CASTing Your Model: Learning to Localize Improves Self-Supervised Representations

Ramprasaath R. Selvaraju[1*] Karan Desai[2*] Justin Johnson[2] Nikhil Naik[1]
[1]Salesforce Research, [2]University of Michigan
{rselvaraju,nnaik}@salesforce.com {kdexd,justincj}@umich.edu

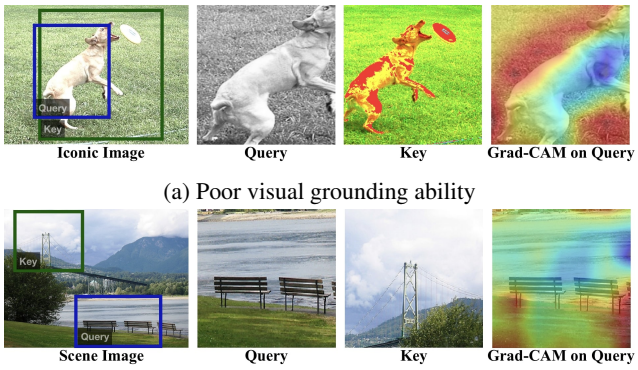
Abstract

Recent advances in self-supervised learning (SSL) have largely closed the gap with supervised ImageNet pretraining. Despite their success these methods have been primarily applied to unlabeled ImageNet images, and show marginal gains when trained on larger sets of uncurated images. We hypothesize that current SSL methods perform best on iconic images, and struggle on complex scene images with many objects. Analyzing contrastive SSL methods shows that they have poor visual grounding and receive poor supervisory signal when trained on scene images. We propose Contrastive Attention-Supervised Tuning (CAST) to overcome these limitations. CAST uses unsupervised saliency maps to intelligently sample crops, and to provide grounding supervision via a Grad-CAM attention loss. Experiments on COCO show that CAST significantly improves the features learned by SSL methods on scene images, and further experiments show that CAST-trained models are more robust to changes in backgrounds. Our code is available at https://github.com/salesforce/CAST/.

1. Introduction

Self-supervised learning (SSL) of visual feature representations has seen great interest in recent years. SSL in computer vision aims to learn feature representations without using any human annotations, which can be utilized by downstream tasks such as supervised image classification [1, 2], object detection [3, 4], and semantic segmentation [5, 6]. Recent SSL methods based on contrastive learning [7, 8] have begun to match or even outperform supervised pretraining on several downstream tasks [9–14].

The promise of self-supervised methods is that they ought to allow us to learn better features by scaling to ever-larger training sets, without the need for expensive human-provided labels. Unfortunately, the success of recent SSL methods has been largely confined to unlabeled images from the ImageNet [2] training set. Naïvely applying them to larger uncurated sets of internet images has shown marginal gains [11, 12, 14] despite using image sets that are orders of magnitude larger than ImageNet (eg. Instagram-1B [15], YFCC100M [16], JFT-300M [17]).



(a) Poor visual grounding ability

(b) Sampling issues with complex images

Figure 1: We identify two issues with recent contrastive approaches to self-supervised learning: (a) **Poor grounding:** On iconic images, contrastive methods can match key and query but use the wrong image regions to do so. Grad-CAM [22] reveals that the model puts high weight (red) on background regions, and low weight (blue) on the object of interest. (b) **Inconsistent Samples:** On complex images, randomly sampled crops may portray different objects, giving an inconsistent learning signal. We show that correcting these issues improves self-supervised learning.

We hypothesize that current SSL methods perform best when trained on *iconic images* of single objects (like those in ImageNet) but struggle when trained on more complex *scene images* with many objects. Indeed, current SSL methods struggle even when trained on curated datasets of scene images [18, 19] such as COCO [20] or Places205 [21].

In this paper, we analyze contrastive self-supervised models to understand the cause of these limitations and propose a solution to overcome them. Specifically, we find that existing contrastive self-supervised models have poor visual grounding ability and they receive imperfect supervisory signal when augmented views contain different visual concepts, which is common in images of complex scenes.

These issues may arise from the practice of training the

*Equal Contribution

instance discrimination task with random views from images. This practice does not encourage semantic understanding, and models often cheat by exploiting low-level visual cues or spurious background correlations. For example, in Figure 1a, the model relies on the grass to match the two augmented views of the dog. Augmented views for training these models commonly start with taking random crops from an image. This strategy may be acceptable for iconic images. However, for scene images, like those in COCO, two views may contain semantically distinct objects (such as the crops in Figure 1b). This fact may explain diminishing improvements of contrastive SSL models trained on varied web images, and the reduction in their performance when trained with scene images alone.

To mitigate these limitations, we propose Contrastive Attention-Supervised Tuning (CAST), a training method to improve the visual grounding ability of contrastive SSL methods. CAST consists of two algorithmic components: (a) an intelligent geometric transform for cropping different views from an input image, based on constraints derived from an unsupervised saliency map, and (b) a Grad-CAM [22]-based attention loss that provides explicit grounding supervision by forcing the model to attend to objects that are common across the crops.

We train the Momentum Contrastive Encoder (MoCo) [12], a leading contrastive learning method, using CAST on the COCO dataset. We evaluate its performance using image classification, object detection, and instance segmentation tasks, obtaining robust gains in all cases. Additional experiments on the Backgrounds Challenge [23] show that CAST-trained models are substantially more resilient to changes in object backgrounds when performing image classification. Finally, qualitative and quantitative experiments show that CAST improves object localization ability of contrastive SSL feature representations on COCO scene images and on downstream image classification tasks. We hope that CAST can enable self-supervised learning from unconstrained web-scale datasets containing images with complex interactions of multiple objects and lead to better out-of-distribution performance and greater robustness to contextual bias.

2. Related Work

Self-supervised learning: SSL methods learn features from unlabeled data using "pretext" tasks that provide free supervision, with the aim of performing well on related supervised learning tasks. A strand of research includes low- to high-level computer vision-based pretext tasks, including image inpainting [24], colorization [25, 26], predicting patch orderings [27, 28] or degree of rotation [29]. Pretext tasks that perform pseudo-labeling and clustering [13, 14, 19, 30–32] have also been shown to be effective. Recently, contrastive learning methods [8] that learn to perform instance discrimination [10, 12, 33–38] have been shown to be the most competitive with fully supervised learning.

As a result, recent work has focused on developing theoretical and empirical understanding of contrastive representations [39–41] and improving the learning framework. For instance, Purushwalkam and Gupta [42] propose a method to improve viewpoint invariance of contrastively-learnt representations. Zhang and Maire [43] utilize a hierarchical region structure of images to guide contrastive learning methods for improved segmentation performance. Zhao *et al.* [44] introduce a data-driven approach to make contrastive self-supervised models invariant to object background. In this paper, we show the utility of visual grounding for improving the contrastive representation learning.

Visual Grounding and Attention: Improving visual grounding of CNNs is an increasingly important computer vision problem , which can benefit applications such as image captioning [45], visual question answering [46], and debiased computer vision [47]. Grounding methods in these problems typically use human attention supervision [48–50]. In our work, we improve visual grounding of self-supervised models using object saliency maps.

Object Saliency Prediction: The goal of object saliency prediction is to identify and segment important objects of interest in an image. Saliency prediction methods can be classified into supervised and unsupervised methods. Supervised methods for saliency prediction [51, 52] typically rely on expensive human annotated training data. Classically, unsupervised saliency prediction methods utilized handcrafted priors based on human perception [53, 54] or image statistics [52, 55, 56]. More recently proposed neural network-based saliency prediction methods [57, 58] utilize saliency maps from the handcrafted unsupervised methods as noisy psuedo-labels for training, thus removing the need for human-labeled data. In this work we make use of Deep-USPS [59], an unsupervised saliency prediction algorithm which uses a two stage mechanism that combines hand-crafted supervision and iterative self-training.

3. Contrastive Attention-Supervised Tuning

Our method, which we call Contrastive Attention-Supervised Tuning (CAST), aims to tune self-supervised models to rely on the appropriate regions during contrastive learning. At a high level, CAST consists of two steps: 1. constrained sampling of the query and key crops from the original image based on constraints generated using an image saliency map, 2. contrastive learning with a loss that forces models to look at the relevant object regions that are common between the query and key crops through Grad-CAM supervision. While our approach is generic and can be applied to any architecture, we describe CAST in context of the Momentum Contrast (MoCo) [12] pretraining setup.

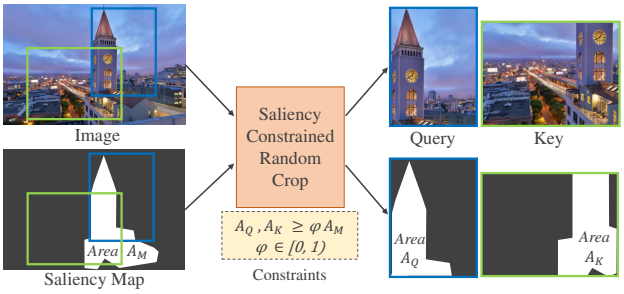


Figure 2: **Saliency-constrained Random Crop.** We compute query and key crops based on a saliency map and specified area constraints.

MoCo learns to perform the instance discrimination [33] task using the InfoNCE [35] contrastive objective (described in detail later in Section 3.3). In this task, a query and a key form a positive pair if they are data-augmented versions of the same image, and form a negative pair otherwise. MoCo builds a dynamic dictionary of negatives with a queue and a moving-averaged encoder which enabled access to a large and consistent dictionary which can be utilized for contrastive learning of representations.

3.1. Saliency-constrained Random Cropping

We aim to improve the visual grounding ability of self-supervised models by explicitly supervising models to look at relevant image regions. We provide this supervision as a *saliency map*—a binary mask indicating these relevant regions. These typically contain all the objects and other important visual concepts present in the image. We utilize Deep-USPS [59] to generate unsupervised saliency maps. But providing localization supervision is not sufficient to fix visual grounding. As shown in Figure 1b, models often receive a noisy training signal—random crops from an image may contain different objects, or none at all. To fix this problem, we design a random crop transform that generates input crops constrained to overlap with the saliency map.

Crop Constraints: Given an input image I with height h and width w, the standard data augmentation involves sampling two independent random crops (query and key) for input to the model. Here, we assume access to a saliency map $M \in \{0,1\}^{h \times w}$, where $M_{ij} = 1$ indicates pixel (i, j) is salient, and area of salient region is $A_M = \sum_{i,j} M_{i,j}$.

Consider the example in Figure 2. Our technique samples random crops based on a constraint specified by a hyperparameter $\phi \in [0, 1)$: *the area of saliency map M covered by each crop must be at least $\phi \cdot A_M$.*

We refer ϕ as the *area-overlap threshold*. Higher values of ϕ imply stricter constraints—enforcing higher overlap between sampled crops and salient regions, whereas setting $\phi = 0.0$ recovers the unconstrained random crop, used by MoCo and other existing SSL methods.

As seen in Figure 2, this simple area-overlap based constraint ensures that both the query and key crops contain some salient regions, and we supervise models to focus on them during training to improve visual grounding.

The premise of our approach is that when contrastive models such as MoCo [12] are given multiple crops from an image, focusing on the salient (object) regions in the crops would make them learn representations that are more generalizable. These models are likely to be more grounded, and are thus less likely to learn unwanted biases. CAST introduces a grounding loss that encourages this behaviour.

Recall that MoCo samples two crops, *query* and *key*, and enforces their representations to be closer compared to the other representations in a large dynamic queue. The random cropping transformations (shown in Fig. 2) used to obtain the query and key crop can also be applied to the image-specific saliency map, M. This results in two corresponding saliency maps M_q and M_k, each containing the salient object regions in the *query* and *key* crop.

However, the entirety of the object may not exist in both the *query* and the *key* crop. Hence, when considering the saliency map corresponding to the *query*, M_q, there can be cases where only a part of the salient region in the query exists in the key. In such scenarios, we consider *all* the regions in the query that correspond to the salient regions in key. See example in Fig. 3, where the two crops contain varying extent of the sheep. In such cases, the saliency map corresponding to the query would contain all regions in the query that contain the sheep. Qualitative examples showing how our random crops differ from regular random crops used by self-supervised learning methods can be found in the Appendix. As described next, we use these saliency maps to supervise where networks attend to.

3.2. Computing Network Importance

We define network importance as the importance placed by the encoder network, f_q on the spatial regions of the query, x^q, in order to predict that the query representation is closest to the representation of the key x^k, as compared to all the representations present in the queue. To compute network importance, we extend Grad-CAM [22] to contrastively-trained models.

To obtain Grad-CAM, we first forward-propagate the query crop x^q to query encoder f_q (see blue encoder box in Fig. 3). The key is then masked with the corresponding saliency map to obtain the salient regions in the key crop (see the bottom part of Fig. 3). This masked key, $x^k_m = x^k * M_k$, is then fed to the key encoder (see green encoder box in Fig. 3), f_k. Following MoCo, we dot-product the query representation, $q = f_q(x^q)$, with the masked key crop representation $k^m_+ = f_k(x^k * M_k)$, and each of the other representations in the dynamic queue, and concatenate them. We then one-hot encode the dot-product for the correct key and compute its gradients (shown as blue backward

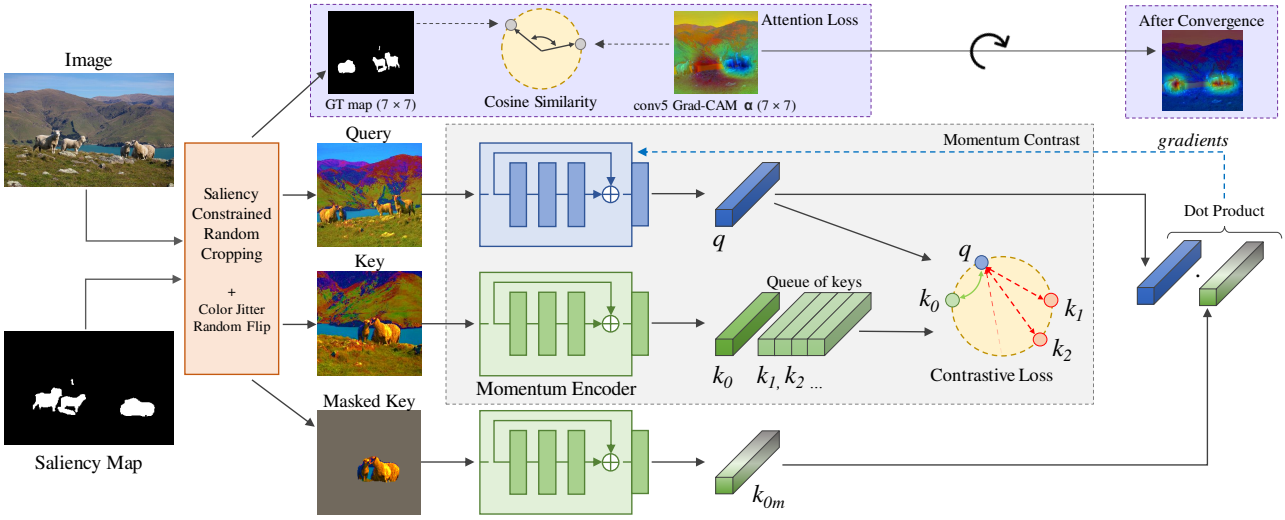


Figure 3: **Contrastive Attention-Supervised Tuning (CAST):** Given an image (top-left), we compute a saliency map (bottom-left), which we use to generate query and key crops and their corresponding saliency maps. We obtain the query and key feature representations with a forward-pass through the encoder network (blue box) and momentum encoder, respectively. Through the contrastive loss, we pull the representations of query and key crop together, while pushing the query representation away from other representations in a dynamic queue. We pass the salient regions in the key crop through the same momentum encoder, and compute the dot product between query and masked key representation. We then compute its gradient with respect to the last encoder convolution layer and weigh the forward activation maps to get Grad-CAM map. Finally, we add an attention loss that encourages Grad-CAM to look at all the salient image regions in the query crop.

arrow in Fig. 3) w.r.t. last convolutional layer activations of the encoder network, $A^{f_q}_{conv5}$, as,

$$\alpha_q = \overbrace{\sum_{i,j}}^{\text{global pooling}} \underbrace{\frac{\partial q \cdot k^m_+}{\partial A^{f_q}_{conv5}}}_{\text{gradients via backprop}} \tag{1}$$

As in Grad-CAM [22], the α_q values indicate the importance of each of the last convolutional layer neurons, n, in the encoder network for matching the query and masked key representation. To get the regions represented by these important convolutional neurons, we use α_q to perform a weighted combination of forward activation maps corresponding to query, $A^{f_q}_{conv5}$, followed by a ReLU to obtain,

$$\mathrm{G}_q = ReLU \underbrace{\left(\sum_n \alpha_{\mathrm{q}} A^{f_{\mathrm{q}}}_{conv5} \right)}_{\text{linear combination}} \tag{2}$$

The higher values in the resulting Grad-CAM map (Fig. 3 top-right) indicates query regions which the network relies on when mapping the masked key regions, $x^k * M_k$, to the entire query crop, x^q. These heatmaps form the basis for enforcing attention supervision, which we explain next.

3.3. CAST Loss

The CAST loss consists of two components: 1. a Contrastive loss, L_{cont} from [12], that measures the similarities of original sample pairs (x_q and x_k) in the representation space (yellow circle in Fig. 3), and 2. an Attention loss, L_{att}, that measures the similarity of Grad-CAM heatmap to its corresponding saliency map, M_q (purple box in Fig. 3). L_{cont} is defined as

$$\mathcal{L}_{\text{cont}} = -\log \frac{\exp\left(\mathrm{q} \cdot \mathrm{k}_+ / \tau\right)}{\sum_{\mathrm{i}=0}^{\mathcal{K}} \exp\left(\mathrm{q} \cdot \mathrm{k}_\mathrm{i} / \tau\right)} \tag{3}$$

where K denotes the number of representations in the queue and τ is the temperature hyperparameter.

As network importances (from above) are gradient based, we penalize errors in the predicted Grad-CAM map, G_q based on cosine distance—emphasizing alignment over magnitude (see the top box in Fig. 3). We minimize the cosine distance loss as,

$$\mathcal{L}_{\text{att}} = 1 - \frac{G_q \cdot M_{\mathrm{q}}}{\|G_q\| \, \|M_{\mathrm{q}}\|} \tag{4}$$

The final Contrastive Attention-Supervised Tuning loss becomes $\mathcal{L}_{\mathcal{CAST}} = L_{cont} + \lambda L_{att}$

The second term encourages the network to base predictions on the correct regions and the first term encourages the network to actually make the right prediction. Note that $A^{f_q}_{conv5}$ is a function of all the encoder parameters until last convolution layer and α_q is a function of the layers from the last convolutional layer until the final fully-connected layer, and the key encoder features. The keys, k and k_m, are detached from the key encoder, and therefore gradients do not get passed through them. Hence, while Grad-CAM is a function of both the query and the key encoder weights, during the update through an optimization algorithm, only

Method	VOC07 clf.	IN-1k clf.	PASCAL VOC Detection			COCO Instance Segmentation					
	mAP	Top-1 acc.	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}_{all}	AP^{mask}_{50}	AP^{mask}_{75}
1) Random Init	–	–	33.8	60.2	33.1	36.7	56.7	40.0	33.7	53.8	35.9
2) ImageNet Fully Sup	–	–	53.5	81.3	59.1	38.9	59.6	42.7	35.4	56.5	38.1
3) COCO Fully Sup	86.2	46.4	50.9	79.2	54.7	40.3	61.3	43.7	36.5	58.1	39.1
4) MoCo-COCO	67.5	46.5	47.5	75.4	51.5	38.3	58.7	41.5	34.9	55.7	37.2
5) + Constrained Crop	71.1 $_{+3.6}$	46.0 $_{-0.5}$	49.0 $_{+1.5}$	77.4 $_{+2.0}$	52.4 $_{+0.9}$	38.3 $_{+0.0}$	58.7 $_{+0.0}$	41.6 $_{+0.1}$	34.8 $_{-0.1}$	55.7 $_{+0.0}$	37.2 $_{+0.0}$
6) + CAST	74.0 $_{+7.0}$	48.7 $_{+2.1}$	54.2 $_{+6.7}$	80.1 $_{+4.7}$	59.9 $_{+8.4}$	39.4 $_{+1.1}$	60.0 $_{+1.3}$	42.8 $_{+1.3}$	35.8 $_{+0.9}$	57.1 $_{+1.4}$	38.6 $_{+1.4}$

Table 1: **Transfer Learning on Downstream Tasks:** We report results on four downstream tasks. For every task, all methods use the same architecture and learning setup. VOC07 and IN-1k use frozen feature extractor, COCO Instance Segmentation and PASCAL VOC Detection involve end-to-end fine-tuning. Gaps with MoCo-COCO are shown on the side (differences ≥ 0.5 are colored). We observe that training with CAST outperforms all baselines by a huge margin on all downstream tasks.

the query encoder weights are updated. In MoCo, since the key encoder is a moving average of the query encoder, the key encoder weights get updated eventually during training.

4. Evaluation

In our experiments, we aim to show that training self-supervised models with localization supervision offers two benefits—better visual grounding ability, and better transfer learning performance. We pretrain MoCo [12] with CAST on images from the COCO dataset [20], and then evaluate the transfer performance and grounding ability of learned features on multiple downstream tasks.

4.1. Transfer Learning on Downstream Tasks

First, we evaluate the quality of the learned features by transferring them to four downstream visual recognition tasks: **(a)** PASCAL VOC [60] linear classification, **(b)** ImageNet-1k [2, 61] linear classification, **(c)** PASCAL VOC object detection, **(d)** COCO [20] instance segmentation. Consistent with prior SSL research, our downstream tasks involve learning setups where the pretrained network is used as either a frozen feature extractor **(a, b)**, or weight initialization for fine-tuning **(c, d)**.

Baselines: We compare MoCo-COCO + CAST with baseline methods to show the importance of different components of our algorithm:

1. **Random Init** uses no pretrained visual features.
2. **MoCo-COCO**, without CAST attention loss ($\lambda = 0$) and constrained random cropping ($\phi = 0$).
3. **MoCo-COCO + Constrained Crop**, without CAST attention loss, to observe gains from better cropping alone.

For all tasks, we follow the same hyperparameters as VirTex [18], using its publicly available code [1]. VirTex uses a similar evaluation setup as the majority of recent work on self-supervised learning [11–14, 62], including our primary baseline, MoCo. We describe the main details here.

PASCAL VOC Linear Classification: We train on VOC07 `trainval` split and report mAP on `test` split. We extract the 7×7 spatial grid of 2048-dimensional features from the last convolutional layer, and downsample them to 2×2 grid via adaptive average pooling. Then, we flatten and L2-normalize these features to yield 8192-dimensional features. We train per-class SVMs for costs $C \in \{0.01, 0.1, 1.0, 10.0\}$, and select best C by 3-fold cross validation. Other SVM hyperparameters are same as [18].

ImageNet-1k Linear Classification: We train on ILSVRC 2012 `train` split and report top-1 center crop accuracy on the `val` split. We train a linear layer on 2048-dimensional global average pooled features extracted from the network. We train for 100 epochs using SGD with momentum 0.9, weight decay 0, and with batch size 256 distributed across 8 Nvidia V100 GPUs. Similar to MoCo, we start with learning rate 30, and divide it by 10 at epochs 60 and 80.

PASCAL VOC Object Detection: We train Faster R-CNN [4] with ResNet-50-C4 backbone. We initialize this backbone with pretrained weights, train on `trainval07+12` split, and evaluate on `test2007` split. We train for 24K iterations using SGD with momentum 0.9, batch size 16 (2 per GPU), and weight decay 10^{-4}. We use maximum learning rate 0.02, perform linear warmup for first 100 iterations, and divide it by 10 at iterations 18K and 22K. We fine-tune the network end-to-end, with batch normalization layers synchronized across GPUs (*SyncBN*) [63].

COCO Instance Segmentation: We train Mask R-CNN [6] models with ResNet-50-FPN backbones [64] on `train2017` split, and evaluate on `val2017` split. We follow $2\times$ training schedule implemented in Detectron2 [65], and fine-tune with SyncBN in the backbone and FPN layers.

Results: We summarize our results in Table 1. MoCo + CAST outperforms MoCo on all downstream tasks, obtaining robust gains on classification, detection, and instance segmentation. The performance improvement is especially large on the VOC detection task, aided by the improved visual grounding in models trained with CAST. We also find that our unsupervised saliency-constrained cropping alone outperforms MoCo on VOC07 and VOC-Detection,

[1] Code available at: https://github.com/kdexd/virtex

Area threshold	$\phi = 0.2$	$\phi = 0.0$
VOC07	74.0	73.3 $_{-0.7}$
IN-1k	48.7	47.4 $_{-1.3}$

(a) Effect of Area threshold ϕ (Fixing $\lambda = 3.0$)

Loss weighing factor	$\lambda = 0.0$	$\lambda = 1.0$	$\lambda = 3.0$	$\lambda = 5.0$
VOC07	71.1	74.0 $_{+2.9}$	74.0 $_{+2.9}$	73.3 $_{+2.2}$
IN-1k	46.5	48.7 $_{+2.2}$	48.7 $_{+2.2}$	47.6 $_{+1.1}$

(b) Effect of loss weighing factor λ (Fixing $\phi = 0.2$)

MoCo Projection Layer	1-layer Linear	2-layer MLP
VOC07	74.0	74.3 $_{+0.3}$
IN-1k	48.7	50.1 $_{+1.4}$

(c) Effect of improving underlying MoCo-COCO

Supervision	VOC07	IN-1k	PASCAL VOC Detection		
	mAP	Top-1	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}
Query	74.0	48.7	54.2	80.1	59.9
Intersection	72.0 $_{-2.0}$	49.4 $_{+0.7}$	53.3 $_{-0.9}$	79.7 $_{-0.4}$	59.0 $_{-0.9}$

(d) Effect of suppressing saliency supervision

Table 2: **Ablations for MoCo-COCO + CAST training:** We conduct ablation studies to isolate the effects of our training components. **(a)** Replacing saliency-constrained random cropping with default version from MoCo ($\phi = 0.0$) hurts performance. **(b)** Increasing weight of CAST loss generally improves performance up to a point ($\lambda = 1.0, 3.0$). **(c)** Adding known improvements to underlying MoCo model (MLP layer) also transfer to CAST. **(d)** Restricting attention supervision to only the intersection of query and key hurts downstream performance.

and gets close to MoCo performance on Imagenet-1k and COCO instance segmentation tasks.

4.2. Ablation Studies

Next, we conduct ablation studies on our training setup to isolate the effect of our design decisions. In all these comparisons, we treat MoCo-COCO with CAST trained with default hyperparameters as our base model. We mainly observe downstream performance of all ablations on VOC07 and IN-1k linear classification setups.

Effect of area threshold ϕ: We use area-overlap based constraints conditioned on saliency maps for sampling random crops, specifying them via an area threshold hyperparameter ϕ. Here, we quantify the downstream performance improvement due to *better* training supervision from strategically sampled crops—we train a model with $\phi = 0.0$ to recover the default random crop used in MoCo. Results are in Table 2a, we observe that removing saliency-constrained random cropping hurts performance, indicating that our saliency-constrained random cropping technique indeed provides better training signal.

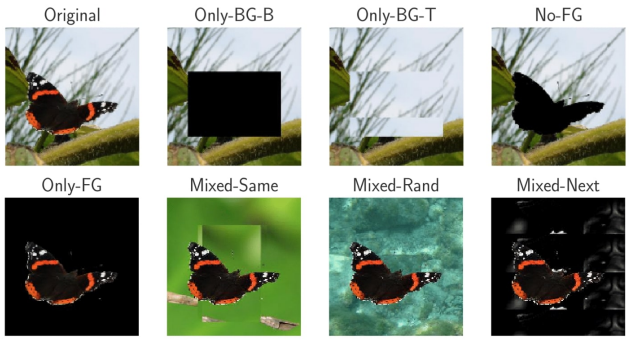


Figure 4: We evaluate CAST using the Backgrounds Challenge [23] dataset designed to evaluate background-robustness of models. FG = Foreground, BG = background. Foreground-background combinations include: Only-BG-B (FG: Black, BG: Unmodified), Only-BG-T (FG: Tiled background, BG: Unmodified), Mixed-Same (FG: Unmodified, BG: Random BG of the same class), Mixed-Rand (FG: Unmodified, BG: Random BG of a random class), and Mixed-Next (FG: Unmodified, BG: Random BG of the next class.)

Effect of loss weighing factor λ: As described in Section 3.3, the CAST loss is a linear combination of contrastive and attention losses. We combine them through a weighted sum, and use λ to scale the attention loss. Here, we experiment with different values of λ with $\lambda \in \{0.0, 1.0, 3.0, 5.0\}$. Note that $\lambda = 0.0$ means MoCo-COCO + Constrained Crop (Table 1, row 2). Results from Table 2b show that non-zero values of λ outperform $\lambda = 0.0$, indicating that attention loss is important in CAST. Higher λ improve performance up to a point—performance improves with $\lambda = 1.0, 3.0$, and slightly degrades with $\lambda = 5.0$.

Effect of improving underlying MoCo-COCO: CAST is a general purpose method that can be added to contrastive SSL methods to improve their visual grounding. Here, we investigate whether improving the underlying SSL method also shows improvements when trained with CAST. We consider MoCo-v2 variant [38], replacing the linear projection with an MLP, inspired by SimCLR [10]. Results from Table 2c show that MoCo-MLP + CAST matches or exceeds MoCo-COCO + CAST on downstream tasks, indicating that CAST can provide additive improvements over its underlying SSL method.

Effect of suppressing saliency supervision: We believe that focusing on salient image regions is important to improve visual grounding. Hence, we force the model to focus on *all* the salient regions inside query crop. In contrast to our proposed approach, we train MoCo-COCO + CAST with *reduced supervision* in this ablation study, enforcing the model to only look at salient regions inside the intersection of query and key crops. Results from Table 2d show that excluding some salient regions from the query crop (lying outside the intersection) significantly hurts downstream performance on multiple tasks. This indicates that looking

MoCo-COCO Performance	Backgrounds Challenge Setting							
	Original	Mixed-Same	Mixed-Rand	Mixed-Next	Only-FG	No-FG	Only-BG-B	Only-BG-T
Default	72.62	45.75	30.44	26.86	30.42	23.95	5.06	12.62
+ Constrained-Crop	74.79 $_{+2.17}$	52.64 $_{+6.89}$	39.14 $_{+8.70}$	34.17 $_{+7.31}$	33.73 $_{+3.31}$	22.74 $_{-1.21}$	4.10 $_{-0.96}$	11.88 $_{-0.74}$
+ CAST	77.33 $_{+4.71}$	54.42 $_{+8.67}$	39.93 $_{+9.49}$	37.46 $_{+10.60}$	43.26 $_{+12.84}$	23.70 $_{-0.15}$	4.40 $_{-0.66}$	12.59 $_{-0.03}$

Table 3: CAST obtains large improvements over MoCo on the Backgrounds Challenge, a 9-class image classification dataset containing foreground objects superimposed on various background types. In settings where the foreground is present (columns 1-5), CAST's visual grounding ability leads to substantial performance gains. When foreground is absent (columns 6-8), CAST performs slightly worse, validating that CAST-trained models learn fewer background correlations.

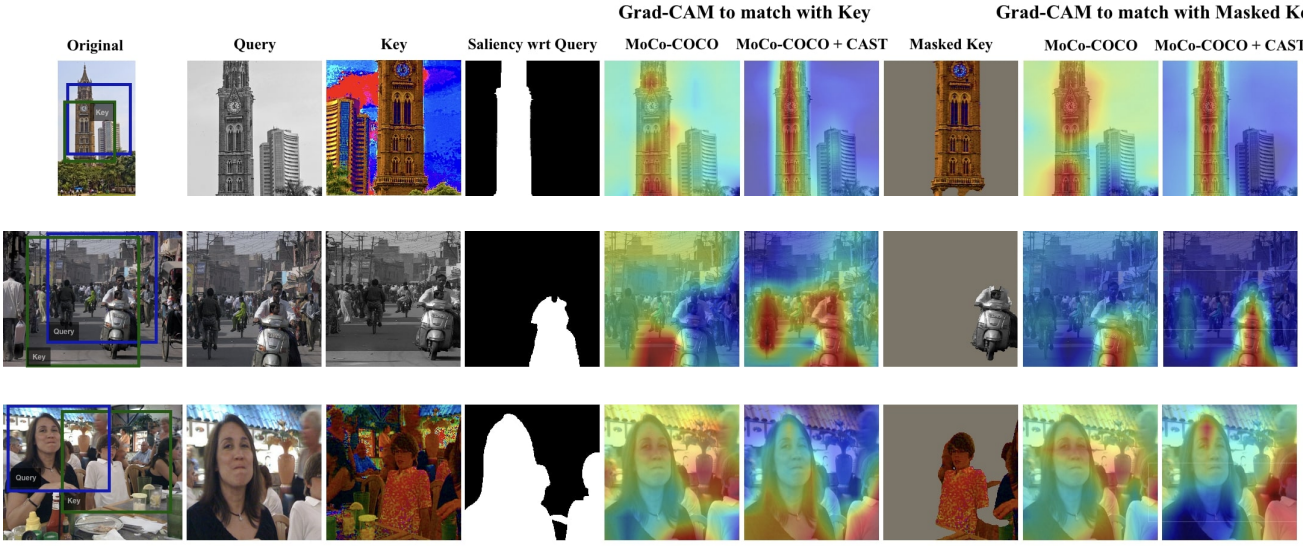


Figure 5: CAST improves visual grounding of the contrastive self-supervised feature encoder. Column (d) shows the saliency map according to the query crop (b). Grad-CAM visualizations in columns (e, f) show the query regions that MoCo and MoCo + CAST models rely on, in order to match the key crop (c). Finally, the MoCo and MoCo + CAST models rely on query regions (h) and (i) to match with the masked key representation (g). The example in top row shows that MoCo also looks at the sky in the background to match the masked key to the tower image in the query, indicating it has learnt spurious correlations. In contrast, the MoCo + CAST model looks just at the salient tower region. Similar trends are seen in the second row. In the third row, where there are two women in the foreground, the two crops contain different women. The MoCo + CAST model is able to localize the woman in black when matching the woman in white (see masked key), indicating that it has learned semantic category-specific representations. The baseline MoCo model looks primarily at the background regions.

beyond the common visual content between two crops to solve instance discrimination yields better visual features.

4.3. Evaluation on Backgrounds Challenge Dataset

The Backgrounds Challenge [23] aims to assess the background-robustness of image classification models by measuring their accuracy on images containing foreground objects superimposed on various background types (see [23] for details on dataset construction). The dataset consists of 9 ImageNet classes with 450 test images per class. The evaluations are performed on eight foreground-background combinations summarized in Figure 4. Since CAST forces a model to attend to salient objects during training, we expect a CAST-trained model to be less dependent on background correlations for classification.

We evaluate the performance of COCO-pretrained models on the Backgrounds Challenge using a linear layer trained with ImageNet-1K (as described in Section 4.1) using three settings: 1. MoCo, 2. MoCo trained with saliency-constrained random cropping alone, 3. MoCo trained with CAST (Table 3). Models trained with cropping constrains and with CAST outperform vanilla MoCo on all eight settings of the Backgrounds Challenge, with CAST obtaining the best performance on the five settings where foreground is present. In the Only-FG setting, where background is set to black, CAST obtains an absolute improvement of 13% over MoCo, indicating that CAST is significantly better at utilizing the foreground information, due to the saliency-driven attention-supervised training. In settings where background is swapped (Mixed-Same, Mixed-Rand, and Mixed-Next), CAST obtains 5-10% absolute im-

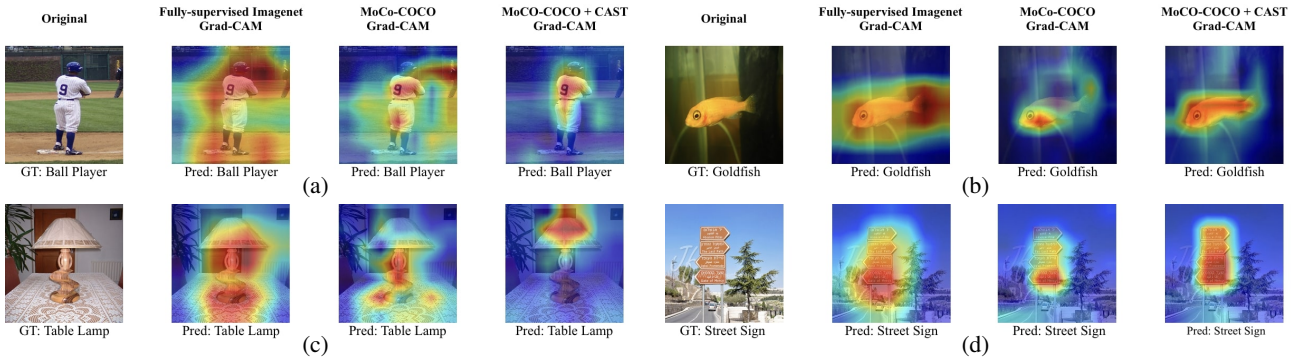


Figure 6: Training with CAST also leads to improvements in grounding in downstream tasks. In this comparison of Grad-CAM attention maps from a fully supervised network and the self-supervised networks (MoCo and MoCo + CAST) on Imagenet-1k, we find that MoCo + CAST models tend to rely less on spurious correlations. (a) The MoCo + CAST model looks just at the player, whereas both fully supervised model and MoCo rely on the regions corresponding to the baseball field. (c) The MoCo + CAST model looks only at the lamp, while other models also rely on the table below. (b, d) The MoCo + CAST model is much more precise at attending to the whole extent of the object of interest as compared to other methods.

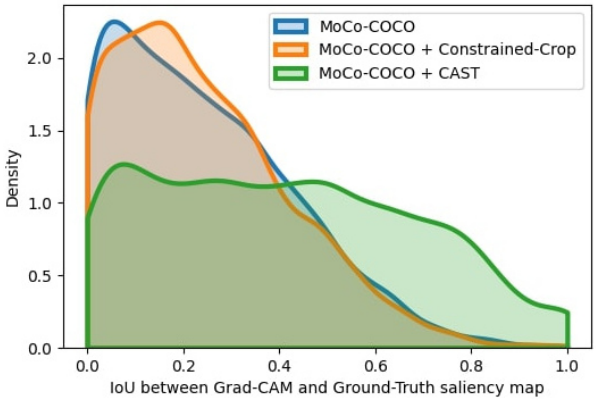


Figure 7: CAST shows quantitative improvement in grounding for contrastive self-supervised models. The distinct rightward shift in the green curve corresponding to the MoCo + CAST model shows that the gradient-based localization supervision loss significantly improves grounding.

provements, indicating that models trained with CAST are less dependent on background correlations. Finally, in settings that do not contain foreground objects (No-FG, Only-BG-B, and Only-BG-T), CAST performs slightly worse than the original model, as we would expect from a model that has learnt to rely less on the background signal in making classification decisions. Qualitative examples showing how CAST makes downstream models rely less on spurious background correlations can be found in Appendix.

4.4. Evaluating Visual Grounding

We use Grad-CAM for qualitative and quantitative evaluation of the visual grounding ability of a contrastive SSL model trained with CAST and its effect on grounding in downstream tasks. Examples in Fig. 5 show that the CAST-trained model seems to learn semantic category-specific feature representations, which allows it to look at objects of interest while performing query-key matching, and avoid learning spurious correlations. We quantify the improvement in grounding due to CAST using the COCO val split. First, we binarize the Grad-CAM maps by thresholding at 0.5. We then compute the intersection over union (IoU) between the Grad-CAM map and the saliency map corresponding to the query image. Fig. 7 shows the density of IoU values for the baseline MoCo-COCO, MoCO-COCO with constrained cropping and MoCO-COCO with CAST. The mean IoU of the MoCo model trained with CAST over the COCO val set is 0.41, substantially larger than the mean IoU of the model trained without CAST, which is 0.24. Moreover, the improvement in grounding ability is largely driven by the gradient-based localization supervision loss, as the mean IoU of a model trained with saliency-driven cropping constraints alone is also 0.24.

Examples in Fig. 6 shows how the improved grounding during pre-training translates to improved grounding in the downstream task of Imagenet linear classification. As seen in Fig. 6 (a,c), the MoCo-COCO+CAST model relies less on spurious background correlations — relying mostly on the player to predict Ball Player and the lamp to predict Table Lamp. Fig. 6 (b,d) show that models pretrained with CAST learn to look at the whole extent of the object of interest as compared to other methods.

5. Conclusion

We introduced a method for visually grounding contrastive self-supervised learning models, which improves feature representations learnt from scene images. These feature representations are also less reliant on background correlations as compared to those trained with contrastive learning alone, which can lead to better out-of-distribution performance and greater robustness to contextual bias and adversarial backgrounds. We hope that our method leads to development of more general-purpose and robust self-supervised methods that learn from noisy, unconstrained, real-world image data from the web.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012. 1

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015. 1, 5

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014. 1

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015. 1, 5

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 1

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *ICCV*, 2017. 1, 5

[7] Michael Gutmann and Aapo Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010. 1

[8] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006. 1, 2

[9] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin, "Unsupervised feature learning via non-parametric instance-level discrimination," 2018. 1

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020. 2, 6

[11] Ishan Misra and Laurens van der Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020. 1, 5

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020. 1, 2, 3, 4, 5

[13] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C.H. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020. 2

[14] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020. 1, 2, 5

[15] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten, "Exploring the limits of weakly supervised pretraining," in *ECCV*, 2018. 1

[16] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li, "YFCC100M: The new data in multimedia research," *Communications of the ACM*, 2016. 1

[17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *ICCV*, 2017. 1

[18] Karan Desai and Justin Johnson, "Virtex: Learning visual representations from textual annotations," *arXiv preprint arXiv:2006.06666*, 2020. 1, 5

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020. 1, 2

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014. 1, 5

[21] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *NeurIPS*, 2014. 1

[22] 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," in *ICCV*, 2017. 1, 2, 3, 4

[23] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry, "Noise or signal: The role of image backgrounds in object recognition," *arXiv preprint arXiv:2006.09994*, 2020. 2, 6, 7

[24] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, pp. 2536–2544, 2016. 2

[25] Richard Zhang, Phillip Isola, and Alexei A. Efros, "Colorful image colorization," in *ECCV*, pp. 649–666, 2016. 2

[26] Richard Zhang, Phillip Isola, and Alexei A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *CVPR*, pp. 645–654, 2017. 2

[27] Carl Doersch, Abhinav Gupta, and Alexei A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015. 2

[28] Mehdi Noroozi and Paolo Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*, pp. 69–84, 2016. 2

[29] Spyros Gidaris, Praveer Singh, and Nikos Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018. 2

[30] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018. 2

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

[32] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi, "Self-labelling via simultaneous clustering and representation learning," *arXiv preprint arXiv:1911.05371*, 2019. 2

[33] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, pp. 3733–3742, 2018. 2, 3

[34] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *CVPR*, pp. 6210–6219, 2019.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018. 3

[36] Yonglong Tian, Dilip Krishnan, and Phillip Isola, "Contrastive multiview coding," in *ECCV*, 2020.

[37] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in Neural*

Information Processing Systems, vol. 33, 2020.

[38] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020. 2, 6

[39] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola, "What makes for good views for contrastive learning," *arXiv preprint arXiv:2005.10243*, 2020. 2

[40] Tongzhou Wang and Phillip Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," *arXiv preprint arXiv:2005.10242*, 2020.

[41] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli, "Understanding self-supervised learning with dual deep networks," *arXiv preprint arXiv:2010.00578*, 2020. 2

[42] Senthil Purushwalkam and Abhinav Gupta, "Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases," *arXiv preprint arXiv:2007.13916*, 2020. 2

[43] Xiao Zhang and Michael Maire, "Self-supervised visual representation learning from hierarchical grouping," *Advances in Neural Information Processing Systems*, vol. 33, 2020. 2

[44] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin, "Distilling localization for self-supervised representation learning," *arXiv preprint arXiv:2004.06638*, 2020. 2

[45] Chih-Yao Ma, Yannis Kalantidis, Ghassan AlRegib, Peter Vajda, Marcus Rohrbach, and Zsolt Kira, "Learning to generate grounded visual captions without localization supervision," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[46] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, "VQA: Visual question answering," in *ICCV*, 2015. 2

[47] Sen Jia, Thomas Lansdall-Welfare, and Nello Cristianini, "Right for the right reason: Training agnostic networks," in *International Symposium on Intelligent Data Analysis*, pp. 164–174, Springer, 2018. 2

[48] Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille, "Attention correctness in neural image captioning," *arXiv preprint arXiv:1605.09553*, 2016. 2

[49] Ramprasaath R. Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh, "Taking a hint: Leveraging explanations to make vision and language models more grounded," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[50] Tingting Qiao, Jianfeng Dong, and Duanqing Xu, "Exploring human-like attention supervision in visual question answering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[51] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2083–2090, 2013. 2

[52] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014. 2

[53] Neil Bruce and John Tsotsos, "Saliency based on information maximization," *Advances in neural information processing systems*, vol. 18, pp. 155–162, 2005. 2

[54] Jonathan Harel, Christof Koch, and Pietro Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, pp. 545–552, 2007. 2

[55] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1915–1926, 2011. 2

[56] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu, "Global contrast based salient region detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 569–582, 2014. 2

[57] Dingwen Zhang, Junwei Han, and Yu Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4048–4056, 2017. 2

[58] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley, "Deep unsupervised saliency detection: A multiple noisy labeling perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9029–9038, 2018. 2

[59] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox, "Deepusps: Deep robust unsupervised saliency prediction via self-supervision," in *Advances in Neural Information Processing Systems*, pp. 204–214, 2019. 2, 3

[60] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, 2009. 5

[61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 5

[62] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra, "Scaling and benchmarking self-supervised visual representation learning," in *CVPR*, 2019. 5

[63] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun, "MegDet: A large mini-batch object detector," in *CVPR*, 2018. 5

[64] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017. 5

[65] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick, "Detectron2." https://github.com/facebookresearch/detectron2, 2019. 5