo-v2 produces more focused and semantically meaningful gradient-based attention maps.

H Broader Impact

The presented research should be categorized as research in the field of unsupervised learning. This work may inspire new algorithms, theoretical, and experimental investigation. The algorithm presented here can be used for many different vision applications and a particular use may have both positive or negative impacts, which is known as the dual use problem. Besides, as vision datasets could be biased, the representation learned by SynCo-v2 could be susceptible to replicate these biases.

I Checkpoint Availability

The pre-trained model checkpoints for models trained on the ImageNet ILSVRC-2012 dataset are available for download: ViT-S (top-1 linear evaluation accuracy 73.1%), ViT-B (top-1 linear evaluation accuracy 77.1%), Swin-T (top-1 linear evaluation accuracy 75.4%), and Swin-S (top-1 linear evaluation accuracy 78.0%).

J Reproducibility Statement

To ensure full reproducibility, we provide comprehensive implementation details in Section A, including architectural specifications (Section A.1), hyperparameters (Sections A.2 and A.4), and training configurations (Sections A.5 and A.6). Source code is available at `https://github.com/giakoumoglou/synco-v2` and pre-trained model checkpoints trained on ImageNet ILSVRC-2012 are available for download (Section I). All experiments use publicly available datasets with standard evaluation protocols (Chen et al., 2020b; Grill et al., 2020), and fixed random seeds are used throughout, though minor variations may occur due to non-deterministic CUDA operations. We believe these resources, combined with the detailed descriptions in our paper, will enable other researchers to replicate our results and build upon our work.

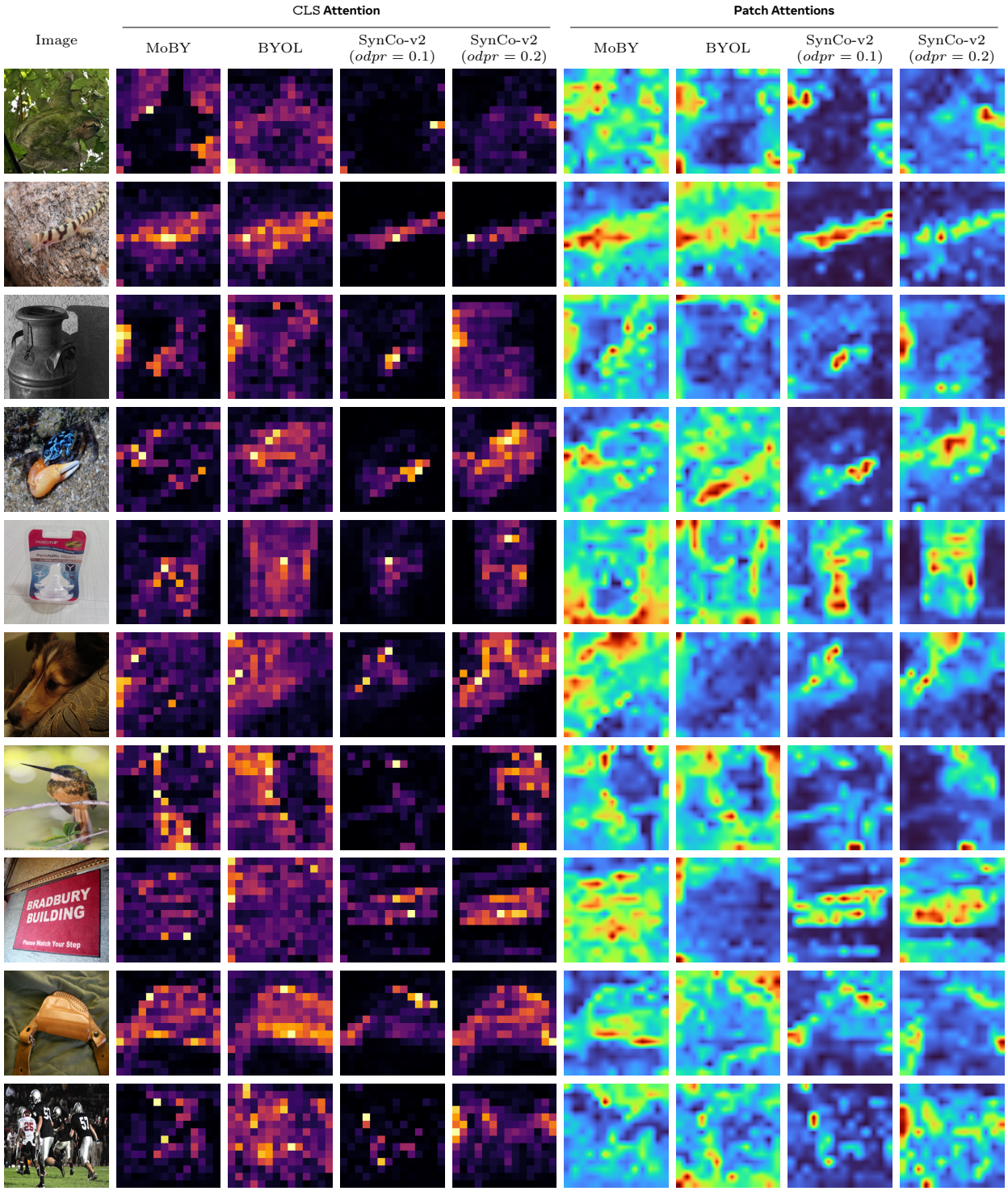


Figure 6 ViT-S/16 attention visualization. We show images 1 to 10.

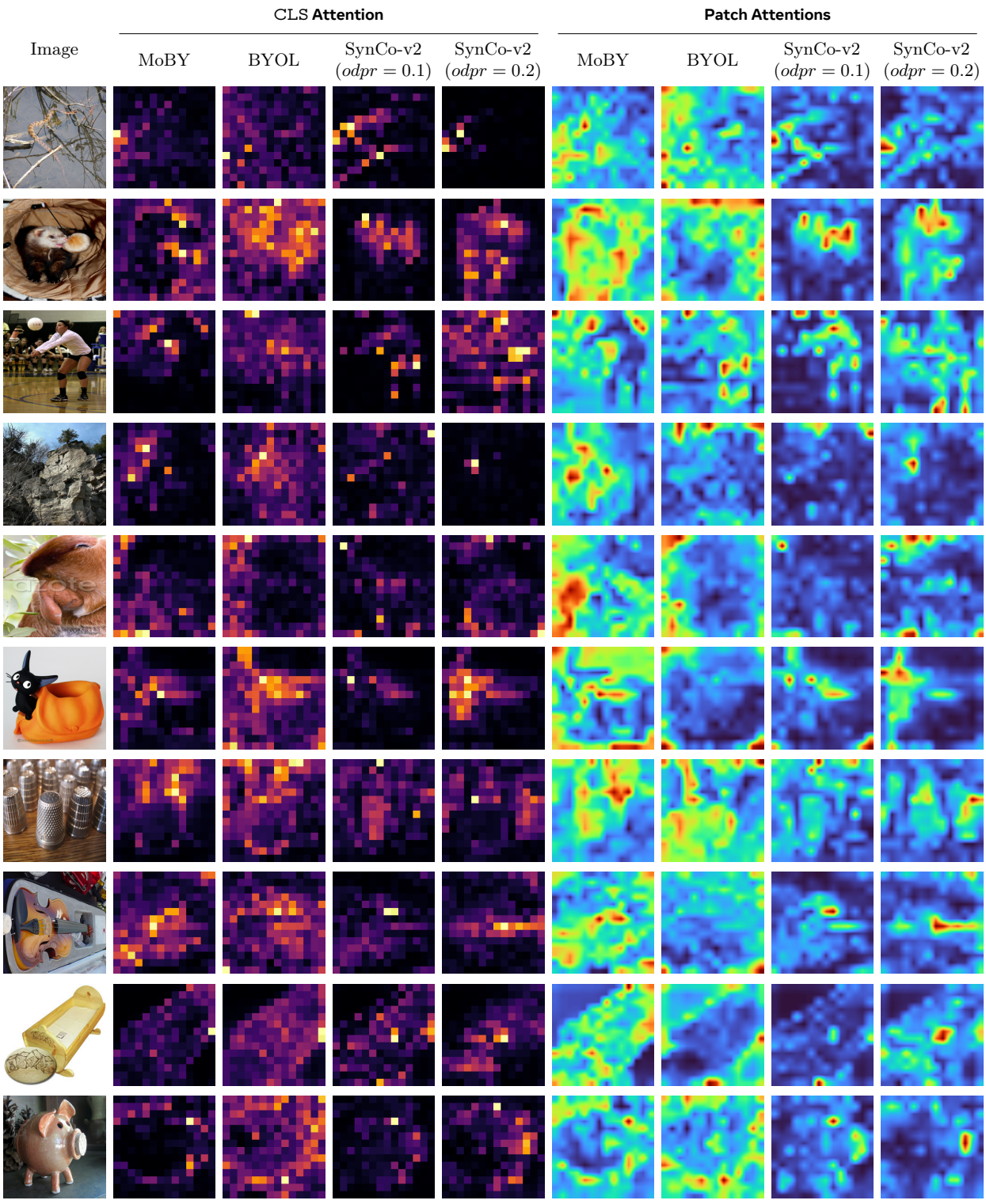


Figure 7 ViT-S/16 attention visualization. We show images 11 to 20.

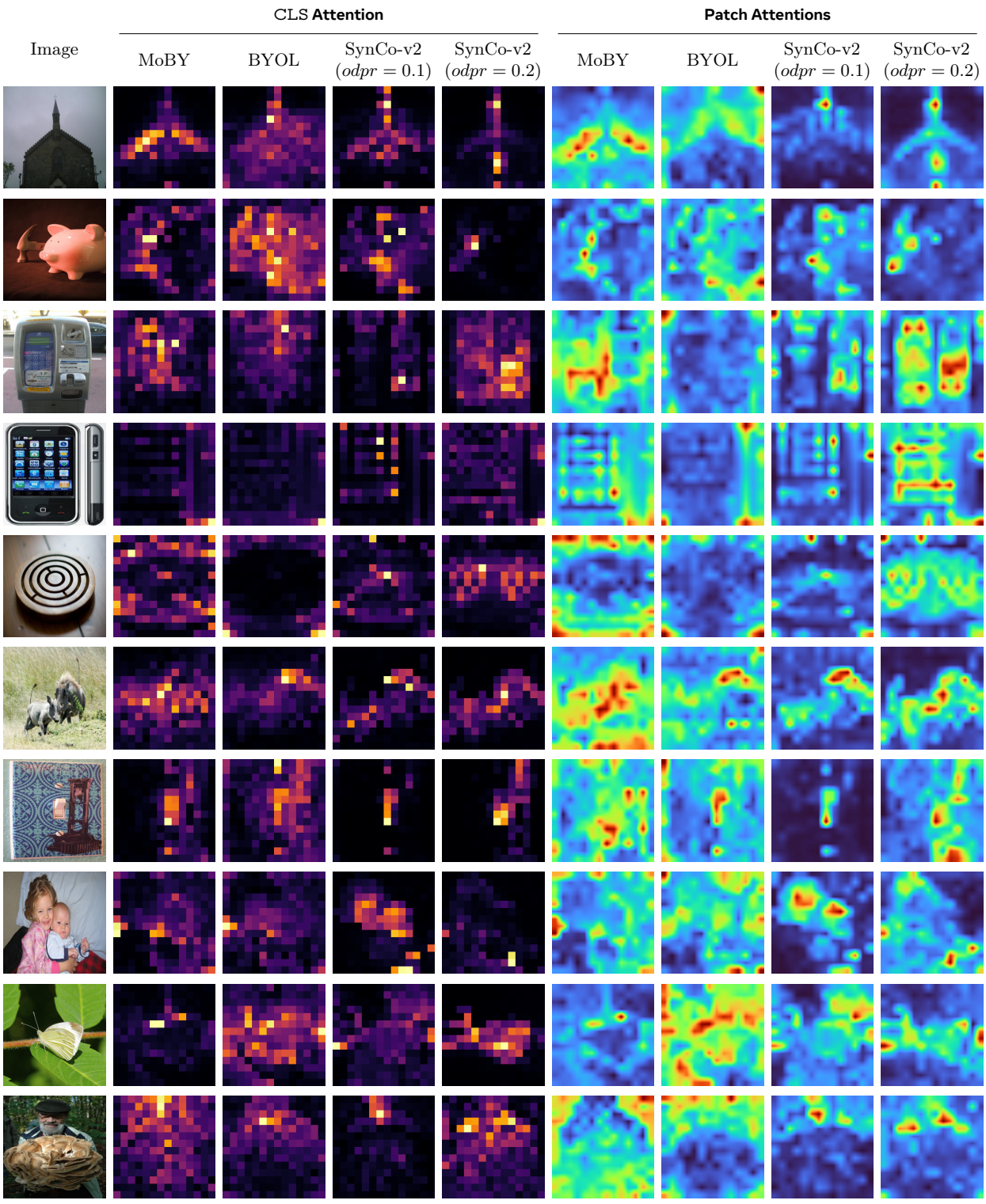


Figure 8 ViT-S/16 attention visualization. We show images 21 to 30.

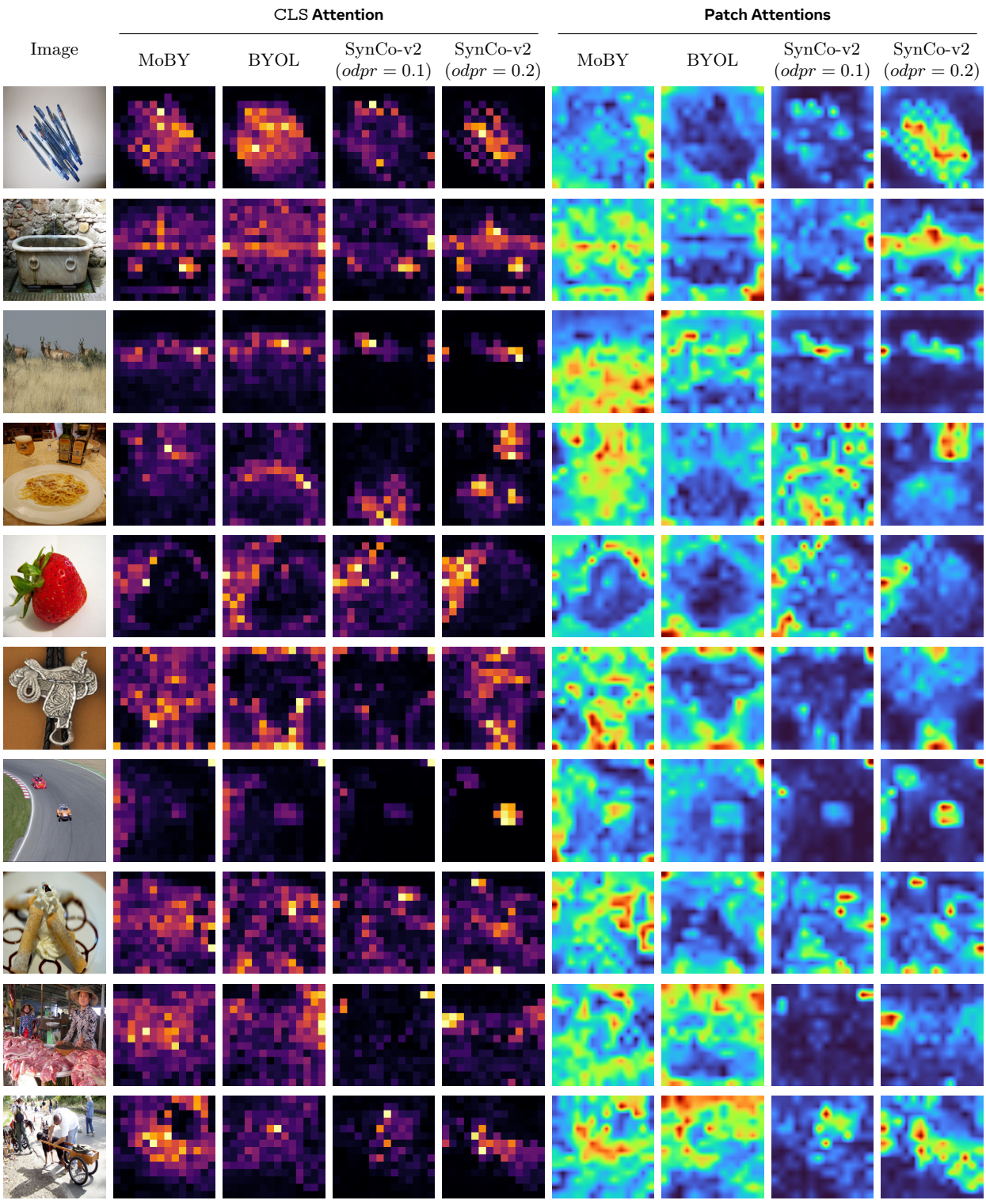


Figure 9 ViT-S/16 attention visualization. We show images 31 to 40.

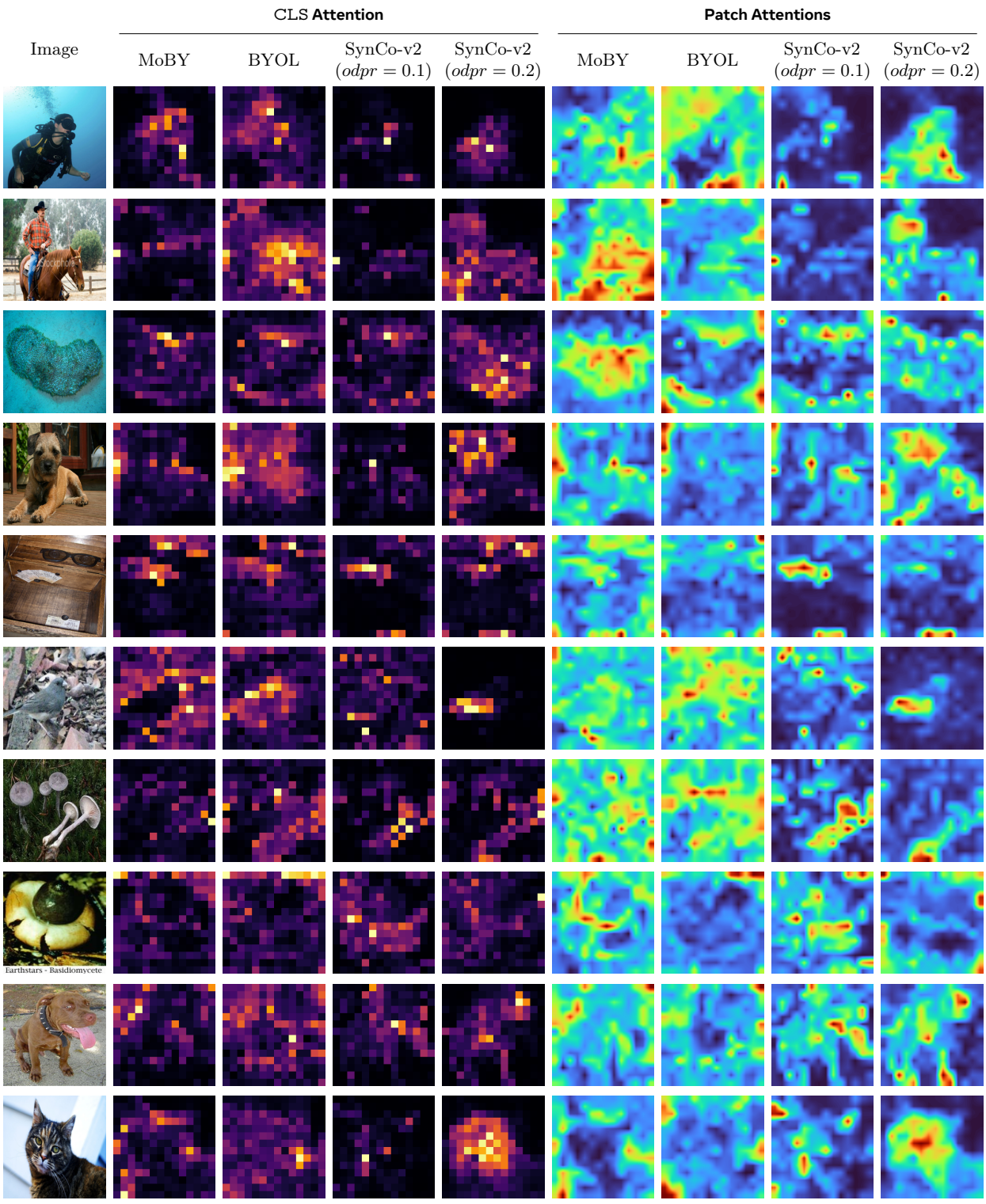


Figure 10 ViT-S/16 attention visualization. We show images 41 to 50.

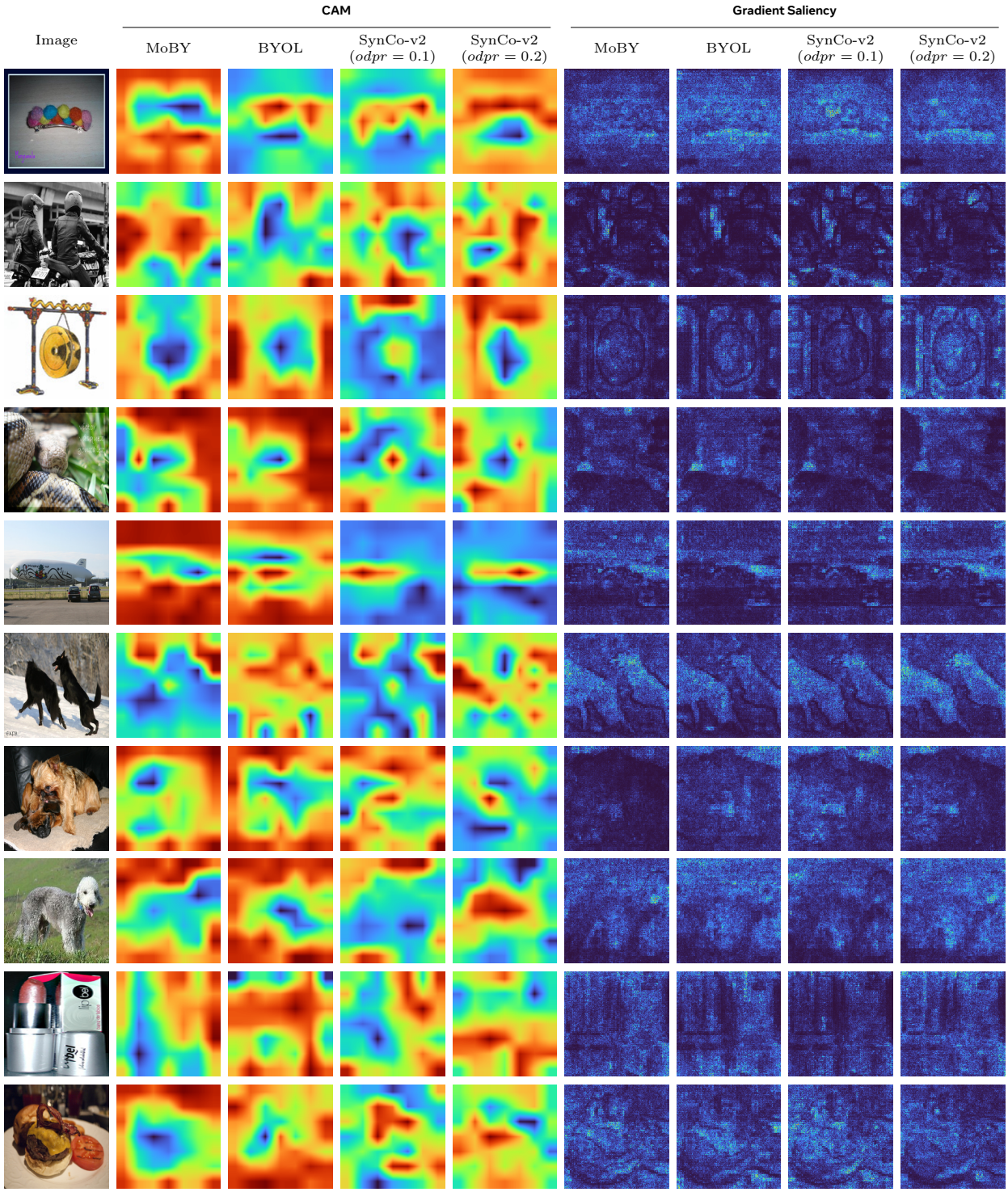


Figure 11 Swin-T/7 visualization. We show images 1-10

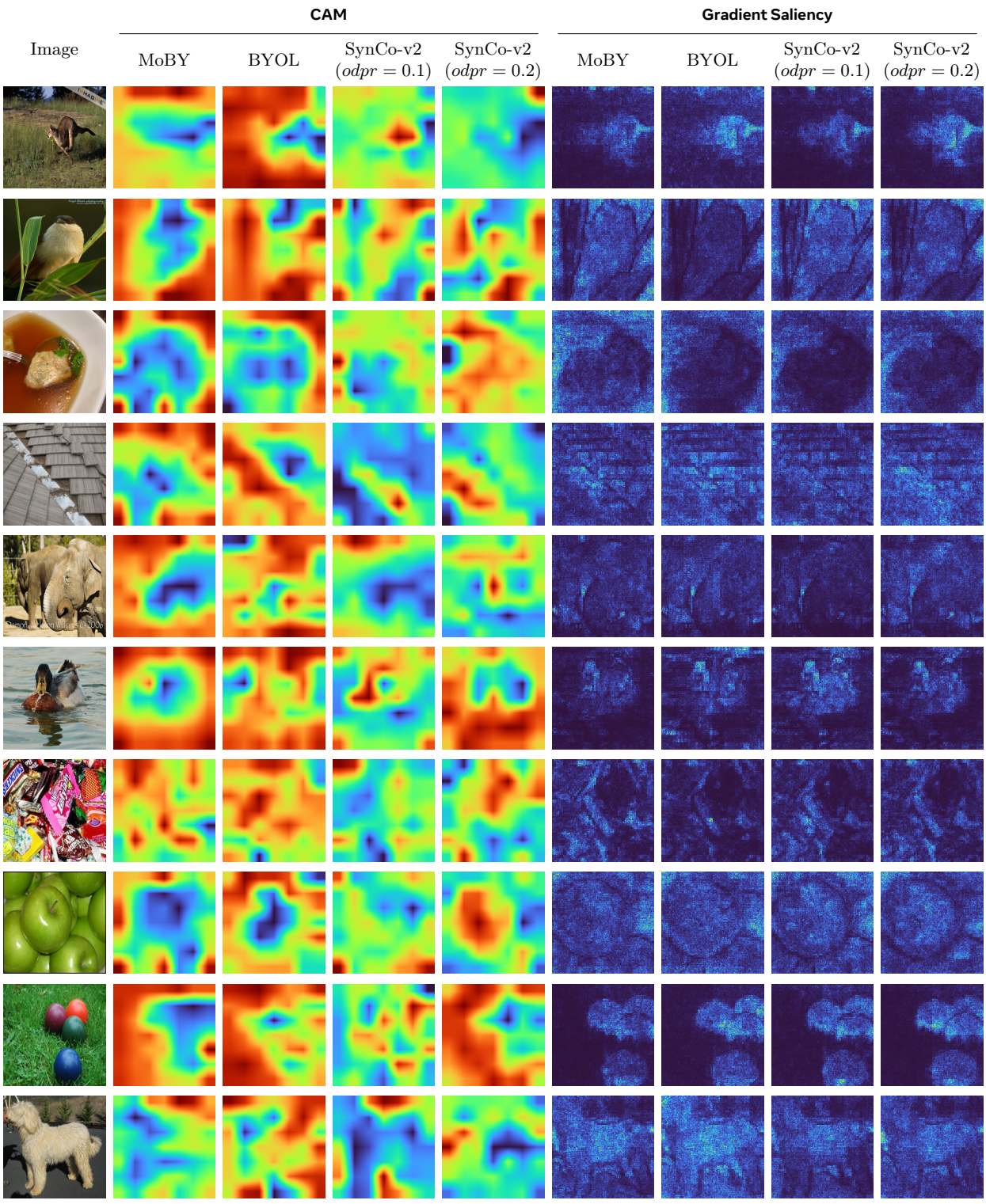


Figure 12 Swin-T/7 visualization. We show images 11-20

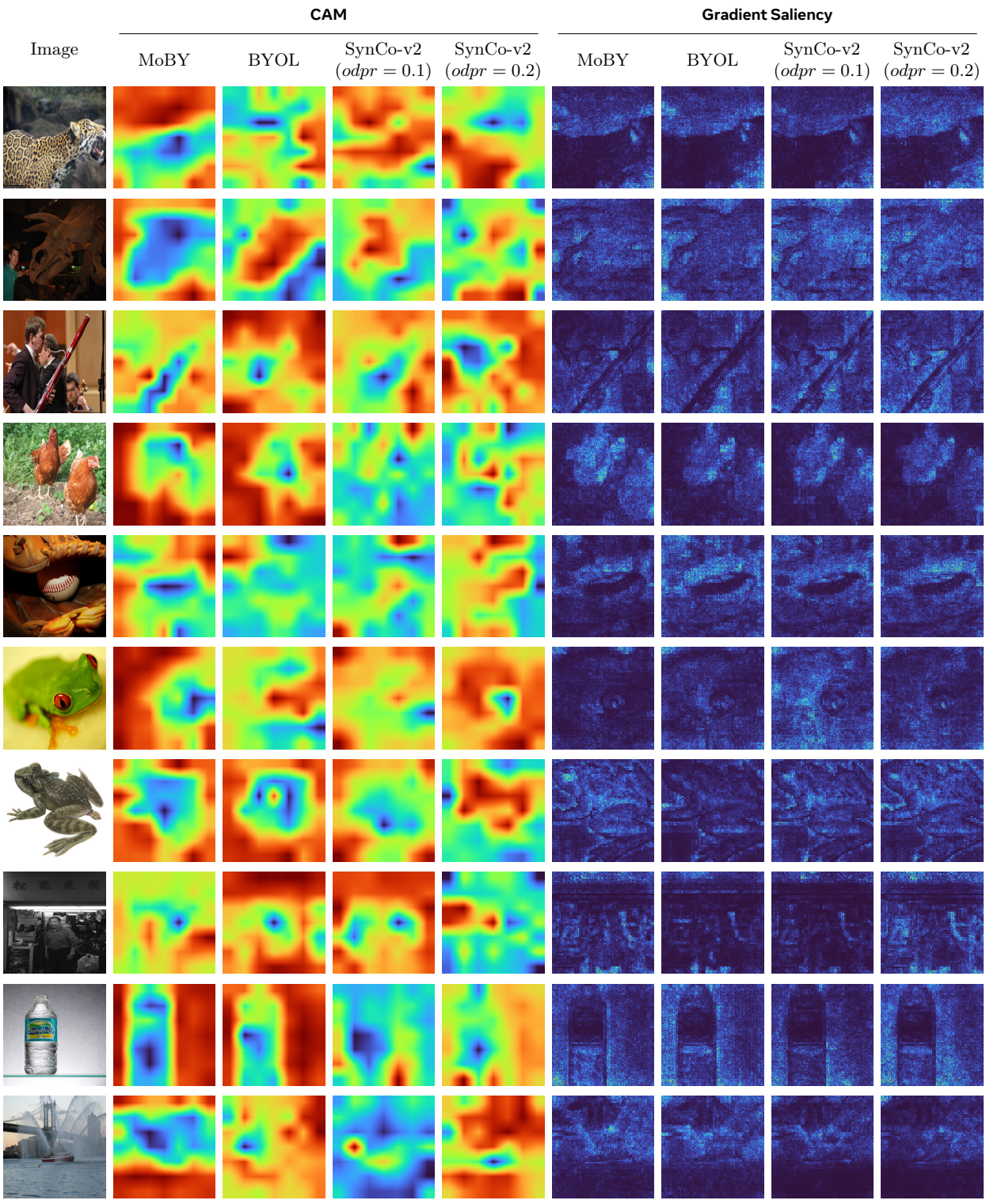


Figure 13 Swin-T/7 visualization. We show images 21-30

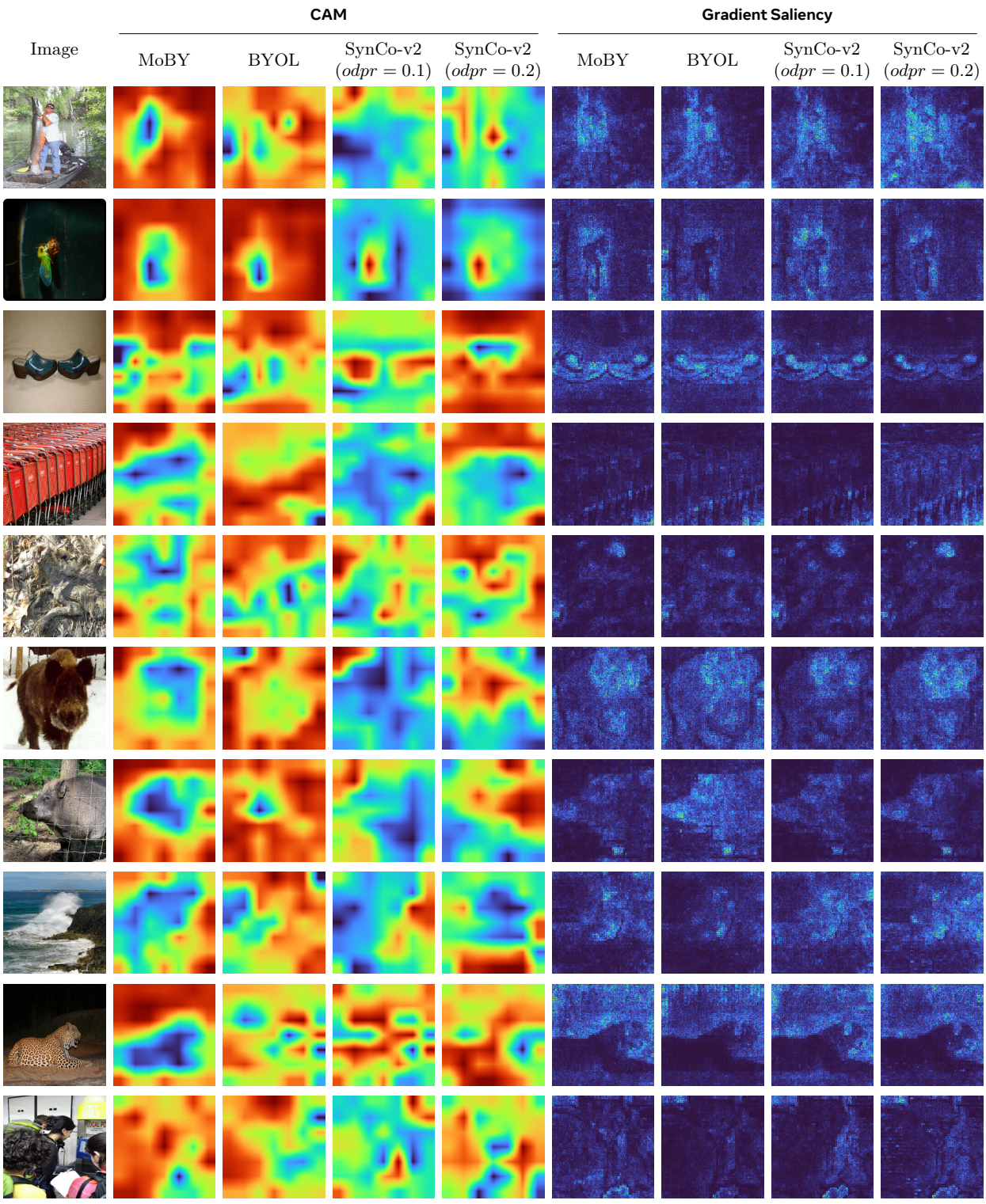


Figure 14 Swin-T/7 visualization. We show images 31-40

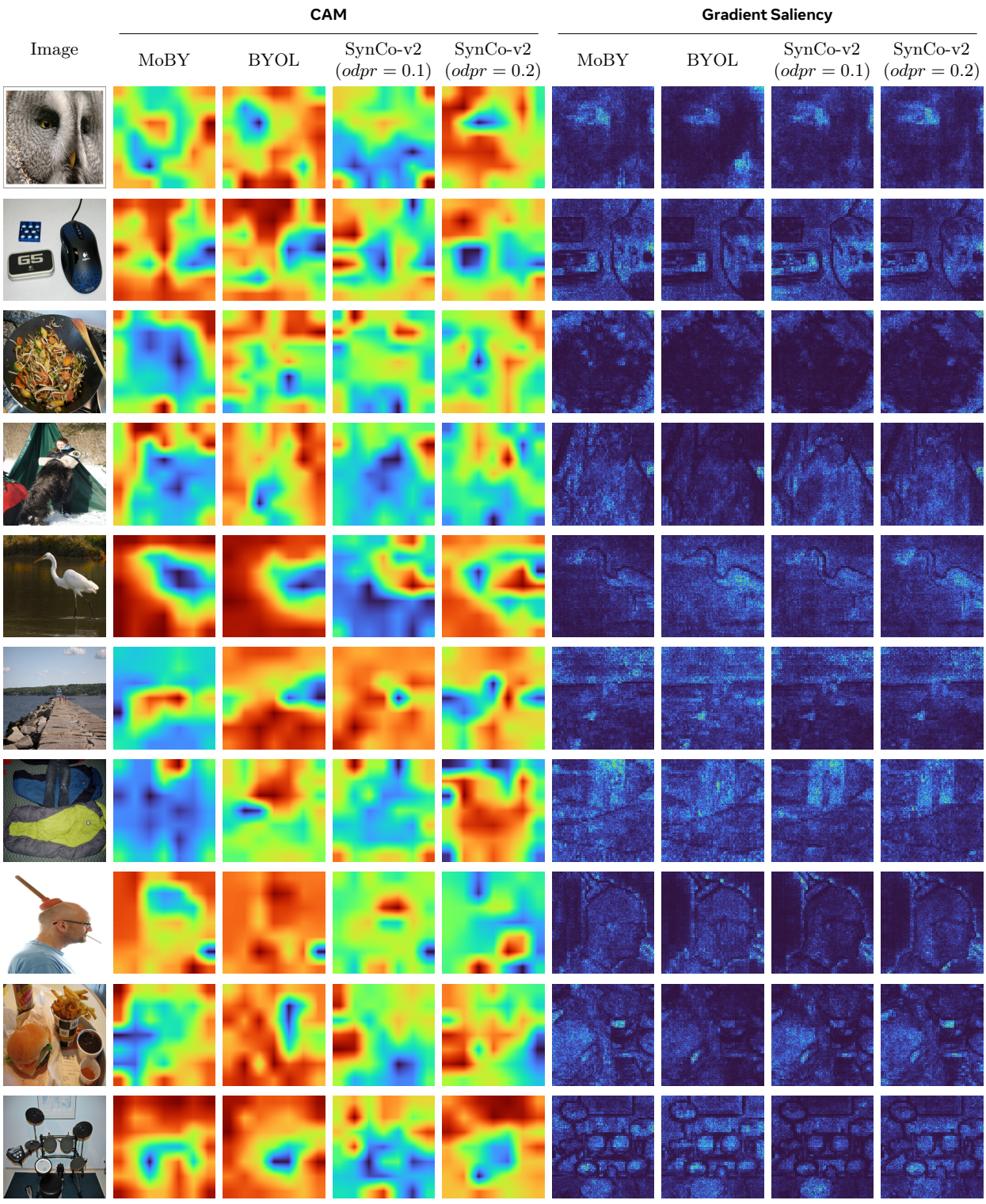


Figure 15 Swin-T/7 visualization. We show images 41-50