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

Supplementary Material

Contents

Dataset	Classes	Train	Test
ImageNet [21]	1000	1,281,167	50,000
ImageNet-100 [47]	100	128,116	5,000
CIFAR-10 [52]	10	50,000	10,000
CIFAR-100 [52]	100	50,000	10,000
Flowers-102 [64]	102	2,040	6,149
Oxford-IIIT Pets [67]	37	3,680	3,669
Food-101 [8]	101	75,750	25,250
Stanford Cars [51]	196	8,144	8,041
Caltech-101 [28]	101	3,060	6,085
DTD [19]	47	3,760	1,880
FGVC Aircraft [62]	100	6,667	3,333
SUN397 [89]	397	19,850	19,850
PASCAL VOC 2007 [26]	20	5,011	4,952

Table 9. **Image dataset characteristics.** This table lists the datasets used for pretraining and/or evaluation.

7. Implementation Details

We implement SynCo-v2 in PyTorch following the implementation of MoBY [90]. We re-implement **(i)** MoBY [90] (contrastive method with memory negatives), and **(ii)** BYOL [34] (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.

7.1. Datasets

Our experimental evaluation spans multiple visual recognition benchmarks with varying complexity and domain characteristics. The primary pretraining dataset is ImageNet ILSVRC-2012 [21], containing 1.28M training images across 1,000 object categories, while ImageNet-100 [47] (a 100-class subset with approximately 128K training images) is used for ablation studies. Transfer learning assessment encompasses eleven downstream classification tasks spanning different visual domains: CIFAR-10 and CIFAR-100 [52] contain natural images at low resolution (32×32); Oxford Flowers-102 [64] features fine-grained flower species; Oxford-IIIT Pets [67] includes cat and dog breeds; Food-101 [8] contains food dishes; Stanford Cars [51] includes fine-grained car models; Caltech-101 [28] covers diverse object categories; DTD [19] focuses on texture recognition; FGVC Aircraft [62] contains fine-grained aircraft variants; SUN397 [89] features scene recognition; and PASCAL VOC 2007 [26] includes multi-label object classification. Detailed statistics for all datasets are provided in Table 9.

7.2. Vision Transformer Architectures

Vision Transformer overview. We briefly describe the mechanism of the Vision Transformer (ViT) [24, 81] and refer to Vaswani *et al.* [81] for details about Transformers and to Dosovitskiy *et al.* [24] for its adaptation to images. We follow the implementation used in DeiT [80]. The ViT architecture takes as input a grid of non-overlapping contiguous image patches of resolution $N \times N$, typically using $N = 16$ (`/16`) in this work. The patches are passed through a linear layer to form a set of embeddings. An extra learnable token, referred to as the class token `[CLS]`, is added to the sequence to aggregate information from the entire sequence [24, 53], even though it is not attached to any label or supervision in our self-supervised setting. The projection head g_θ is attached at its output. The set of patch tokens and `[CLS]` token are fed to a standard transformer network with pre-norm layer normalization [5, 93]. The transformer consists of a sequence of self-attention and feed-forward layers with skip connections. The self-attention mechanism computes attention weights between all pairs of tokens, allowing each position to attend to all positions in the previous layer. This global receptive field is a key distinction from convolutional networks and enables the model to capture long-range dependencies effectively.

Swin Transformer overview. The Swin Transformer [58] introduces a hierarchical architecture with shifted window-based self-attention, addressing computational efficiency while maintaining the advantages of transformers. The window size used in this architecture is typically 7×7 (`/7`). Unlike ViT which uses global self-attention, Swin Transformer employs local window-based self-attention that limits computation to non-overlapping windows within the feature map. To enable connections across windows, it alternates between regular and shifted window partitioning across successive layers. The hierarchical structure progressively merges neighboring patches to create deeper feature maps at lower resolutions, similar to convolutional networks. This design reduces computational complexity from quadratic to linear with respect to image size while allowing for efficient modeling of both local and global relationships through the shifted window mechanism.

Architecture. For the DeiT-Small architecture [80], we use a patch size of 16×16 (`/16`), 12 transformer blocks with an embedding dimension of 384 and 6 attention heads, resulting in 22M parameters (denoted as DeiT-Small`/16` or DeiT-S). For the Swin-Tiny architecture [58], we use a hierarchical design with shifted windows of size 7×7 (`/7`), featuring 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-Tiny`/7` or Swin-T). The projection head g_θ consists of a 2-layer MLP with a hidden dimension of 4096, batch normalization [44], and ReLU activation [63], 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 [13, 34, 36]. Detailed specifications are provided in Table 10.

	DeiT-S`/16`	Swin-T`/7`
Blocks	12	[2,2,6,2]
Dim	384	[96,192,384,768]
Heads	6	[3,6,12,24]
Params	22M	28M
Throughput	940	755

Table 10. **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.

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

Table 11. **Parameters used to generate image augmentations.** This table lists the augmentation probabilities and intensity settings used for pretraining.

7.3. Image Augmentations

We use the same augmentation as used in BYOL [34]. 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 11.

7.4. Pretraining

We pretrain all models on ImageNet ILSVRC-2012 [21] for 300 epochs. We use the AdamW optimizer [60] 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 batch size of 512 split over 4 NVIDIA L40S GPUs (128 per GPU). We apply asymmetric drop path rates [43] of 0.2 for the *online* encoder and 0.0 for the *target* encoder.

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. 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 [30]. We adopt the hyperparameters $\alpha_k \in (0, 0.5)$, $\beta_k \in (1, 1.5)$, $\gamma_k \in (0, 1)$, $\sigma = 0.01$, $\delta = 0.01$, and $\eta = 0.01$ from Giakoumoglou and Stathaki [30], which were *empirically* determined based on the intuition of each method for convolutional networks and remain fixed throughout training. Additionally, Giakoumoglou and Stathaki [30] reported *no* significant differences in ablations over σ, δ, and η.

BYOL. For reproduction of BYOL [34], 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.

7.5. Linear Probing

For linear evaluation, we follow the standard linear evaluation protocol of [36, 49, 50], 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 DeiT-S and the final global average pooled features for Swin-T. 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 ImageNet 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. For all other downstream tasks, we fix the learning rate to 1.0 to ensure consistent evaluation across different datasets without extensive hyperparameter tuning. We apply label smoothing with $\epsilon = 0.1$ for improved generalization.

7.6. 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 [24, 49, 50]. 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.

8. Extended Results

In this section, we provide comprehensive evaluations of SynCo-v2 on k-NN classification, robustness benchmarks, adversarial attacks, and out-of-distribution detection tasks.

8.1. 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 12, SynCo-v2 consistently outperforms both BYOL and MoBY across all evaluated neighborhood sizes for both DeiT-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 DeiT-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.

8.2. 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

Method	$k = 10$	$k = 20$	$k = 100$	$k = 200$
		DeiT-S		
BYOL [34] (*repr.*)	62.53	62.43	60.49	59.12
MoBY [90] (*repr.*)	64.38	64.26	61.42	59.74
SynCo-v2 (ours)	**71.02**	**70.45**	**67.66**	**65.86**
		Swin-T		
BYOL [34] (*repr.*)	58.10	57.96	55.14	53.34
MoBY [90] (*repr.*)	67.84	67.14	63.68	61.82
SynCo-v2 (ours)	**69.28**	**68.72**	**65.24**	**63.17**

Table 12. **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**.

model reliability. For robustness evaluation, we employ four datasets: **ImageNet-v2** [70], 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) [38], 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) [40], containing naturally occurring adversarial examples that are difficult for ImageNet-trained models; and **ImageNet-Watermark** (IN-W) [55], 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) [82], containing 50,000 black-and-white sketch drawings of ImageNet objects; **ImageNet-R** (IN-R) [39], consisting of 30,000 artistic renditions across various styles (paintings, cartoons, sculptures, graffiti, embroidery) to test domain shift robustness; and **ImageNet-O** (IN-O) [40], 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 [39] 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 13, 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.

8.3. 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 [61], 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 [48]. Our evaluation includes: **FGSM** (Fast Gradient Sign Method) [32], a single-step gradient-based attack with $\epsilon = 8/255$; **PGD** (Projected Gradient Descent) [61], an iterative gradient-based attack with $\epsilon = 8/255$, step size $\alpha = 2/255$, and 10 iterations; **C&W** (Carlini & Wagner) [9], 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** [2], a score-based black-box attack using ℓ_∞ norm with 1,000 queries; **AutoAttack** [20], an ensemble of attacks that adaptively combines multiple methods using ℓ_∞ norm; **TIFGSM** (Translation-Invariant FGSM) [23], which incorporates translation invariance with $\epsilon = 8/255$, $\alpha = 2/255$, and 10 steps; and **OnePixel** [78], 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 14, 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.

8.4. Computational Efficiency Analysis

To assess practical applicability, we analyze computational requirements, including parameters, memory usage, and wall-clock training time. Table 15 reports detailed measurements for all methods. Due to computational limits, results come from single runs rather than multiple trials, and we omit standard deviations; training times may vary slightly from system factors. Our findings show that SynCo-v2 delivers strong performance with minimal computational cost. Synthetic negative generation uses simple transformations (interpolation, extrapolation, mixup, noise injection, and gradient-based perturbations) applied directly to memory-queue embeddings, adding negligible overhead relative to

Method	Epochs		Robustness					Out-Of-Distribution		
		IN	MF	T-0.7	TI	IN-C	IN-A	IN-W	IN-S	IN-R	IN-O
					DeiT-S						
BYOL [34] (repr.)	300	70.3	58.0	66.4	72.5	41.5	7.0	54.2	18.2	27.8	87.2
MoBY [90] (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 [34] (repr.)	300	68.5	56.5	64.7	70.6	40.4	6.9	52.8	17.7	27.1	**85.0**
MoBY [90] (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

Table 13. **Robustness and out-of-distribution evaluation of self-supervised methods.** Results show top-1 accuracy (in %) on various ImageNet variants using DeiT-S and Swin-T backbones, except ImageNet-O where FPR95 is reported. Best results are in **bold**.

Method	Epochs	Clean	FGSM	PGD	C&W	Square	Auto	TIFGSM	OnePixel
				DeiT-S					
BYOL [34] (repr.)	300	70.3	25.0	3.0	29.5	**22.4**	2.7	11.8	68.9
MoBY [90] (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 [34] (repr.)	300	68.5	**24.4**	**2.9**	28.8	21.9	2.6	11.5	67.1
MoBY [90] (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**

Table 14. **Adversarial robustness evaluation of self-supervised methods.** Results show top-1 accuracy (in %) under various attacks on ImageNet validation set using DeiT-S and Swin-T architectures. Best results are in **bold**.

backbone forward/backward passes. SynCo-v2 keeps the same parameter count as contrastive baselines, requiring no architectural changes or extra components. Memory overhead from storing synthetic negatives (768 samples) is small, and the on-the-fly generation removes the need for pre-computed datasets or external augmentation models.

9. Extended Related Work

In this section, we provide additional context in self-supervised learning, contrastive methods, and synthetic negative generation to complement the discussion in the main paper.

> *"If AI is a cake, self-supervised learning is the bulk of the cake."* — Yann LeCun [54]

9.1. Self-supervised Learning

Self-supervised learning has evolved through multiple paradigms, each with distinct approaches to learning visual representations. Early pretext task methods [31, 65, 96] defined auxiliary supervised tasks to learn features, but were limited by task-specific biases. *Joint embedding methods* [11, 13, 34, 36] 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 [12] introduced the idea of treating images as sequential data for autoregressive prediction, which was subsequently followed by BEiT [6] and BEiT-v2 [68], transferring BERT-style [22] masked prediction methodologies to computer vision. MAE [37] demonstrated that extensive masking of image patches (reaching 75%) establishes an effective self-supervised objective, while SimMIM [92] streamlined the methodology with lightweight prediction architectures. Subsequent improvements include: MaskFeat [86] which predicted HOG features rather than raw pixels, Context Autoencoder [18] which exploited contextual relationships, and MSN [3] which integrated masking with siamese network architectures. Contemporary research has emphasized efficiency and effectiveness through methodologies including SiamMAE [35], MixMAE [56], PixMIM [57], and TinyMIM [71]. While recent developments include Crop-

Method	Total Params	Trainable Params	Avg Mem ↓ (MB)	Time/ Epoch ↓ (min)	100 Ep. ↓ (h)	300 Ep. ↓ (h)
			ViT-Small			
BYOL [34]	50.7M	26.3M (51.93%)	21,316	31.60	52.66	157.97
MoBY [90]	50.7M	26.3M (51.93%)	21,325	32.71	54.52	163.55
SynCo-v2 (ours)	50.7M	26.3M (51.93%)	21,325	31.60	52.66	157.99
			Swin-Tiny			
BYOL [34]	65.6M	33.8M (51.61%)	29,167	39.72	66.19	198.58
MoBY [90]	65.6M	33.8M (51.61%)	29,177	39.78	66.31	198.93
SynCo-v2 (ours)	65.6M	33.8M (51.61%)	29,176	38.75	64.58	193.75

Table 15. **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 ↓).

MAE [27] featuring efficient siamese cropped autoencoder designs and ColorMAE [41] 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 [37], joint embedding approaches often surpass them in linear probing tasks and exhibit stronger emergent properties such as unsupervised semantic segmentation [11, 66].

9.2. Contrastive Learning

Contrastive learning has progressed from early instance discrimination methods [88] to sophisticated frameworks employing momentum encoders [16, 36], large batch sizes [13], and advanced augmentation strategies [14]. Key innovations include the introduction of momentum-based memory banks to increase the number of negative samples [36], the demonstration that very large batch sizes can replace memory banks [13], and the development of asymmetric network designs that avoid collapse without explicit negative samples [15, 34]. More recent works have explored nearest-neighbor contrastive learning [25], representation decorrelation [95], relative location prediction [79], and region-level contrastive learning [91]. Despite these advances, the role of negative sample quality versus quantity remains an active research question [1, 72], with ongoing debates about whether contrastive methods can match the performance of generative approaches without extensive computational resources.

9.3. Hard Negative Mining

Hard negative mining has a rich history in computer vision, originally developed for object detection [29, 75] and later adapted for metric learning [73, 77]. In contrastive learning, various strategies have emerged to identify and leverage hard negatives: selecting negatives with high similarity to the anchor [72], using nearest neighbors in the embedding space [25], employing debiased sampling to address false negatives [94], and generating synthetic hard negatives through mixing [46] or adversarial perturbations [42]. MoCHI [46] introduced hard negative mixing for contrastive learning in convolutional networks, while DCL [94] explored debiased contrastive learning to mitigate the impact of false negatives. SynCo [30] 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.

9.4. Self-supervised Vision Transformers

The adaptation of transformers to computer vision [24, 58, 80] has enabled new self-supervised learning approaches. DINO [11] demonstrated that self-distillation with vision transformers produces strong emergent properties in attention mechanisms, while MoCo-v3 [17] and MoBY [90] 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 [37], BEiT [6], and SimMIM [92] 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 [97] which integrates masked image modeling with self-distillation, and I-JEPA [4] 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 [66], SEER [33], and V-JEPA [7].

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 [10], or massive computational scale, making it an accessible and effective alternative for practitioners with limited resources.

9.5. Emergent Properties

A particularly intriguing aspect of self-supervised vision transformers is their emergent semantic segmentation capabilities without explicit supervision. DINO [11] 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 [66] demonstrated that these properties strengthen with scale, achieving impressive zero-shot segmentation performance. Recent work has explored the mechanisms behind these emergent properties: TokenCut [85] leverages self-attention for unsupervised image segmentation, LOST [76] localizes objects in images without supervision using ViT features, and FreeSOLO [84] 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.*

10. Discussion

In this section, we discuss the computational efficiency and design principles of SynCo-v2. Our findings challenge the prevailing assumption that negative samples are secondary in self-supervised vision transformers, demonstrating their critical role in achieving competitive performance and offering a simpler alternative to dominant generative methods.

10.1. Computational Efficiency and Overhead

While synthetic negative generation introduces minimal computational overhead during training, the benefits substantially outweigh the costs. The memory overhead of storing synthetic negatives is negligible ($\mathcal{O}(|\mathcal{S}| \cdot d)$ where $|\mathcal{S}| = 768 \ll K = 4096$), as we generate only 768 synthetic negatives compared to 4096 memory bank negatives. The generation process adds ***negligible*** additional training time, as it primarily involves computing similarity scores to identify hard negatives and applying the *six* transformation strategies—operations that require minimal computation compared to forward passes through the encoder. The efficient implementation ensures that our method remains practical for large-scale pretraining while delivering consistent performance gains.

10.2. 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 [30]. 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 [30].

10.3. Strong and Consistent Gains

Our method demonstrates substantial improvements across all evaluation metrics, achieving state-of-the-art results among methods with comparable architectural complexity and training budgets. As shown in Table 1, SynCo-v2 achieves 73.1% top-1 accuracy on ImageNet linear evaluation with DeiT-S, representing a *significant* **+0.8%** improvement over the MoBY baseline and a *remarkable* **+2.8%** gain over BYOL. With Swin-T, we obtain 75.4% accuracy, *outperforming* MoBY by **+0.7%** and BYOL by **+6.9%**. These gains are particularly noteworthy given that they are achieved through a simple modification to the contrastive learning framework—generating synthetic hard negatives *"on-the-fly"*—without requiring architectural changes, complex training tricks, or the multi-crop augmentation strategies employed by methods like DINO. Furthermore, the consistency of improvements across two distinct transformer architectures (DeiT and Swin), multiple downstream tasks (Tables 2 and 3), and emergent properties (Figure 3) demonstrates that synthetic hard negatives provide genuine representational benefits rather than task-specific overfitting. The additional computational cost is minimal, as synthetic negative generation requires only lightweight transformations on existing embeddings, making the cost-benefit ratio highly favorable for practitioners seeking improved performance without substantial infrastructure investment.

10.4. 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 [11] and DINO-v2 [66] 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 (MoBY) and self-distillation without negatives (BYOL). This suggests that synthetic hard negatives provide a complementary mechanism for improving attention quality that could potentially be combined with DINO-style training procedures. Future work exploring the integration of synthetic hard negatives with multi-crop augmentation, longer training schedules, and other attention-enhancing techniques could yield even stronger emergent properties while maintaining the simplicity of contrastive learning.

10.5. Revisiting Negative Samples

Our findings challenge the prevailing assumption that negative samples are less important than other design choices in self-supervised learning for vision transformers. The success of DINO [11] and its variants has led many researchers to focus on self-distillation approaches that avoid negatives entirely, potentially overlooking the untapped potential of improved negative sampling strategies. Our work demonstrates that contrastive learning with high-quality negatives can achieve competitive performance and strong emergent properties, offering a simpler alternative to methods requiring complex training tricks (centering, sharpening, multi-crop augmentation).

> *This suggests that the community should reconsider contrastive learning as an approach that excels in generalizing on previously unseen data, developing emergent abilities, and achieving improved representation quality, compared to dominant generative approaches [6, 37].*

10.6. 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 [69], ALIGN [45]) 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 [66]) or projection head designs could yield additional improvements. Finally, extending our analysis to other transformer variants (*e.g.*, Swin-V2 [59], CvT [87], PVT [83]) would validate the generality of our findings across diverse architectural designs.

11. Limitations

In this section, we discuss the limitations of our method, including constraints on model scale, training duration, and hyperparameter tuning..

11.1. Architectural Scale

Due to limited computational resources, our experiments were constrained to smaller architectures (DeiT-Small with 22M parameters and Swin-Tiny with 28M parameters). We acknowledge that evaluating our approach on larger models such as DeiT-Base/ViT-Base (86M parameters), DeiT-Large/ViT-Large (304M parameters), and Swin-Base (88M parameters) would provide valuable insights into scalability and performance gains at increased model capacity. However, these larger architectures require > 8 GPUs to maintain the respective batch sizes (even for batch size 512), which was beyond our available computational budget. We believe the current architectures are sufficient to demonstrate the effectiveness of synthetic hard negatives for vision transformers, particularly given that: **(i)** we show consistent improvements across two distinct transformer architectures (DeiT and Swin), suggesting the approach generalizes across different design paradigms, and **(ii)** the method achieves meaningful gains over strong baselines with the same computational constraints.

11.2. Training Budget

Similarly, our experiments were limited to 300-epoch 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 [11] and DINO-v2 [66]) and higher resolutions could reveal whether synthetic hard negatives provide additional benefits with extended training budgets.

11.3. Hyperparameter Tuning

We did not perform extensive hyperparameter tuning or ablations for all possible combinations of synthetic negative strategies and their parameters due to computational constraints. Instead, we adopted most hyperparameters from SynCo [30] (which was designed for convolutional networks) and made minimal adaptations based on empirical observations with vision transformers. In particular, we also adopted the cooldown strategy from SynCo [30], where synthetic hard negatives are disabled in the final 100 epochs to stabilize training and prevent overfitting to increasingly difficult negatives. Given that our method already achieves substantial performance gains over the baseline with these empirically set hyperparameters, and considering the minimal sensitivity observed across different configurations, we found *no compelling* need for exhaustive tuning. While we demonstrate in the *main paper* that using all six synthetic negative strategies yields the best performance, a more systematic exploration of strategy combinations, the number of synthetic negatives per strategy, hard negative selection thresholds, and transformation parameters could potentially uncover marginal improvements. However, the *consistent gains* achieved with our current empirical configuration suggest that the method is reasonably robust to hyperparameter choices, making it practical for adoption without extensive tuning.

12. Attention Visualization

In this section, we provide extensive attention visualizations to demonstrate the emergent semantic segmentation properties of vision transformers.

12.1. Vision Transformer

For DeiT-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 [11]. 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.

12.2. 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) [74], and **(ii)** Gradient Saliency. For CAM visualization [74], 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: $\text{Saliency} = \left|\frac{\partial y^c}{\partial \mathbf{x}}\right|$, 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 DeiT and Swin, we observe consistent patterns where SynCo-v2 produces more focused and semantically meaningful gradient-based attention maps.

13. 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.

14. Checkpoint Availability

The pre-trained model checkpoints for models trained on the ImageNet ILSVRC-2012 dataset are available for download: DeiT (top-1 linear evaluation accuracy 73.1%) and Swin (top-1 linear evaluation accuracy 75.4%).

References

[1] Adnan Ali, Jinlong Li, Huanhuan Chen, and Ali Kashif Bashir. From overfitting to robustness: Quantity, quality, and variety oriented negative sample selection in graph contrastive learning, 2024. 6

[2] 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. 4

[3] 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. 5

[4] 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. 6

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 2

[6] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. 5, 6, 8

[7] 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, 2024. 6

[8] 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. 1

[9] 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. 4

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

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 5, 6, 7, 8, 9

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 5

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 2, 5, 6

[14] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners, 2020. 6

[15] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 6

[16] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020. 6

[17] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021. 6

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

[19] 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. 1

[20] 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. 4

[21] 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. 1, 2

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. 5

[23] 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. 4

[24] 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. 2, 3, 6

[25] 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. 6

[26] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2009. 1

[27] Alexandre Eymaël, Renaud Vandeghen, Anthony Cioppa, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Efficient image pre-training with siamese cropped masked autoencoders, 2024. 6

[28] 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. Special issue on Generative Model Based Vision. 1

[29] 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, 2010. 6

[30] Nikolaos Giakoumoglou and Tania Stathaki. Synco: Synthetic hard negatives for contrastive visual representation learning, 2025. 3, 6, 7, 9

[31] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. 5

[32] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014. 4

[33] 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. 6

[34] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 1, 2, 3, 4, 5, 6, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[35] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders, 2023. 5

[36] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. 2, 3, 5, 6

[37] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 5, 6, 8

[38] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 4

[39] 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*, 2021. 4

[40] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 4

[41] Carlos Hinojosa, Shuming Liu, and Bernard Ghanem. Colormae: Exploring data-independent masking strategies in masked autoencoders, 2024. 6

[42] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples, 2020. 6

[43] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. 3

[44] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 2

[45] 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. 8

[46] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020. 6

[47] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021. 1

[48] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 4

[49] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning, 2019. 3

[50] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019. 3

[51] 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. 1

[52] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 1

[53] 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. 2

[54] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 5

[55] 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. 4

[56] Jihao Liu, Xin Huang, Jinliang Zheng, Yu Liu, and Hongsheng Li. Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers, 2023. 5

[57] Yuan Liu, Songyang Zhang, Jiacheng Chen, Kai Chen, and Dahua Lin. Pixmim: Rethinking pixel reconstruction in masked image modeling, 2023. 5

[58] 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. 2, 6

[59] 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. 8

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 2

[61] 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. 4

[62] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013. 1

[63] Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. pages 807–814, 2010. 2

[64] 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. 1

[65] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2016. 5

[66] 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. 6, 7, 8

[67] 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. 1

[68] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers, 2022. 5

[69] 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. 8

[70] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019. 4

[71] Sucheng Ren, Fangyun Wei, Zheng Zhang, and Han Hu. Tinymim: An empirical study of distilling mim pre-trained models, 2023. 5

[72] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. 6

[73] 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, 2015. 6

[74] 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, 2019. 9

[75] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining, 2016. 6

[76] 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. 7

[77] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 6

[78] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. pages 828–841. IEEE, 2019. 4

[79] 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. 6

[80] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021. 2, 6

[81] 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. 2

[82] 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. 4

[83] 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. 8

[84] 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. 7

[85] 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. 7

[86] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training, 2023. 5

[87] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021. 8

[88] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018. 6

[89] 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. 1

[90] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers, 2021. 1, 4, 5, 6, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[91] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning, 2021. 6

[92] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022. 5, 6

[93] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. 2

[94] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022. 6

[95] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. 6

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

[97] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022. 6

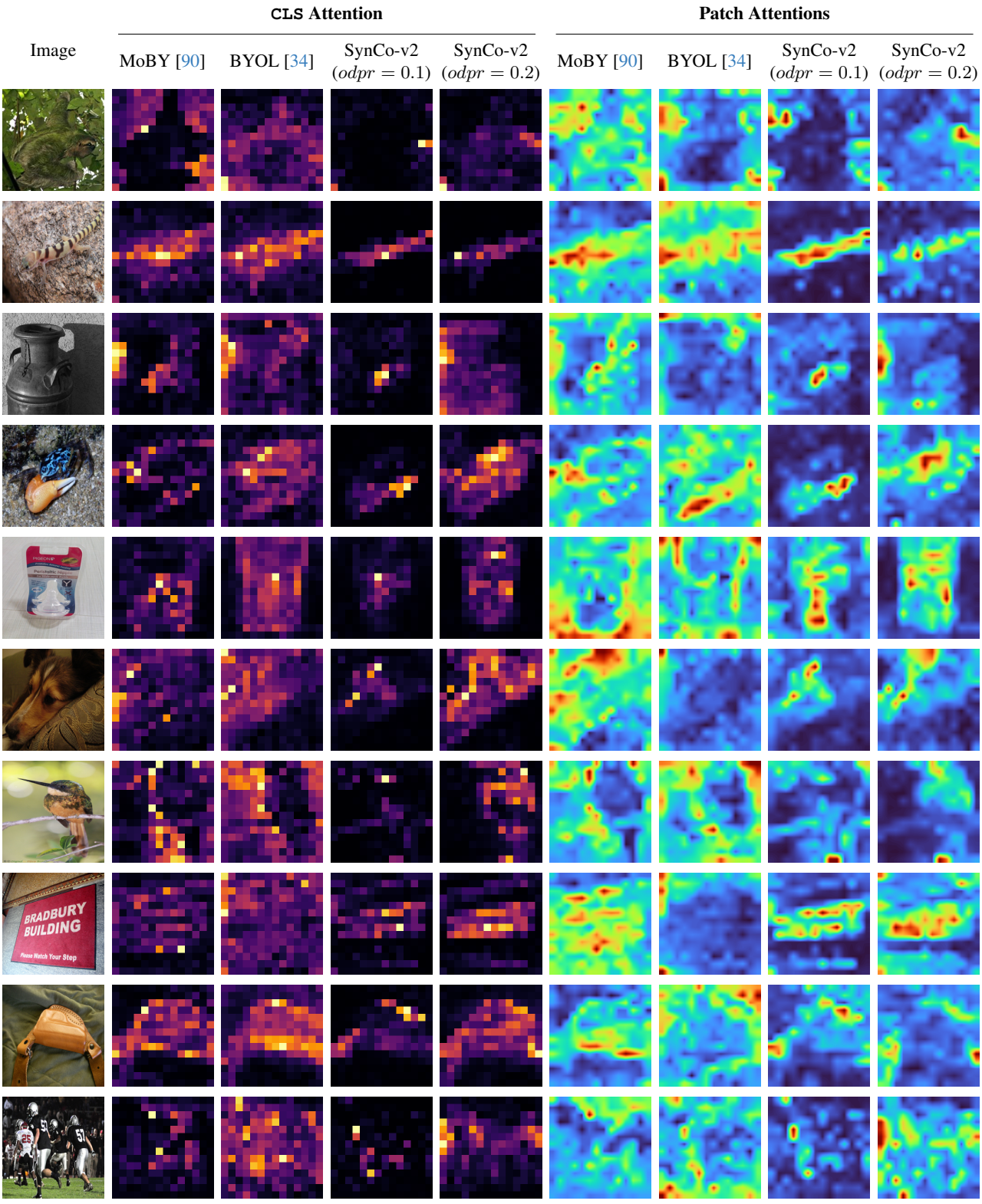

Figure 6. **DeiT-S/16 attention visualization.** We show images 1 to 10.

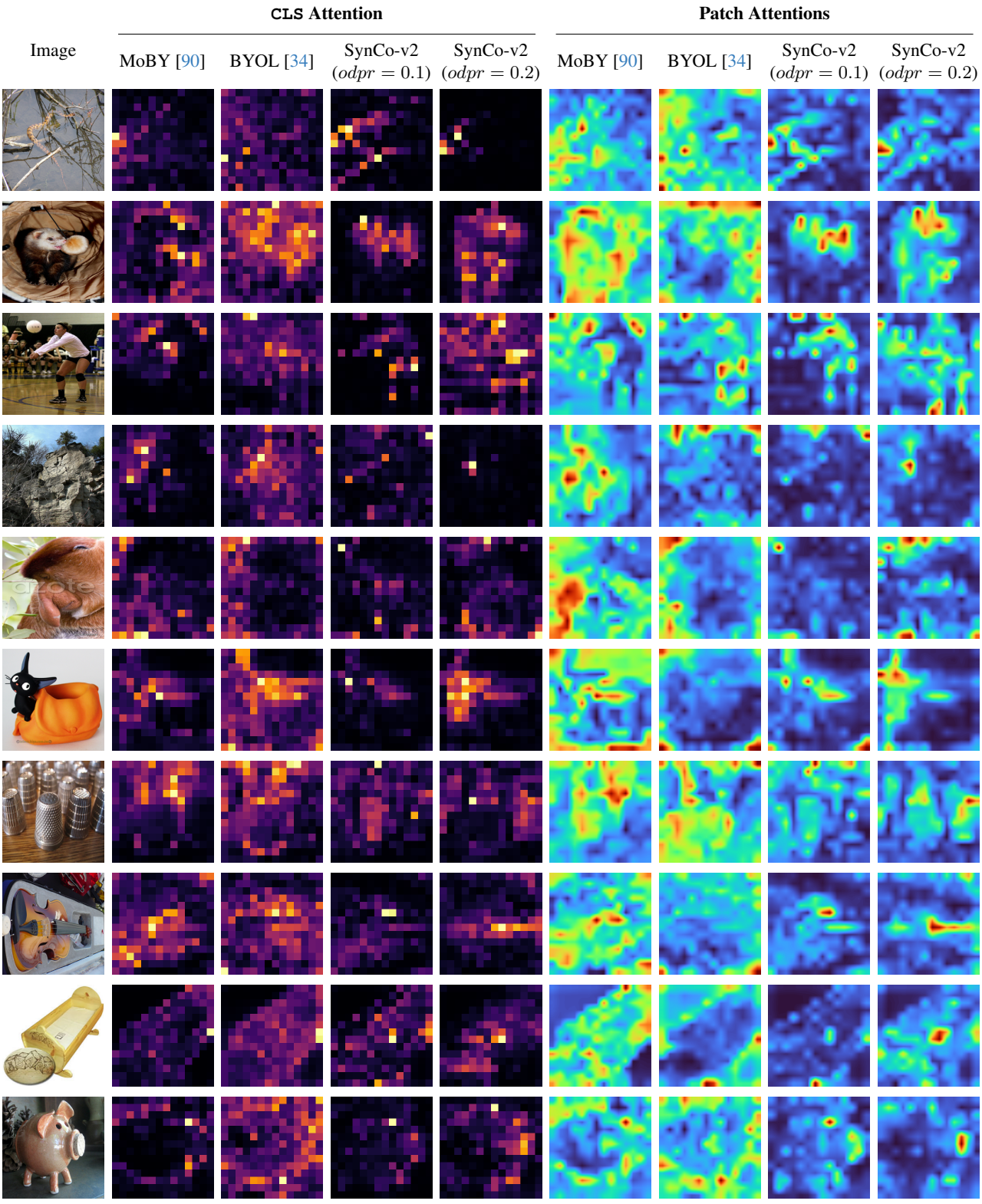

Figure 7. **DeiT-S/16 attention visualization.** We show images 11 to 20.

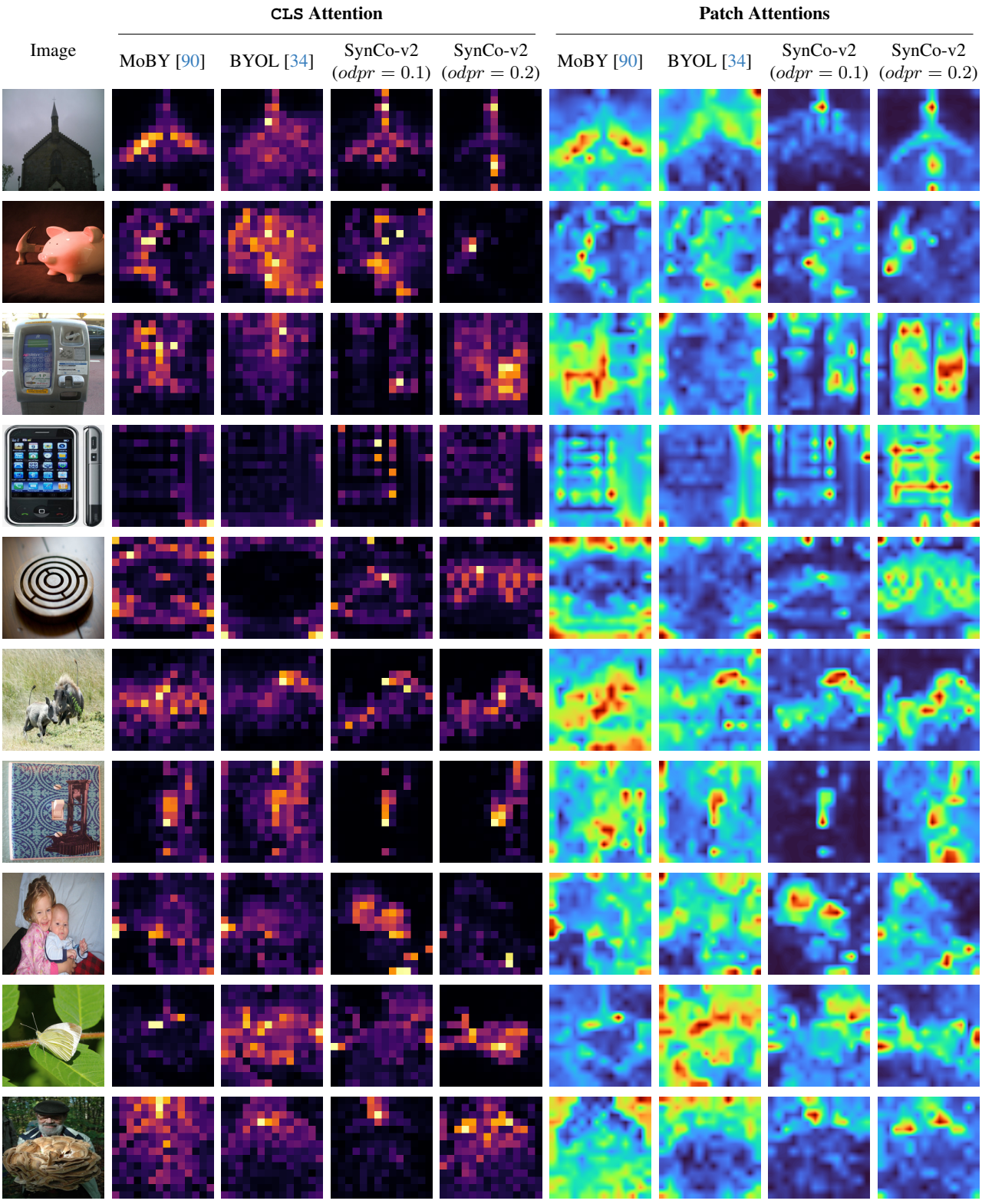

Figure 8. **DeiT-S/16 attention visualization.** We show images 21 to 30.

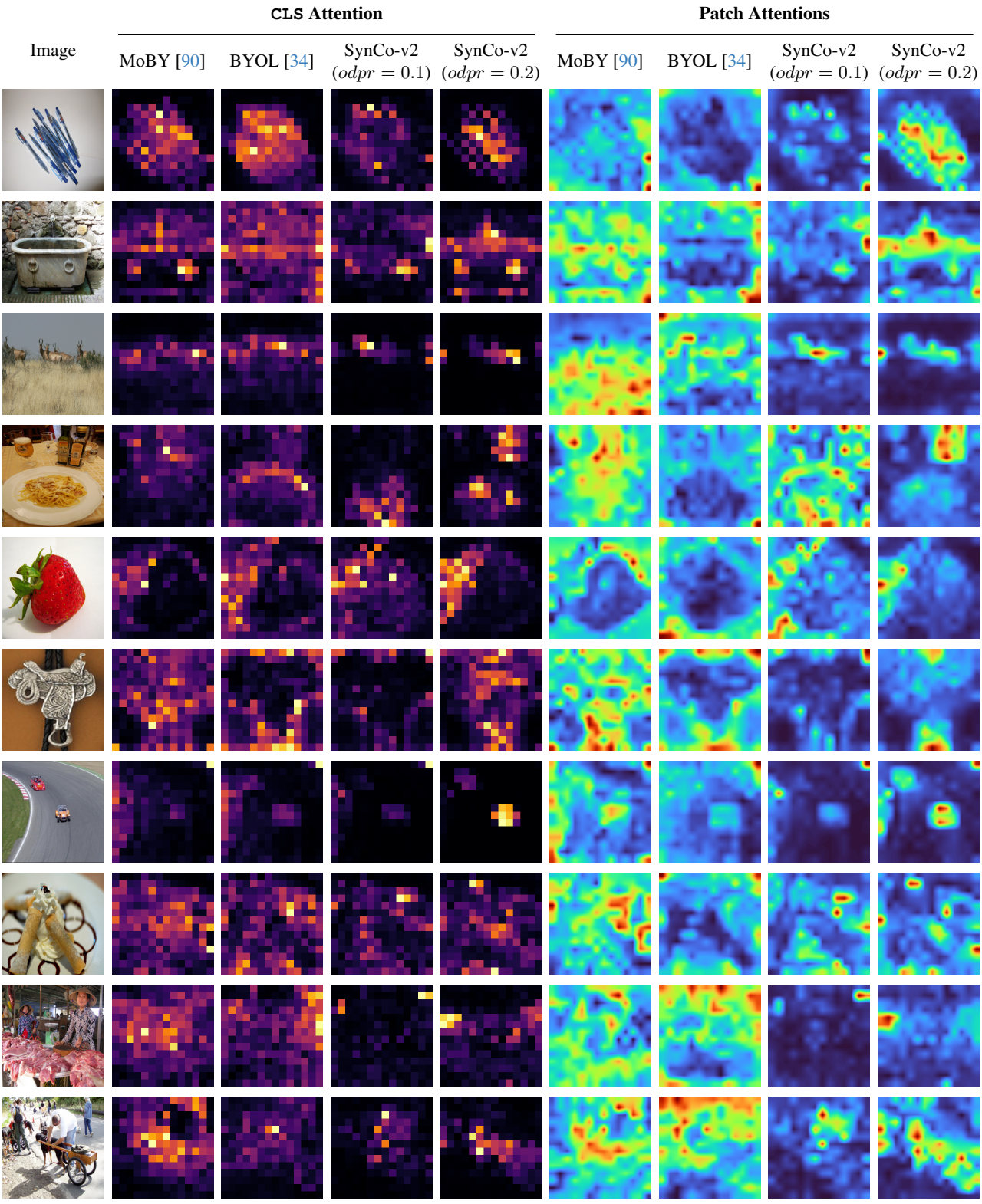

Figure 9. **DeiT-S/16 attention visualization.** We show images 31 to 40.

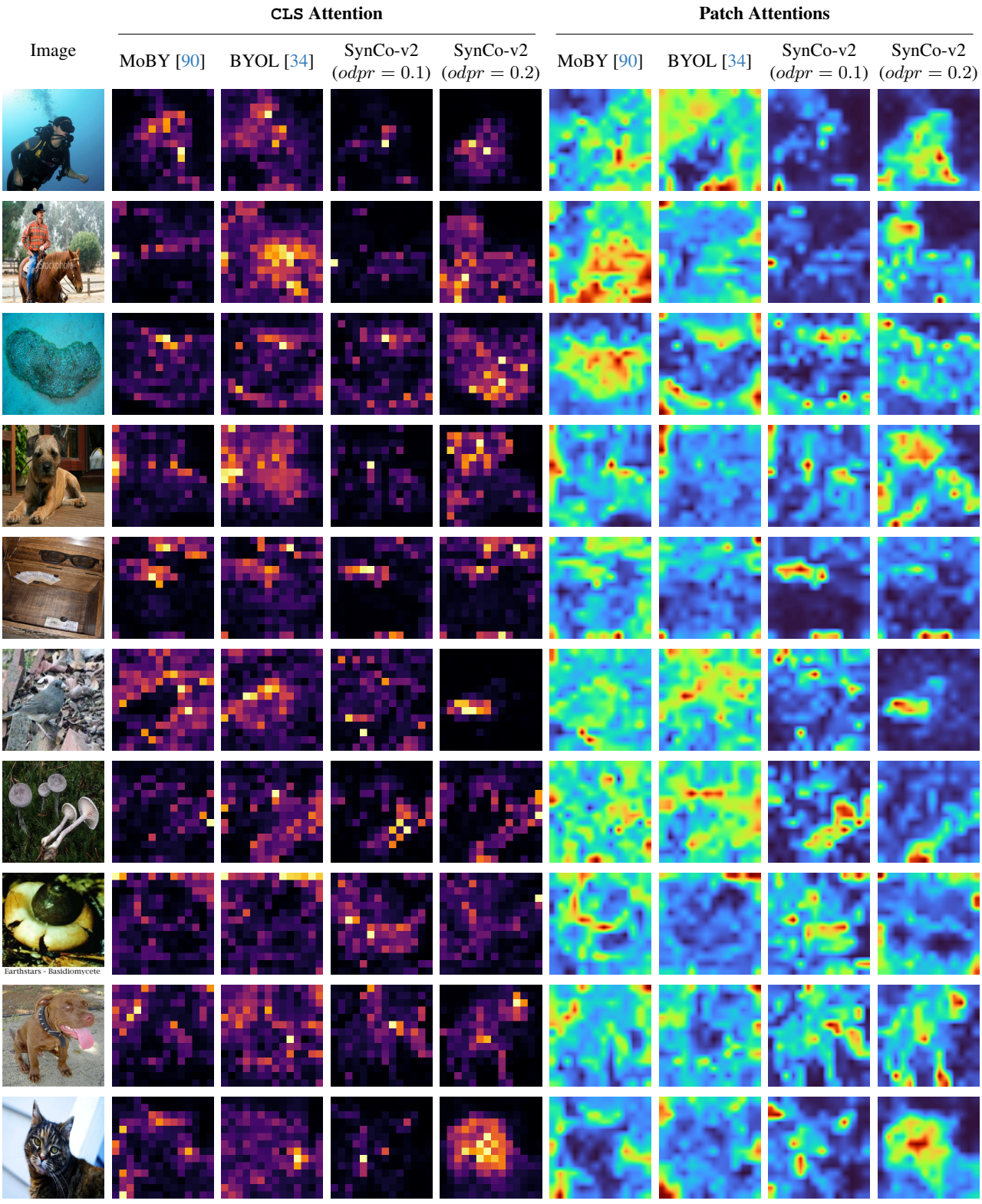

Figure 10. **DeiT-S/16 attention visualization.** We show images 41 to 50.

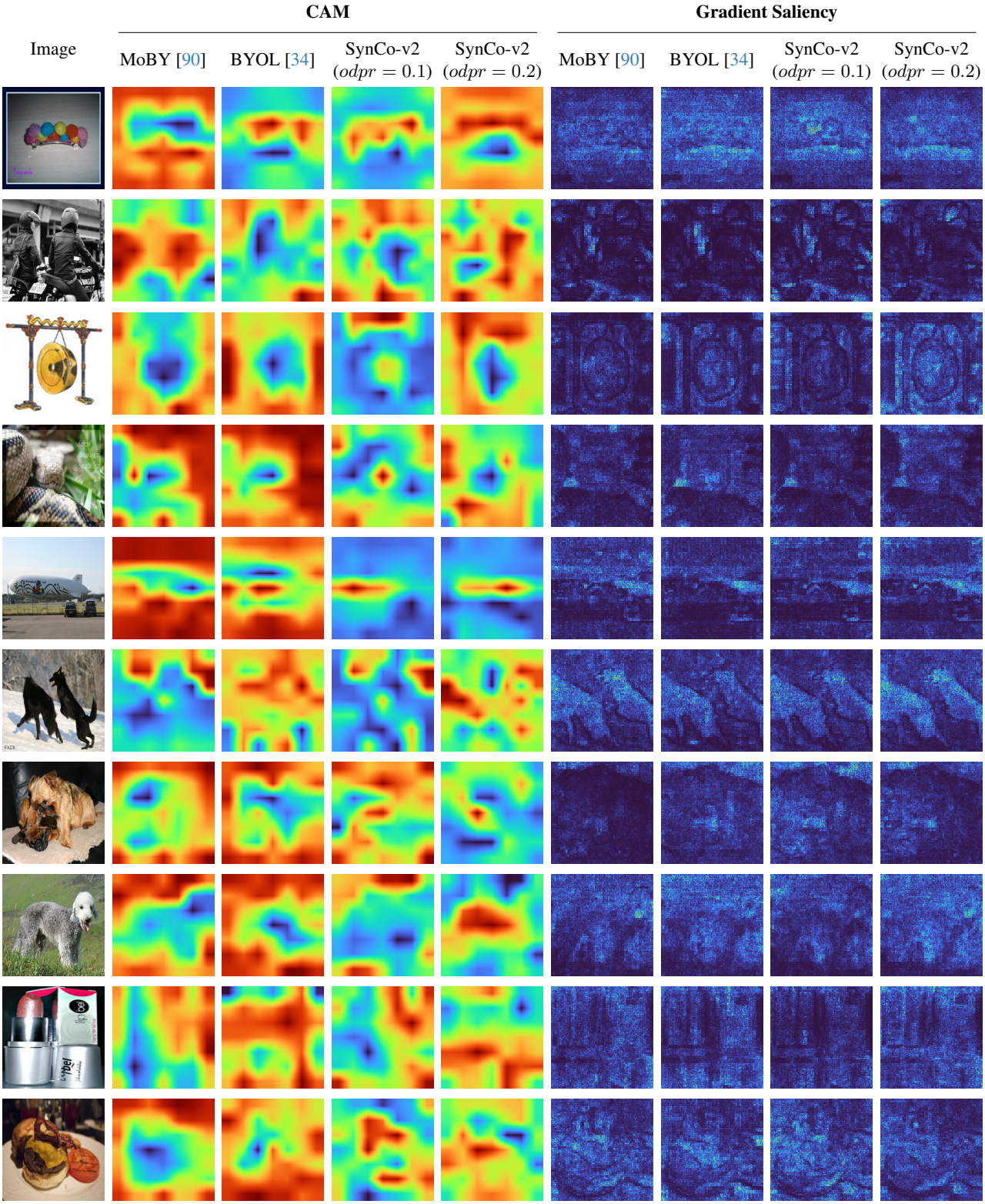

Figure 11. **Swin-T/7 visualization**. We show images 1-10

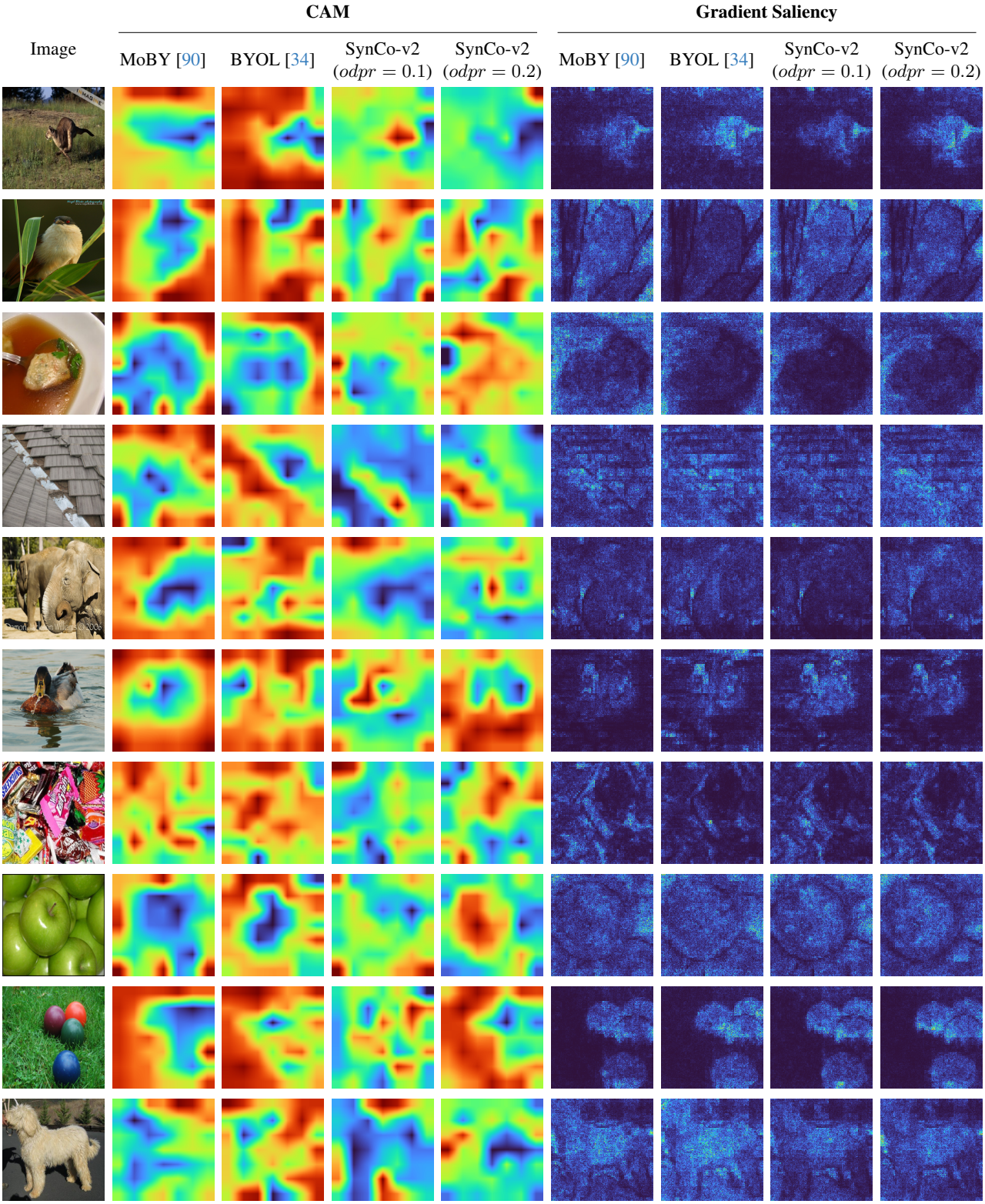

Figure 12. **Swin-T/7 visualization**. We show images 11-20

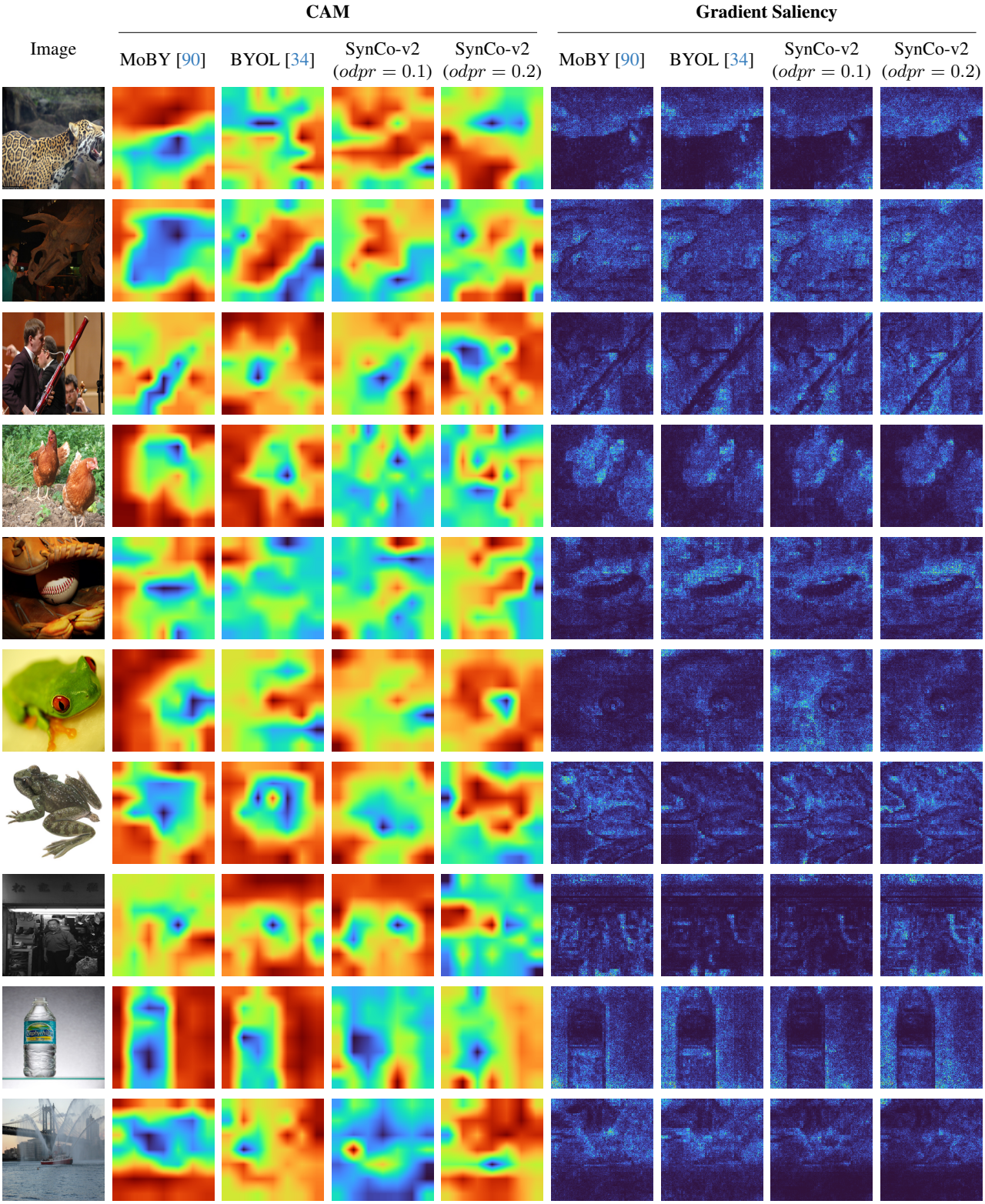

Figure 13. **Swin-T/7 visualization**. We show images 21-30

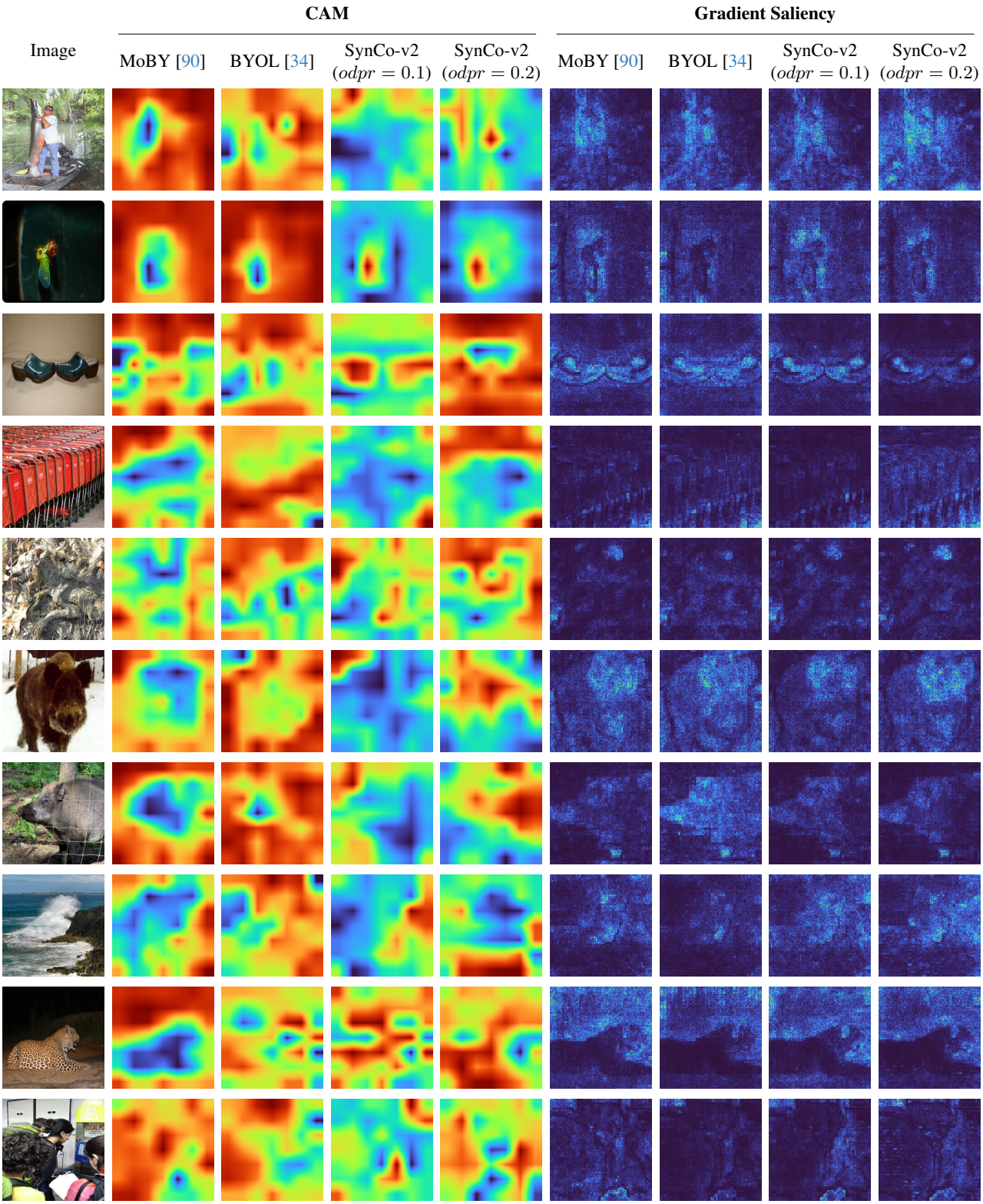

Figure 14. **Swin-T/7 visualization**. We show images 31-40

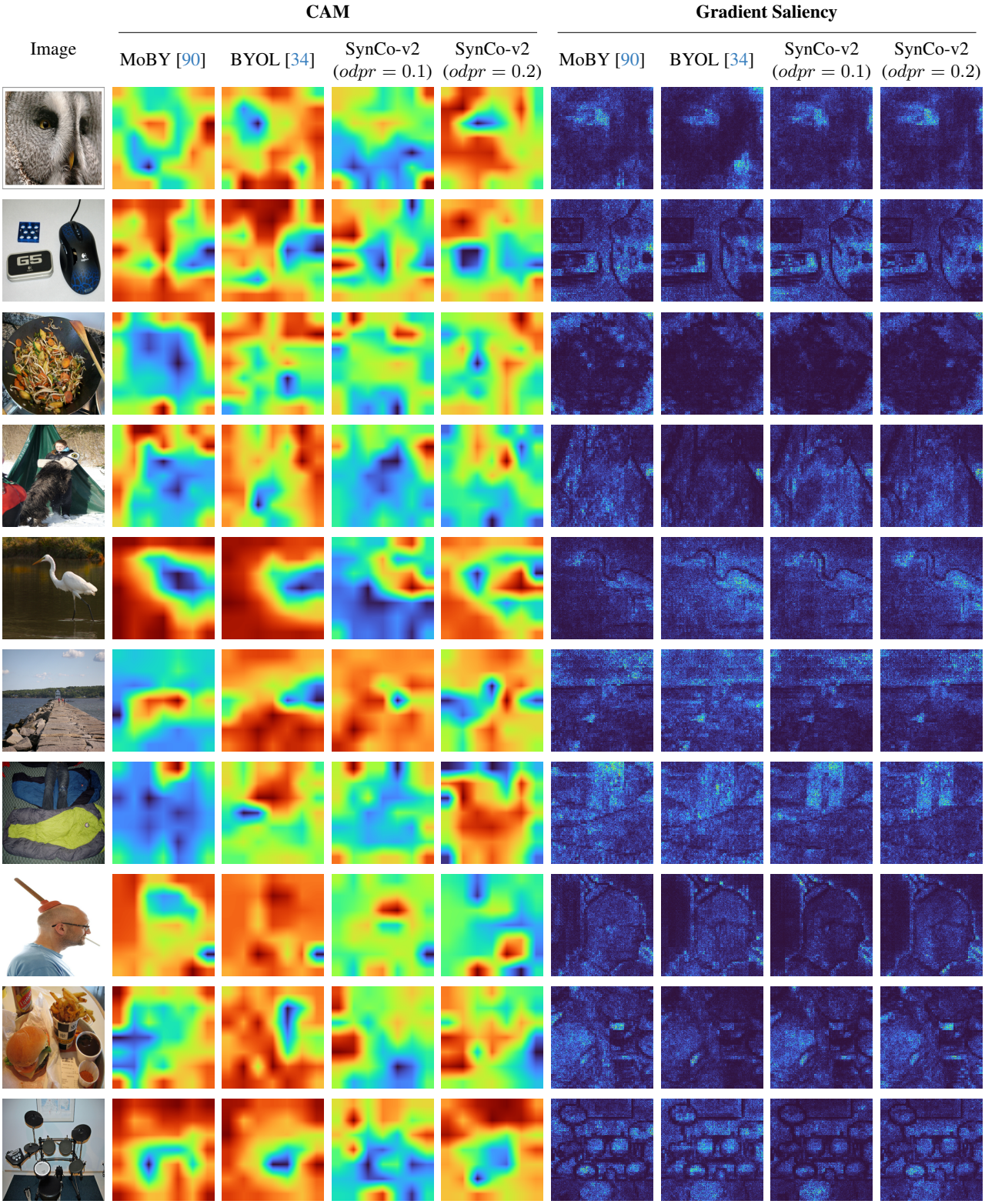

Figure 15. **Swin-T/7 visualization**. We show images 41-50