Fixing Localization Errors to Improve Image Classification

Guolei Sun*[1], Salman Khan*[2], Wen Li[3], Hisham Cholakkal[2], Fahad Shahbaz Khan[2], and Luc Van Gool[1]

[1] ETH Zurich, Switzerland
[2] Mohamed Bin Zayed University of Artificial Intelligence, UAE
[3] University of Electronic Science and Technology of China
guolei.sun@vision.ee.ethz.ch

Abstract. Deep neural networks are generally considered black-box models that offer less interpretability for their decision process. To address this limitation, Class Activation Map [46] (CAM) provides an attractive solution that visualizes class-specific discriminative regions in an input image. The remarkable ability of CAMs to locate class discriminating regions has been exploited in weakly-supervised segmentation and localization tasks. In this work, we explore a new direction towards the possible use of CAM in deep network learning process. We note that such visualizations lend insights into the workings of deep CNNs and could be leveraged to introduce additional constraints during the learning stage. Specifically, the CAMs for negative classes (negative CAMs) often have false activations even though those classes are absent from an image. Thereby, we propose a loss function that seeks to minimize peaks within the negative CAMs, called '*Homogeneous Negative CAM*' loss. This way, in an effort to fix localization errors, our loss provides an extra supervisory signal that helps the model to better discriminate between similar classes. Our designed loss function is easy to implement and can be readily integrated into existing DNNs. We evaluate it on a number of classification tasks including large-scale recognition, multi-label classification and fine-grained recognition. Our loss provides better performance compared to other loss functions across the studied tasks. Additionally, we show that the proposed loss function provides higher robustness against adversarial attacks and noisy labels.

1 Introduction

The conventional training strategy for deep neural networks (DNNs) involves loss functions that operate on the logit space [19, 30]. Given an input, a DNN model learns a function that maps it to the output label space, where the loss

*Equal contribution

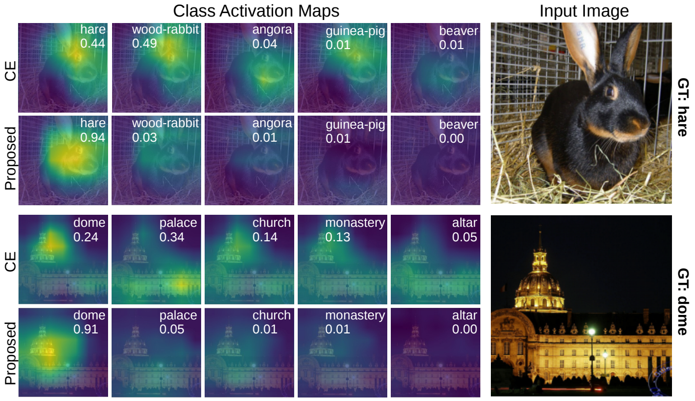

Fig. 1. Comparison of the class activation maps (CAMs) between the baseline (CE loss) and our proposed loss, for sample images from ImageNet [6]. Positive CAMs are shown on the far left, followed by top four negative CAMs, which are ranked based on classification probability. For each CAM, the corresponding class name and predicted probability are shown in the upper right region. For the baseline, there are many false activations in the negative CAMs (see *first* and *third* row). In contrast, our method produces clearer negative CAMs, thus avoiding localization errors and leading to a higher classification accuracy.

is computed. The network thus learned is considered a ‘black-box’ model whose prediction process lacks transparency for human understanding and interpretation. To resolve this limitation of DNNs, a number of approaches have been proposed to visualize the decision process within deep networks [31,32,42]. These approaches provide interpretable and intuitive explanations for DNN decisions, making them more transparent and explainable. One popular way to visualize the internal mechanics of DNNs is using the attention visualization corresponding to each category.

Zhou *et al.* [46] proposed class activation mapping (CAM), which illustrates the discriminative spatial regions in an image that are relevant to a specific class. Due to their remarkable ability to locate class-specific discriminative regions, CAMs have been shown to provide cost-free localizations for objects using just the image-level labels. In this work, we show that the interpretation provided by CAMs, into the internal mechanics of DNNs, can be exploited to add additional constraints and provide an extra supervisory signal during network optimization. Concretely, since a CAM provides coarse object location for a class, if the class is *absent*, the corresponding CAM should be *relatively* clear and have *no* or *less* attentive regions (peaks), compared with the CAM for the *positive* class. Hence, our novel loss function, called *Homogeneous Negative CAM* (HNC) loss, is proposed to suppress the peaks in the activation maps corresponding to the negative classes.

Despite the simplicity of our approach, it provides clear gains in problems such as image recognition, multi-label classification and fine-grained recognition.

For example, compared to the Cross Entropy (CE) loss baseline, HNC loss delivers *absolute* top-1 accuracy gains of 1.2% and 1.1% on CIFAR-100 and ImageNet datasets, respectively. The suppression of negative CAMs provides an additional supervision to the deep network, which helps resolve confusions regarding the final prediction, thereby helps improve the overall classification performance. As shown in Fig. 1, removing false peaks from negative CAMs also results in visualizations that are more consistent and faithful to given class labels for an image. Furthermore, we demonstrate that the HNC loss improves robustness of the learned model towards adversarial attacks and noisy labels.

2 Related Works

In this section, we first introduce popular network visualization approaches, and then review the recent advances in loss functions for optimizing the DNNs.

Network Visualization. Patterns that can maximally activate particular units within a deep network were synthesized using gradient information in [11,27,32]. Deep feature representations have also been inverted to reconstruct the corresponding input image [9,27]. Another category of visualization methods including DeConvNet [42] and Guided Back-propagation [33] amplify the salient patterns in an image by modifying the raw gradients. As such, the above-mentioned visualization methods are either non-discriminative for different classes or illustrate model behavior as a whole, instead of providing an image-specific visualization. To address this requisite, [46] proposed an activation visualization mechanism (i.e. CAM) that sheds light on the implicit attention of a DNN on an image. While [46] is applicable to a specific class of architectures (e.g., without fully connected layers), [31] extended the concept to work with a broader range of DNN architectures. Due to the class-discriminative nature and simplistic design of [46], we base our loss formulation on class activation maps.

Loss Functions. A major factor in deep neural network's design is the choice of a correct objective function. Cross-entropy loss is hitherto the most popular loss function for computer vision problems such as classification, retrieval, detection and segmentation [13]. For special cases, alternative loss functions have been proposed in the literature, which can be grouped into two main classes, (a) max-margin loss functions and (b) data-imbalance losses. The margin maximizing loss functions put relative constraints with respect to other class boundaries such that each class is well-separated in the output space [7,12,26]. These constraints are generally posed as an angular margin [7,25,26,37], a spatial distance measure [16] or as a ranking penalty for multi-label classification problems [12,44]. In the second category, cost-sensitive objectives [8,17,18] are designed to re-weight the loss such that all classes in a long-tail data distribution are adequately modeled. From another perspective, a set of loss functions seek to re-balance back-propagated gradients by focusing on difficult examples and putting less emphasis on easy cases [23,29].

The closest to our approach are [2,10,15]. Among these, [10] seeks to minimize the peaks in the output 'logit' space to improve generalization on fine-grained tasks. Guo *et al.* [15] work on CAMs, but impose a consistency constraint that tries to obtain similar CAMs for original and transformed images. Finally, [2] flattens out the negative class scores in the logit-space to achieve adversarial robustness. In contrast to these loss functions, we seek to remove peaks with in the CAMs for negative classes, thus ensuring that implicit CNN attention conforms with the information available from ground-truth labels. We extensively compare our approach with the above mentioned loss functions and demonstrate significant improvements.

3 Method

In this section, we first introduce class activation maps, and then give a detailed description of our loss, followed by gradients analysis. Finally, we show the comparative analysis between two proposed variants of HNC loss.

3.1 Class Activation Maps

Consider the multi-class image classification task with n classes. Let I be a training image with ground-truth label $l \in J$, where $J = \{1, 2, ..., n\}$ is the label set. Let $\boldsymbol{F} \in \mathbb{R}^{c \times h \times w}$ denotes the high-level feature maps, output from network's last convolution layer, where c, h, w denote number of channels, height and width of the feature maps, respectively. After passing $\boldsymbol{F}$ through a global average pooling (GAP) layer and a fully connected (FC) layer with weight matrix $\boldsymbol{W} \in \mathbb{R}^{c \times n}$, class confidence scores $\boldsymbol{s} = \{s_i : i \in J\} \in \mathbb{R}^n$ are obtained to make the final predictions.

Class activation mapping [46] is a simple visualization approach that has shown great potential in localizing discriminative regions corresponding to a class. As a result, it has been used in both weakly supervised and fully supervised settings for a variety of tasks, such as classification [46], object localization [3,49], segmentation [34,47] and counting [4]. As shown in [46], we can simply convolve the feature maps $\boldsymbol{F}$ and $\boldsymbol{W}$ to obtain class activation maps $\boldsymbol{M} \in \mathbb{R}^{n \times h \times w}$,

$$\boldsymbol{M}_o = \sum_k w_{k,o} \boldsymbol{F}_k, \tag{1}$$

where $\boldsymbol{M}_o \in \mathbb{R}^{h \times w}$ is the class activation map (CAM) corresponding to an output class 'o', $w_{k,o}$ is the element in the k^{th} row and o^{th} column of matrix $\boldsymbol{W}$, and $\boldsymbol{F}_k \in \mathbb{R}^{h \times w}$ is the feature map corresponding to the k^{th} channel. For simplicity, the bias term is omitted in Eq. 1.

Most previous works use the CAM of the positive class, e.g., as a clue for coarse object localization [1,4], and ignore the CAMs for negative classes. However, we find that there are many false peaks in negative CAMs as showed in Fig. 1, which in turn negatively affect the classification performance resulting in false positives. Following this intuition, we propose a novel loss to suppress the highly activated regions in the negative CAMs.

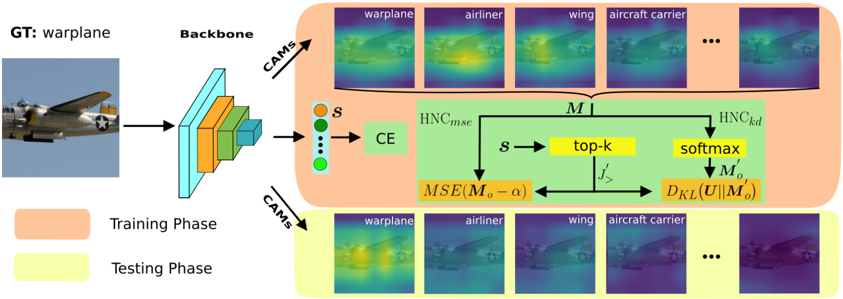

Fig. 2. Overview of our proposed *Homogeneous Negative CAM* loss. From *left* to *right*, a positive class activation map (CAM) followed by a group of negative CAMs is shown. Negative CAMs are ranked based on their classification scores. Our proposed loss (two variants: HNC$_{mse}$ and HNC$_{kd}$) is designed to suppress the false activations in the negative CAMs. As shown on top, during the early training phase, there are several false peaks in the negative CAMs. After training with our loss, the negative CAMs generated during the inference stage are flattened out, which leads to a correct prediction.

3.2 Our Proposed Loss

Our Idea. The basic idea of our '*Homogeneous Negative CAM*' (HNC) loss is to suppress the false activations on the class activation maps for negative classes (see Fig. 2). When the positive class and the negative class are very different (*e.g.*, plane vs. tree), it is understandable that the CAM for a negative class should not be focused on any particular region. For the situation where the positive and negative classes are similar (*e.g.*, warplane vs. airliner), our loss remains relevant. By using our proposed loss, the CAM for the negative class is forced to be *relatively* clearer (less peaks), helping the network to resolve the confusions between similar classes and leading to correct prediction. Suppressing the false activations in the CAMs thus provides additional supervision to the network, compared to the normal CE loss which only suppresses the class confidence scores for negative classes (average of negative CAMs).

We develop two alternatives for enforcing homogeneity in negative CAMs. The first approach simply uses the Mean Square Error (MSE) loss to suppress the peak responses, while the second approach minimizes the KL-divergence between negative CAMs and a uniform map. We refer to these two approaches as the HNC$_{mse}$ loss and HNC$_{kd}$ loss, respectively.

HNC$_{mse}$ Loss. The general idea is to suppress the CAMs for the top-k negative classes (with k highest confidence scores) using the MSE loss.

We define $J^{'}$ as the set of all negative classes: $J^{'} = \{i : i \in J \wedge i \neq l\}$. Let $\boldsymbol{s}' = \{s_i : i \in J^{'}\}$ be the set containing the confidence scores of all negative classes. We compute the k^{th} highest values of $\boldsymbol{s}'$ and denote it as t_k. Next, we obtain $J^{'}_{>}$ by thresholding $\boldsymbol{s}$ using t_k, defined as follows:

$$J^{'}_{>} = \{i : i \in J^{'} \wedge s_i \geq t_k\}$$

where, $J^{'}_{>}$ contains the negative classes whose confidence scores are within the top-k of all negative classes. Then, our HNC$_{mse}$ loss is defined as follows:

$$\mathrm{HNC}_{mse}(\boldsymbol{M}, l) = \frac{1}{hw} \sum_{o \in J^{'}_{>}} \sum_{i,j} \Big(\boldsymbol{M}_o(i,j) - \alpha\Big)^2, \tag{2}$$

where i, j denote the indices and α is the constant towards which the peaks in $\boldsymbol{M}_o$ are suppressed and we set it as 0 for all $o \in J^{'}_{>}$.

HNC$_{kd}$ Loss. In an ideal situation, the negative CAMs should be clear, providing no focused region for negative classes. Thus, we propose to force the top-k negative CAMs to have a uniform spatial distribution. Let $\boldsymbol{U} \in \mathbb{R}^{h \times w}$ be a uniform probability matrix with all elements equal to $1/(hw)$. Our HNC$_{kd}$ loss minimizes the KL-divergence between the negative CAMs and $\boldsymbol{U}$:

$$\mathrm{HNC}_{kd}(\boldsymbol{M}, l) = \sum_{o \in J^{'}_{>}} D_{KL}\big(\boldsymbol{U} || \boldsymbol{M}^{'}_o\big), \tag{3}$$

where $\boldsymbol{M}^{'}_o = \sigma(\boldsymbol{M}_o)$ and σ is the softmax activation function to convert $\boldsymbol{M}_o$ to a probability map. We denote,

$$D_{KL}(\boldsymbol{U} || \boldsymbol{M}^{'}_o) = \sum_{i,j} \boldsymbol{U}(i,j) \log \frac{\boldsymbol{U}(i,j)}{\boldsymbol{M}^{'}_o(i,j)} = const - \frac{1}{hw} \sum_{i,j} \log\big(\boldsymbol{M}^{'}_o(i,j)\big), \tag{4}$$

where *const* is a constant. After removing the constant and combining Eq. 3 and Eq. 4, we get the HNC$_{kd}$ loss as:

$$\mathrm{HNC}_{kd}(\boldsymbol{M}, l) = -\frac{1}{hw} \sum_{o \in J^{'}_{>}} \sum_{i,j} \log\big(\boldsymbol{M}^{'}_o(i,j)\big). \tag{5}$$

Overall Loss. The overall loss is the weighted combination of cross entropy and HNC losses. We note that our proposed loss can also be used together with other image classification losses, e.g., Focal loss [23] and LGM loss [36]. In this work, we stick with combining our loss function with the basic cross entropy loss to demonstrate the *concept* (*idea*) and clearly show its benefit. Hence, cross entropy loss is the fair baseline and used frequently in our experiments (§4). As shown by our results (§4), this combination works well on various tasks and datasets. The cross entropy (CE) loss is defined on class confidence scores $\boldsymbol{s}$ as:

$$\mathrm{CE}(\boldsymbol{s}, l) = -\log \frac{\exp(s_l)}{\sum_{i \in J} \exp(s_i)}, \tag{6}$$

where s_i is the i^{th} element of $\boldsymbol{s}$. The overall loss is defined as follows:

$$\mathcal{L}_{cl}(\boldsymbol{s}, \boldsymbol{M}, l) = \mathrm{CE}(\boldsymbol{s}, l) + \lambda\ \mathrm{HNC}(\boldsymbol{M}, l). \tag{7}$$

Here, λ is the hyper-parameter controlling the weight of the HNC loss, which can be implemented according to Eq. 2 or Eq. 5.

Multi-label Classification. For multi-label classification, we adopt the *weighted sigmoid cross-entropy* (SCE) loss, as in [22]. For an image I, let $\boldsymbol{l} = \{l_i \in J\}$ denotes the set containing all ground-truth classes. Then, the loss function is,

$$\begin{aligned} \mathrm{SCE}(\boldsymbol{s}, \boldsymbol{l}) &= -\frac{1}{n}\Big(\sum_{o \in \boldsymbol{l}} u_o \log \frac{1}{1+\exp(-s_o)} + \sum_{o \notin \boldsymbol{l}} u_o \log \frac{\exp(-s_o)}{1+\exp(-s_o)}\Big), \\ u_o &= \exp(1-p_o)[o \in \boldsymbol{l}] + \exp(p_o)[o \notin \boldsymbol{l}], \end{aligned} \tag{8}$$

where p_o is the probability of positive samples for class 'o' in the training set. Despite the SCE loss being used, we can generate class activation maps and $J_{>}^{'}$ in the same way as multi-class classification. The overall loss for multi-label classification ($\mathcal{L}_{mlc}$) is as follows:

$$\mathcal{L}_{mlc}(\boldsymbol{s}, \boldsymbol{M}, \boldsymbol{l}) = \mathrm{SCE}(\boldsymbol{s}, \boldsymbol{l}) + \lambda \ \mathrm{HNC}(\boldsymbol{M}, \boldsymbol{l}). \tag{9}$$

Here, the HNC loss can be implemented according to Eq. 2 or Eq. 5.

3.3 Gradient Analysis

We consider the overall loss given by Eq. 7. Since, $s_o = \frac{1}{hw}\sum_{i,j} \boldsymbol{M}_o(i,j)$, we can compute the derivative of the overall loss with respect to $\boldsymbol{M}_o(i,j)$ to obtain the gradient formulae denoting the effect of change in class-activation maps on the net loss. For simplicity, we write $\boldsymbol{M}_o(i,j)$ as $\boldsymbol{M}_o^{i,j}$ here. First, for the cross entropy loss, by chain rule:

$$\begin{aligned} \frac{\partial \mathrm{CE}(\boldsymbol{s}, l)}{\partial \boldsymbol{M}_o^{i,j}} &= \frac{\partial \mathrm{CE}}{\partial s_o} \cdot \frac{\partial s_o}{\partial \boldsymbol{M}_o^{i,j}}, \text{ where } \frac{\partial \mathrm{CE}}{\partial s_o} = \frac{\exp(s_o)}{\sum_k \exp(s_k)} - y_o, \frac{\partial s_o}{\partial \boldsymbol{M}_o^{i,j}} = \frac{1}{hw}, \\ \frac{\partial \mathrm{CE}(\boldsymbol{s}, l)}{\partial \boldsymbol{M}_o^{i,j}} &= \beta \frac{\exp(\beta \sum_{i,j} \boldsymbol{M}_o^{i,j})}{\sum_k \exp(\beta \sum_{i,j} \boldsymbol{M}_k^{i,j})} - \beta y_o, \end{aligned} \tag{10}$$

where $\boldsymbol{y}$ is a one-hot encoded vector and $\beta = \frac{1}{hw}$. Similarly for HNC_{mse},

$$\frac{\partial \mathrm{HNC}_{mse}}{\partial \boldsymbol{M}_o^{i,j}} = 2\beta(\boldsymbol{M}_o^{i,j} - \alpha). \tag{11}$$

For the KL divergence, the derivative is given by:

$$\frac{\partial \mathrm{HNC}_{kl}}{\partial \boldsymbol{M}_o^{i,j}} = \frac{\exp(\boldsymbol{M}_o^{i,j})}{\sum_{i',j'} \exp(\boldsymbol{M}_o^{i',j'})} - \beta. \tag{12}$$

Discussion. For cross entropy loss, we observe that the gradient for every location of the class activation map $\boldsymbol{M}_o$, is always the *same* regardless of the pixel intensities, as seen from Eq. 10. It is expected, since the CE works directly on the class confidence scores, which is the average of $\boldsymbol{M}_o$. However, for our loss, the gradients (Eq. 11 and Eq. 12) for different locations of the $\boldsymbol{M}_o$ can be *different.* Specifically, a false peak region of the top-k negative CAMs has higher gradients

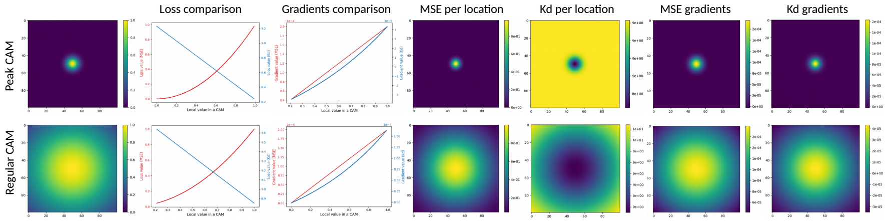

Fig. 3. Comparison between HNC$_{mse}$ and HNC$_{kd}$. Peak and Regular CAM cases are shown as toy examples to illustrate the behavior of both losses. Both pixel maps (columns 4-7) and line plots (columns 2-3) are shown. (*best viewed with zoom*)

than non-peaked regions. As a result, those regions would be suppressed more, helping network to better differentiate the positive class and those top-k negative classes (the most confusing ones).

One may argue that when using a normal CE loss, some regions of the negative CAMs can be very negative and our loss can have the effect of further increasing those negative values. Note that this scenario is unlikely because our loss is used together with CE loss. CE loss pushes the overall scores of negative CAMs to be small while our loss pushes the negative CAMs to be homogeneous (without peaks). The overall effect is that peaks will be suppressed.

3.4 Comparison: HNC$_{mse}$ vs. HNC$_{kd}$

In order to study the comparative nature of both proposed loss functions, we consider two typical cases for CAM. (a) A highly-focused CAM with a single peak region (*peak CAM*). (b) A normal case where the CAM is neither too focused nor too spread out (*regular CAM*). These two example cases are shown from top to bottom in Fig. 3. For each case, we illustrate a comparison between loss and gradient values for HNC$_{mse}$ and HNC$_{kd}$ loss functions. From the gradients maps (last two columns in Fig. 3), we clearly observe that for our loss, different locations of the CAMs can have different gradients, which is consistent with our analysis in §3.3. Below, we derive an alternate form for HNC$_{kd}$ that will help us better understand the comparison between the two variants.

Proposition 1. *The minimization of HNC$_{kd}$ is equivalent to minimizing the maximum (peak) value in $\boldsymbol{M}_o$, while simultaneously maximizing the average CAM response $\bar{M}_o$ (mean of $\boldsymbol{M}_o$) : $o \in J'_>$ to obtain a homogeneous CAM.*

Proof. Consider the loss defined in Eq. 5. By putting $\boldsymbol{M}'_o = \sigma(\boldsymbol{M}_o)$ and simplifying, we get:

$$\mathrm{HNC}_{kd}(\boldsymbol{M}, l) = \sum_{o \in J'_>} \Big[\log \sum_{i',j'} \exp(\boldsymbol{M}_o^{i',j'}) - \bar{M}_o \Big],$$

where, the first term on the right is the Log-Sum-Exp (LSE) function, which is a smooth approximation of the max operation. Then, since $\text{LSE}(\boldsymbol{M}_o) > \max(\boldsymbol{M}_o)$, HNC_{kd} acts as an upper bound for the following expression:

$$\text{HNC}_{kd}(\boldsymbol{M}, l) > \sum_{o \in J'_{>}} \left[\max(\boldsymbol{M}_o) - \bar{M}_o\right].$$

As a result, when minimizing HNC_{kd}, we are effectively reducing the peak values in negative CAMs, while simultaneously maximizing the average CAM response. □

The above proposition shows that the loss values for HNC_{kd} follow a linear relation with the local values in the input CAM. On the other hand, HNC_{mse} imposes a quadratic penalty that focuses more on the extreme values. Thereby, HNC_{mse} is relatively more sensitive to outliers in the CAM, while HNC_{kd} applies a relatively smoother penalty. In our experiments, we notice nearly similar performance from both HNC_{kd} and HNC_{mse}.

4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our proposed loss function. Specifically, we evaluate our loss on general image recognition (see §4.1), multi-label classification (see §4.2), fine-grained classification (see §4.3), adversarial robustness (see §4.4), and noisy label learning (see §4.5). Then, ablation studies are performed in §4.6. All experiments are carried out using the Pytorch framework on NVIDIA Tesla V100 GPUs.

4.1 General Image Classification

For the task of general image classification, we evaluate our loss on the CIFAR-100 [20] and ImageNet (ILSVRC 2012) [6]. Below, we summarize our results.
CIFAR-100 Classification: CIFAR-100 consists of 60,000 images in total. Among these images, 50,000 are used for training while the remaining 10,000 are used for testing. CIFAR-100 has a total of 100 classes, each with 600 images. The results are averaged over 5 runs.

We train two backbone networks, ResNet-56 and ResNet-110, from scratch with our loss. We use input images with the original resolution after standard data augmentation, i.e., random flipping and cropping with a padding of 4 pixels on each side. The learning rate is initially set to 0.1, and dropped by a factor of 0.1 at 84 and 122 epochs. We train our model for 164 epochs in total.

Table 1 shows the comparisons between our loss and other recent or top-performing loss functions including Center loss [39], Large-margin Gaussian Mixture (LGM) loss [36], Focal loss [23], Class-balanced (CB) Focal loss [5], Angular softmax (A-Softmax) loss [25], Large-margin cosine (LMC) loss [38], Additive Angular Margin (AAM) loss [7], and Anchor loss [29]. Among these loss functions, [2, 7, 25, 36, 38] focus on margin maximization between classes to enhance

Loss Functions	Publication	ResNet-56		ResNet-110	
		Top-1	Top-5	Top-1	Top-5
Cross Entropy	-	72.40	92.68	73.79	93.11
Center Loss [39]	ECCV16	72.72	93.06	74.27	93.20
LGM Loss [36]	CVPR18	73.08	93.10	74.34	93.06
Focal Loss [23]	ICCV17	73.09	93.07	74.34	93.34
CB Focal Loss [5]	CVPR19	73.09	93.07	74.34	93.34
A-Softmax Loss [25]	CVPR17	72.20	91.28	72.72	90.41
LMC Loss [38]	CVPR17	71.52	91.64	73.15	91.88
AAM Loss [7]	CVPR19	71.41	91.66	73.72	91.86
Anchor Loss* [29]	ICCV19	-	-	74.38	92.45
Ours (HNC$_{mse}$)	-	73.35	93.11	**75.00**	93.58
Ours (HNC$_{kd}$)	-	**73.47**	**93.29**	74.76	**93.65**

Table 1. Performance comparisons between different loss functions on CIFAR-100. *: the number is taken from the corresponding paper. Results show that our loss outperforms other losses by a clear margin.

Loss Functions	ResNet-101		ResNet-152	
	Top-1	Top-5	Top-1	Top-5
CE (reproduced)	23.2	6.7	22.9	6.6
LGM* [36]	22.7	7.1	-	-
Ours (HNC$_{mse}$)	22.3	**6.4**	21.9	6.1
Ours (HNC$_{kd}$)	**22.1**	**6.4**	**21.8**	**6.0**

Table 2. Error rates of different losses on ImageNet. For ResNet-101, our loss outperforms the baseline by 1.1% in terms of Top-1 error.

Loss Functions	Top-1
CE	86.0
Center Loss [39]	86.5
Focal Loss [23]	85.8
Ours (HNC$_{mse}$)	**87.1**
Ours (HNC$_{kd}$)	86.9

Table 3. Accuracy of different losses with ResNet-50 on CUB-200-2011. Ours surpasses CE by 1.1%.

the performance, [39] performs clustering and [5, 23, 29] focus on discriminating hard examples. In contrast, our approach develops a simple constraint for intermediate CAMs of negative classes.

The results show that our loss clearly outperforms other methods. Remarkably, compared to the CE loss, our loss achieves 1.07% and 1.21% improvements (top-1 accuracy) on ResNet-56 and ResNet-110, respectively. The fact that our loss has a larger margin over CE using ResNet-110 than ResNet-56 is possibly due to the higher redundancy in a larger network, which can lead to more serious over-fitting. Among other loss functions, both the LGM loss [36] and Focal loss [23] perform well, but are inferior to our loss for both ResNet-56 and ResNet-110. Note that CB Focal [5] was designed for targeting class-imbalance in the training set. For CIFAR-100, since all classes have the same number of images, CB Focal performs as well as the Focal loss. The loss functions that operate on the hyper-sphere manifold [7, 25, 38] perform a bit lower which demonstrates the manifold assumption does not hold true for CIFAR-100.

ImageNet Classification: ImageNet [6] is a large-scale dataset for visual recognition. It contains ∼1.2 million training and 50,000 validation images.

We train ResNet-101 and ResNet-152 with the proposed loss. Basically, input image is random cropped to size of 224 × 224 by scale and aspect ratio. Following [40], an initial learning rate of 0.1 is used and dropped by a factor of 0.1 after every 30 epochs. We use a weight decay of 0.0001 and a momentum of 0.9. The

Method		All						
		mAP	F1-C	P-C	R-C	F1-O	P-O	R-O
ResNet-101† [48]		75.2	69.5	80.8	63.4	74.4	82.1	68.0
ResNet-101-SRN* [48]		77.1	71.2	81.6	65.4	75.8	82.7	69.9
Baseline	ResNet-101*	74.9	69.7	70.1	69.7	73.7	73.6	73.7
Ours	ResNet-101+HNC$_{mse}$	**77.8**	**72.3**	**78.9**	67.4	**76.5**	**81.9**	71.9
	ResNet-101+HNC$_{kd}$	77.6	**72.3**	75.8	**69.7**	76.1	78.2	**74.1**
Baseline	AC* [15]	77.5	72.2	77.4	68.3	76.3	79.8	73.1
Ours	AC+HNC$_{mse}$	**78.5**	72.8	**79.6**	67.9	**76.9**	**82.3**	72.1
	AC+HNC$_{kd}$	78.2	**72.9**	76.8	**70.0**	76.6	78.6	**74.7**

Table 4. Comparisons between methods w/ and w/o our loss on MS-COCO using different metrics. 'ResNet-101†' represents the baseline implemented with complex data augmentations in [48] and * means the number is taken from [15]. Our loss provides a gain over both basic and strong baselines.

training is terminated at 120 epochs. Training is conducted on 8 Tesla V100 GPUs, using a total batch size of 256. To make fair comparison, all models are trained under the same strategy, unless specifically stated.

Table 2 shows results of our loss on ImageNet. Though our loss is simple, it proves very effective for large-scale recognition task. By simply replacing the CE with HNC loss, the error rate is reduced by a margin of 1.1% on both ResNet-101 and ResNet-152. Our loss also outperforms the recently proposed LGM loss [36], which is based on the assumption that deep features follow a Gaussian Mixture distribution. Note that, similar to CIFAR-100, both variants of our loss give comparable results.

4.2 Multi-label Classification

We conduct multi-label classification experiments on MS-COCO dataset [24]. It contains 82,783 training and 40,504 validation images, annotated with 80 labels. Since ground-truth labels are not available for the test set, we train our network on the training set and evaluate it on the validation set, following [15]. Our loss is tested on the official implementation of [15].

We follow the same training strategy as in [15]. Namely, an input size of 288 × 288 is used, and we fine-tune ResNet-101, pretrained on the ImageNet dataset. The initial learning rate is 0.001, and dropped with a factor 0.1 after 6 and 8 epochs. Following other works in multi-label classification [15, 43], the evaluation metrics we choose are: mean Average Precision (mAP), as well as macro and micro precision/recall/F1-score (denoted as P-C, R-C, F1-C, P-O, R-O, F1-O, respectively). For details of these metrics, we refer to [43].

The performance comparisons between HNC and baselines are shown in Table 4. In terms of mAP, both HNC$_{mse}$ and HNC$_{kd}$ outperform the baselines. Specifically, for the baseline ResNet-101, HNC$_{mse}$ achieves a gain of 2.9% in mAP. For the stronger baseline method named Attention Consistency (AC) [15], HNC$_{mse}$ is also superior and achieves a 1.0% increment. Notably, the AC approach is also a loss working on the CAMs. It forces the transformed CAMs of original images to be consistent with the CAMs of the transformed images. Our loss is related to

Table 5. Performance comparison of our loss against FGSM attack with different perturbations (ϵ) on CIFAR-100 using ResNet-110 architecture.

ϵ	CE	Center Loss [39]	**Ours** (HNC$_{mse}$)	**Ours** (HNC$_{kd}$)
0.05	14.55	14.76	23.92	**29.80**
0.1	9.89	10.42	17.20	**21.80**
0.2	6.15	7.17	11.01	**12.96**
0.3	3.79	5.54	6.90	**7.06**

Noise type	r	0.2	0.4	0.6	0.8
symmetric	CE	51.98	38.76	22.48	9.16
	HNC$_{mse}$	**58.98**	**48.03**	**32.86**	**14.73**
	HNC$_{kd}$	56.59	44.86	29.15	12.20

(a)

Noise type	r	0.1	0.2	0.3	0.4
asymmetric	CE	63.10	56.60	49.33	40.89
	HNC$_{mse}$	**67.18**	**62.91**	**56.12**	**46.51**
	HNC$_{kd}$	65.33	59.72	51.98	43.04

(b)

Table 6. CIFAR-100 results with symmetric **(a)** and asymmetric **(b)** noise.

AC since both can reduce the over-fitting (due to redundancy) in the network. But AC does not explicitly consider the negative CAMs, which have many false activations and need to be suppressed.

4.3 Fine-grained Classification

For fine-grained classification, we evaluate our loss on the CUB-200-2011 dataset [35], which is widely used for this task. It contains 5,994 training and 5,794 test images, each of which belongs to one of 200 bird classes.

We fine-tune ResNet-50, which is pretrained on the ImageNet dataset. The initial learning rate is set to 0.001 and reduced by 0.1 after 50 epochs. A total batch size of 16 is used and the model is trained using 2 GPUs.

Table 3 shows the results of different losses with ResNet-50 on CUB-200-2011. Both HNC$_{mse}$ and HNC$_{kd}$ outperform the baseline (ResNet-50 with CE loss), by a margin of 1.1% and 0.9%, respectively. Remarkably, our proposed loss also outperforms other losses, including Center loss [39], and Focal loss [23].

4.4 Adversarial Robustness

Since our proposed loss suppresses negative CAMs, we anticipate this strategy to be helpful against adversarial attacks. Adversarial examples are generated by intentionally adding small but imperceptible perturbations to the inputs, which cause the model to make wrong predictions with high confidence [14]. We consider the most challenging attack case, i.e., the white-box attack, where all the model parameters and training details are known to the adversary. Specifically, we use the fast gradient sign method (FGSM) [14], which adopts the gradient back-propagated from the training loss to determine the direction of the perturbation. An adversarial example I^* is generated by: $I^* = I + \epsilon \cdot sign(\nabla_I L(I, l))$, where ϵ is the magnitude of perturbation, I is the input image, l is the ground-truth label for the input, and $L(I, l)$ is the classification loss function. We select $\epsilon \in \{0.05, 0.1, 0.2, 0.3\}$ for our experiments.

	Setting	CE	**Ours** (HNC)
ResNet-110	-	73.79	**75.00**
DenseNet-BC	k=12, d=100	77.32	**78.78**
ResNeXt-29	c=8, d=64	81.77	**82.32**
ResNeXt-29	c=16, d=64	81.98	**82.81**

Table 7. Comparison between HNC and CE on various networks. Our loss outperforms baseline for all considered architectures.

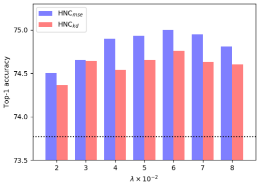

Fig. 4. Top-1 accuracy of HNC for different λ. Dotted line shows the CE loss.

Table 5 shows performance of different losses under FGSM attack on CIFAR-100 with ResNet-110. We compare with Center loss and CE loss. For all considered ϵ, our loss is more robust than the others.

The higher robustness of HNC loss is potentially because it constraints the intermediate activations that have been shown to provide better deterrence against perturbations [28, 41]. Interestingly, we found HNC$_{kd}$ to be considerably better than HNC$_{mse}$ in this task. This is primarily due to the reason that HNC$_{mse}$ focuses on the outliers, thus adversarial noise that is generally low in strength can sneak in easily. In contrast, HNC$_{kd}$ gives an equal penalty to all deviated negative CAM values, thus blocking away the maliciously crafted perturbations.

4.5 Learning from Noisy Labels

Here, we show the effectiveness of our loss for learning from noisy labels. This area has recently attracted lots of research attention. We test on CIFAR-100 using ResNet-110 following the training strategy described in §4.1. We use both the symmetric noise setting, where label noise is uniformly distributed among all categories with probability r, and asymmetric noise setting, where each ground-truth class is flipped to the next class circularly with probability r. Thus, $r \in [0, 1]$ denotes the noise rate. Following [21], we choose r of 0.2, 0.4, 0.6, and 0.8 for symmetric noise, and r of 0.1, 0.2, 0.3, 0.4 for asymmetric noise. Following [45], we retain 10% of training data as validation set.

The results are shown in Table 6 where we report test accuracy of the last epoch. Our HNC loss outperforms the baseline (CE) with a large margin. Remarkably, for symmetric noise with noise rate 0.6, our loss obtains a absolute improvement of 10.38%, compared with the baseline.

4.6 Ablation Study

We conduct ablation studies on the CIFAR-100. Firstly, we show how λ, the factor balancing the CE and HNC, affects the accuracy. Fig. 4 shows the change in performance with respect to λ. For both HNC$_{mse}$ and HNC$_{kd}$, the effect of λ follows a similar trend. In the beginning, classification accuracy increases when λ is increased. This is potentially because the negative CAMs are more suppressed and thus become smoother when λ increases. However, after a certain

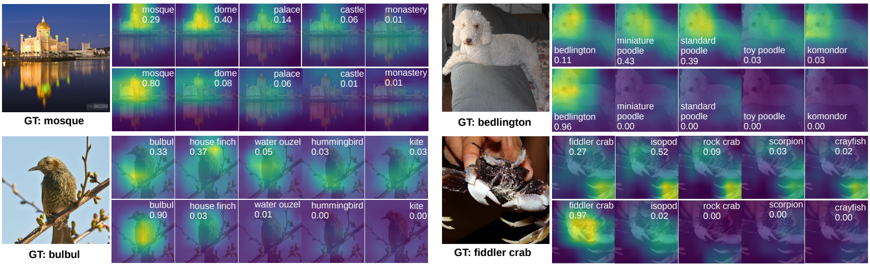

Fig. 5. Qualitative comparisons of CAMs between our method and baseline (CE), for images from ImageNet [6]. For each sample, we first show original image (left), followed by the CAMs of the baseline (top) and our loss (below). The CAMs follow the sequence: the positive CAM on left and then top-4 negative CAMs (ranked by the score). For each CAM, the class name and probability are shown in white. For all considered cases, our loss clears the negative CAMs, thus leading to the correct prediction.

threshold, the performance drops as λ is further increased. This is because with very large λ, the relative weight of the CE loss is lower and the network loses focus on classification task. For all considered values of λ, our loss outperforms the baseline (CE loss), which again validates the soundness of our proposal.

We also compare HNC performance across various architectures. Since HNC$_{mse}$ and HNC$_{kd}$ performs similarly, we compare HNC$_{mse}$ with CE in Table 7. Our loss outperforms CE for different architectures, which shows its effectiveness.

4.7 Qualitative Results and Analysis

We show the qualitative results in Fig. 5. The main effect of our loss on CAMs can be summarized into two cases: (i) clearing the negative CAMs and locating a similar discriminative region as the baseline (top samples in Fig. 5); (ii) clearing the negative CAMs but locating a totally different and more discriminative region (below samples in Fig. 5). We conjecture that if our constraints can be satisfied when locating a similar region, the first case happens. However, if the positive class and negative classes are very difficult to separate, i.e. the relevant region located by the baseline is not discriminative enough, then the network is forced to find a different and more informative region to classify objects.

5 Conclusion

In this paper, we propose a novel loss (HNC) to suppress false activations in the negative CAMs. Its effectiveness is demonstrated by extensive experiments on various tasks: generic image recognition, multi-label classification, fine-grained classification, adversarial robustness, and learning from noisy labels. We observe that our loss successfully clears the negative CAMs and leads to consistent visualizations and improved performance across all studied tasks.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: CVPR (2019) 4
2. Chen, H.Y., Liang, J.H., Chang, S.C., Pan, J.Y., Chen, Y.T., Wei, W., Juan, D.C.: Improving adversarial robustness via guided complement entropy. ICCV (2019) 3, 4, 9
3. Choe, J., Oh, S.J., Lee, S., Chun, S., Akata, Z., Shim, H.: Evaluating weakly supervised object localization methods right. In: CVPR (2020) 4
4. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: CVPR (2019) 4
5. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR (2019) 9, 10
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 2, 9, 10, 14
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019) 3, 9, 10
8. Dong, Q., Gong, S., Zhu, X.: Class rectification hard mining for imbalanced deep learning. In: ICCV (2017) 3
9. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: CVPR (2016) 3
10. Dubey, A., Gupta, O., Raskar, R., Naik, N.: Maximum-entropy fine grained classification. In: NeurIPS (2018) 3
11. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341**(3), 1 (2009) 3
12. Gong, Y., Jia, Y., Leung, T.K., Toshev, A., Ioffe, S.: Deep convolutional ranking for multi label image annotation. In: ICLR (2014) 3
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016) 3
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2015) 12
15. Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S.: Visual attention consistency under image transforms for multi-label image classification. In: CVPR (2019) 3, 4, 11
16. Hayat, M., Khan, S., Zamir, S.W., Shen, J., Shao, L.: Gaussian affinity for max-margin class imbalanced learning. ICCV (2019) 3
17. Huang, C., Li, Y., Change Loy, C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR (2016) 3
18. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Transactions on Neural Networks and Learning Systems **29**(8), 3573–3587 (2018) 3
19. Khan, S., Rahmani, H., Shah, S.A.A., Bennamoun, M.: A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision **8**(1), 1–207 (2018) 1
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 9
21. Kun, Y., Jianxin, W.: Probabilistic End-to-end Noise Correction for Learning with Noisy Labels. In: CVPR (2019) 13
22. Li, D., Chen, X., Huang, K.: Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In: ACPR (2015) 7
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 3, 6, 9, 10, 12

24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 11
25. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: CVPR (2017) 3, 9, 10
26. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: ICML (2016) 3
27. Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. IJCV **120**(3), 233–255 (2016) 3
28. Mustafa, A., Khan, S., Hayat, M., Goecke, R., Shen, J., Shao, L.: Adversarial defense by restricting the hidden space of deep neural networks. ICCV (2019) 13
29. Ryou, S., Jeong, S.G., Perona, P.: Anchor loss: Modulating loss scale based on prediction difficulty. In: ICCV (2019) 3, 9, 10
30. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015) 1
31. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017) 2, 3
32. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013) 2, 3
33. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014) 3
34. Sun, G., Wang, W., Dai, J., Van Gool, L.: Mining cross-image semantics for weakly supervised semantic segmentation. arXiv preprint (2020) 4
35. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 12
36. Wan, W., Zhong, Y., Li, T., Chen, J.: Rethinking feature distribution for loss functions in image classification. In: CVPR (2018) 6, 9, 10, 11
37. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Processing Letters **25**(7), 926–930 (2018) 3
38. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: CVPR (2018) 9, 10
39. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016) 9, 10, 12
40. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: ECCV (2018) 10
41. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (2019) 13
42. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014) 2, 3
43. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. TKDE **26**(8), 1819–1837 (2013) 11
44. Zhang, Y., Gong, B., Shah, M.: Fast zero-shot image tagging. In: CVPR (2016) 3
45. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: NeurIPS (2018) 13
46. Zhou, B., Khosla, A., A., L., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. CVPR (2016) 1, 2, 3, 4
47. Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., Jiao, J.: Weakly supervised instance segmentation using class peak response. In: CVPR (2018) 4

48. Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with image-level supervisions for multi-label image classification. In: CVPR (2017) 11
49. Zhu, Y., Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Soft proposal networks for weakly supervised object localization. In: ICCV (2017) 4