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

Ramprasaath R. Selvaraju1* Karan Desai2* Justin Johnson2 Nikhil Naik1
1Salesforce Research, 2University 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.

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]).

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 instance discrimination task with random views from im-

*Equal Contribution
Objective 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 [50, 51] typically rely on expensive human annotated training data. Classically, unsupervised saliency prediction methods utilized handcrafted priors based on human perception [52, 53] or image statistics [51, 54, 55]. More recently proposed neural network-based saliency prediction methods [56, 57] utilize saliency maps from the handcrafted unsupervised methods as noisy pseudo-labels for training, thus removing the need for human-labeled data. In this work we make use of Deep-USPS [58], 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 encourage them 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.

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

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 image regions. These typically contain all the objects and other important visual concepts present in the image. In this work we utilize Deep-USPS [58] 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 \( \phi \) as the area-overlap threshold. Higher values of \( \phi \) 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 the 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 Fig. S8 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, \( x^m_k = x^k \odot M_k \), 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 \( k^m = f_k(x^k \odot M_k) \), 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 gradients 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.

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, $x^k \ast M_k$, 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

$$L_{cont} = -\log \frac{\exp (q \cdot k_+ / \tau)}{\sum_{k=0}^{K} \exp (q \cdot k_i / \tau)}$$

where $K$ denotes the number of representations in the queue and $\tau$ 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,

$$L_{att} = 1 - \frac{G_q \cdot M_q}{\|G_q\| \|M_q\|}$$

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

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 $A^{f_5}_{conv5}$ is a function of all the encoder parameters until last convolution layer and $\alpha_q$ is a function of the layers from the last convolutional layer until the final fully-connected layer, and the key encoder features. They keys, $k$ and $k_m$, are detached from the key encoder, and therefore gradients...
We compare MoCo-COCO + CAST with base-
initialization for fine-tuning
VOC object detection,
(d) by transferring them to four downstream visual recogni-
tion tasks involving 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.

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 [59] linear classification, (b) ImageNet-1k [2, 60] 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 (λ = 0)
   and constrained random cropping (φ = 0).
3. MoCo-COCO + Constrained Crop, without CAST at-
tention loss, to observe gains from better cropping alone.

For all tasks, we follow the same hyperparameters as Vir-
Tex [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, 61], 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
fast 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 ∈ {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 iter-
ations 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 syn-
chronized across GPUs (SyncBN) [62].

COCO Instance Segmentation: We train Mask R-
CNN [6] models with ResNet-50-FPN backbones [63] on
train2017 split, and evaluate on val2017 split. We follow
2× training schedule implemented in Detectron2 [64], 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, obtain-
ing 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

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.

| Method               | VOC07 clf. | IN-1k clf. | PASCAL VOC Detection | COCO Instance Segmentation |
|----------------------|------------|------------|-----------------------|---------------------------|
|                      | mAP        | Top-1 acc. |                       |                           |
| 1) Random Init       | 33.8       | 60.2       | 33.1                  | 36.7                      |
| 2) MoCo-COCO         | 66.9       | 46.5       | 47.5                  | 51.5                      |
| 3) + Constrained Crop| 69.3±1.3   | 46.0−0.5   | 49.0±1.5              | 52.4±0.9                 |
| 4) + CAST            | 73.1±6.2   | 48.7±2.1   | 54.2±6.7              | 59.9±4.4                 |

1Code available at: https://github.com/kdexd/virtex
constrained random cropping hurts performance, indicating that our saliency-constrained random cropping technique indeed provides better training signal.

**Effect of loss weighing factor $\lambda$:** 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 $\lambda$ to scale the attention loss. Here, we experiment with different values of $\lambda$ 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 $\lambda$ outperform $\lambda = 0.0$, indicating that attention loss is important in CAST. Higher $\lambda$ 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 our saliency-constrained random cropping technique indeed provides better training signal.

### 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.

#### 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 $\phi$:** We use area-overlap based constraints conditioned on saliency maps for sampling random crops, specifying them via an area threshold hyperparameter $\phi$. 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...
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.

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 combinations...
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.

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. More examples can be found in appendix (Fig. S9 and Fig. S10).
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.

Appendices

A. Introduction

This supplementary material is organized as follows. We first provide pseudo-code for our approach, Contrastive Attention-Supervised Tuning (CAST), showing how it can be easily applied on top of any contrastive self-supervised approaches. We then provide a sample of the kinds of random crops that are generated by our saliency-constrained random cropping approach and compare them to regular random crops used by self-supervised learning methods. We then provide randomly sampled qualitative examples that show that CAST-trained feature representations obtain better grounding, both on their own and when finetuned on downstream tasks, such as image classification. We then show qualitative examples from the Background Challenge evaluation reported in Section 4.3 of the main paper.

B. Random Crops

As shown in Fig. 1b of the main paper, when training with complex scene-level images, 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. In Figure S8, we show how our Saliency-Constrained Random Crops differ from unconstrained random crops used in MoCo and other contrastive approaches.

B.1. SSL Grounding

Similar to Section 4.4 in the main paper, we use Grad-CAM to evaluate the visual grounding ability of a contrastive SSL model trained with CAST and its effect on grounding in downstream tasks. Randomly sampled examples in Figure S9 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.

B.2. Grounding of CASTed models on Downstream tasks

Fig. S10 shows randomly sampled qualitative examples showing how training with CAST also leads to improvements in grounding in downstream tasks. Notice in Fig. S10 (a, b) how the CASTed model looks at the whole extent of the object regions. We can also see in Fig. S10 (c) how in order to predict “Table Lamp”, the MoCo pretrained model also looks at the table where as the CASTed model only looks at the relevant lamp regions. In Fig. S10 (f), in addition to not predicting the correct class “French Horn”, the
MoCo pretrained model is unable to localize the category, where as the CASTed model correctly predicts and attends to the “French Horn”.

**B.3. Qualitative examples from Background Challenge**

The Backgrounds Challenge [23] (Sec. 4.3 of the main text) aims to assess the background-robustness of image classification models by measuring their accuracy on im-
Figure S8: Saliency-constrained random crops always have salient regions that are overlapping between two randomly sampled crops. This can be compared to unconstrained random crops used in MoCo. Notice how in the random crops of (a), the stop sign (salient region) is missing from query, however query and key crops from our saliency-constrained random cropping approach contain a part of the stop sign. Similarly in (b) the salient region corresponding woman is missing from one of the crops, and in (c), the salient regions corresponding to the surfer is not present in both the query and key crops. The crops which do not contain overlapping salient regions (top-row in each subfigure) tend to provide noisy training signal to contrastive approaches forcing them to incorrectly produce similar features for crops containing varying context, e.g., stop sign and sky in (a), surfer and waves in (c), etc. Saliency-constrained random cropping mitigates this problem.

References

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

[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.

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

[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.

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

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Gir-
Figure S9: 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). Notice how in all the examples, models trained with CAST look at the salient objects of interest to match the query with the key or the saliency-masked key.
Figure S10: Downstream task Grounding. Qualitative comparison of Grad-CAM attention maps for the correct class from fully supervised networks and the self-supervised networks (MoCo and MoCo + CAST) on the Imagenet-1k classification task showing cases where the CAST fixes the mistakes made by MoCo model. We can see that CASTed models tend to look at the full extent of the object and less at the background regions. Note that even for incorrect predictions, Grad-CAM maps are computed for the ground-truth class.

[Refs]
[1] Shick, “Mask R-CNN,” in ICCV, 2017. 1, 5
[2] Michael Gutmann and Aapo Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in AISTATS, 2010. 1
[3] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in CVPR, 2006. 1, 2
[4] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin, “Unsupervised feature learning via non-parametric instance-level discrimination,” 2018. 1
[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, “A simple framework for contrastive learning of visual representations,” in ICML, 2020. 2, 6
[6] Ishan Misra and Laurens van der Maaten, “Self-supervised learning of pretext-invariant representations,” in CVPR, 2020. 1, 5
[7] 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
[8] Junnan Li, Pan Zhou, Caixing Xiong, Richard Socher, and Steven C.H. Hoi, “Prototypical contrastive learning of unsupervised representations,” arXiv preprint arXiv:2005.04966, 2020. 2
[9] 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
[10] 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
[11] Bart Thomée, 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
[12] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in ICCV, 2017. 1
[13] Karan Desai and Justin Johnson, “Virtex: Learning visual representations from textual annotations,” arXiv preprint arXiv:2006.06666, 2020. 1, 5
[14] 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
[15] 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
[16] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, “Learning deep features for scene recognition using places database,” in NeurIPS, 2014. 1
[17] 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
[18] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry, “Noise or signal: The role of image backgrounds in...
Figure S11: Backgrounds-challenge. Models pretrained with CAST are less likely to rely on background correlations for classification. Notice how in (a), the MoCo pretrained model incorrectly predicts “Reptile” when the grass is changed to a background from a reptile class, while the CAST pretrained model still correctly predicts “Dog”. Similarly in (c), when chairs are introduced in the background, the MoCo pretrained model changes its prediction from “Reptile” to “Instrument”, indicating that the model relies on spurious correlations to predict the class of interest. The CAST pretrained model still correctly predicts “Reptile.”
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

Richard Zhang, Phillip Isola, and Alexei A. Efros, “Colorful image colorization,” in ECCV, pp. 649–666, 2016. 2

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

Carl Doersch, Abhinav Gupta, and Alexei A. Efros, “Unsupervised visual representation learning by context prediction,” in ICCV, 2015. 2

Mehdi Noroozi and Paolo Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in ECCV, pp. 69–84, 2016. 2

Spyros Gidaris, Praveer Singh, and Nikos Komodakis, “Unsupervised representation learning by predicting image rotations,” in ICLR, 2018. 2

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, “Deep clustering for unsupervised learning of visual features,” in ECCV, 2018. 2

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

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi, “Self-labeling via simultaneous clustering and representation learning,” arXiv preprint arXiv:1911.05371, 2019. 2

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

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018. 2

Yonglong Tian, Dilip Krishnan, and Phillip Isola, “Contrastive multiview coding,” in ECCV, 2020.

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.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, “Improved baselines with momentum contrastive learning,” arXiv preprint arXiv:2003.04297, 2020. 2, 6

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

Tongzhou Wang and Phillip Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” arXiv preprint arXiv:2005.10242, 2020.

Yuanlong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli, “Understanding self-supervised learning with dual deep networks,” arXiv preprint arXiv:2010.00578, 2020. 2

Senthil Purushwalkam and Abhinav Gupta, “Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases,” arXiv preprint arXiv:2007.13916, 2020. 2

Xiao Zhang and Michael Maire, “Self-supervised visual representation learning from hierarchical grouping,” Advances in Neural Information Processing Systems, vol. 33, 2020. 2

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

Stanislav Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, “VQA: Visual question answering,” in ICCV, 2015. 2

Sen Jia, Thomas Lamsdall-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

Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille, “Attention correctness in neural image captioning,” arXiv preprint arXiv:1605.09553, 2016. 2

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.

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

Huaiyu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanxing 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

Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun, “Salient object detection: A discriminative regional feature integration approach,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2814–2821, 2014. 2

Neil Bruce and John Tsotsos, “Saliency based on information maximization,” Advances in neural information processing systems, vol. 18, pp. 155–162, 2005. 2

Jonathan Harel, Christof Koch, and Pietro Perona, “Graph-based visual saliency,” in Advances in neural information processing systems, pp. 545–552, 2007. 2

Stas Goferman, Lihl Zelnik-Manor, and Ayelet Tal, “Context-aware saliency detection,” IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 10, pp. 1915–1926, 2011. 2

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

Dingwen Zhang, Junwei Han, and Yu Zhang, “Supervision for good views for contrastive learning,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 4048–4056, 2017. 2

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 Recogni-
[58] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mum-madi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox, “Deepups: Deep robust unsupervised saliency prediction via self-supervision,” in Advances in Neural Information Processing Systems, pp. 204–214, 2019.

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

[60] 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.

[61] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra, “Scaling and benchmarking self-supervised visual representation learning,” in CVPR, 2019.

[62] 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.

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

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