Soft Augmentation for Image Classification

Yang Liu, Shen Yan, Laura Leal-Taixé, James Hays, Deva Ramanan
Argo AI
youngleoel@gmail.com, shenyan@google.com, leal.taixe@tum.de, hays@gatech.edu, deva@cs.cmu.edu

Abstract

Modern neural networks are over-parameterized and thus rely on strong regularization such as data augmentation and weight decay to reduce overfitting and improve generalization. The dominant form of data augmentation applies invariant transforms, where the learning target of a sample is invariant to the transform applied to that sample. We draw inspiration from human visual classification studies and propose generalizing augmentation with invariant transforms to soft augmentation where the ***learning target softens non-linearly as a function of the degree of the transform*** *applied to the sample: e.g., more aggressive image crop augmentations produce less confident learning targets. We demonstrate that soft targets allow for more aggressive data augmentation, offer more robust performance boosts, work with other augmentation policies, and interestingly, produce better calibrated models (since they are trained to be less confident on aggressively cropped/occluded examples). Combined with existing aggressive augmentation strategies, soft targets 1)* **double** *the top-1 accuracy boost across Cifar-10, Cifar-100, ImageNet-1K, and ImageNet-V2, 2) improve model occlusion performance by up to* $4\times$*, and 3)* **half** *the expected calibration error (ECE). Finally, we show that soft augmentation generalizes to self-supervised classification tasks. Code available at* https://github.com/youngleox/soft_augmentation

1. Introduction

Deep neural networks have enjoyed great success in the past decade in domains such as visual understanding [42], natural language processing [5], and protein structure prediction [41]. However, modern deep learning models are often over-parameterized and prone to overfitting. In addition to designing models with better inductive biases, strong regularization techniques such as weight decay and data augmentation are often necessary for neural networks to achieve ideal performance. Data augmentation is often a computationally cheap and effective way to regularize models and mitigate overfitting. The dominant form of data augmentation modifies training samples with invariant transforms – transformations of the data where it is assumed that the identity of the sample is *invariant* to the transforms.

Indeed, the notion of visual invariance is supported by evidence found from biological visual systems [54]. The robustness of human visual recognition has long been documented and inspired many learning methods including data augmentation and architectural improvement [19, 47]. This paper focuses on the counterpart of human visual robustness, namely **how our vision fails**. Instead of maintaining perfect invariance, human visual confidence degrades **non-linearly** as a function of the degree of transforms such as occlusion, likely as a result of information loss [44]. We propose modeling the transform-induced information loss for learned image classifiers and summarize the contributions as follows:

- We propose Soft Augmentation as a generalization of data augmentation with invariant transforms. With Soft Augmentation, the learning target of a transformed training sample *softens*. We empirically compare several softening strategies and prescribe a robust non-linear softening formula.
- With a frozen softening strategy, we show that replacing standard crop augmentation with soft crop augmentation allows for more aggressive augmentation, and **doubles** the top-1 accuracy boost of RandAugment [8] across Cifar-10, Cifar-100, ImageNet-1K, and ImageNet-V2.
- Soft Augmentation improves model occlusion robustness by achieving up to more than $4\times$ Top-1 accuracy boost on heavily occluded images.
- Combined with TrivialAugment [37], Soft Augmentation further reduces top-1 error and improves model calibration by reducing expected calibration error by more than **half**, outperforming 5-ensemble methods [25].
- In addition to supervised image classification models, Soft Augmentation also boosts the performance of self-supervised models, demonstrating its generalizability.

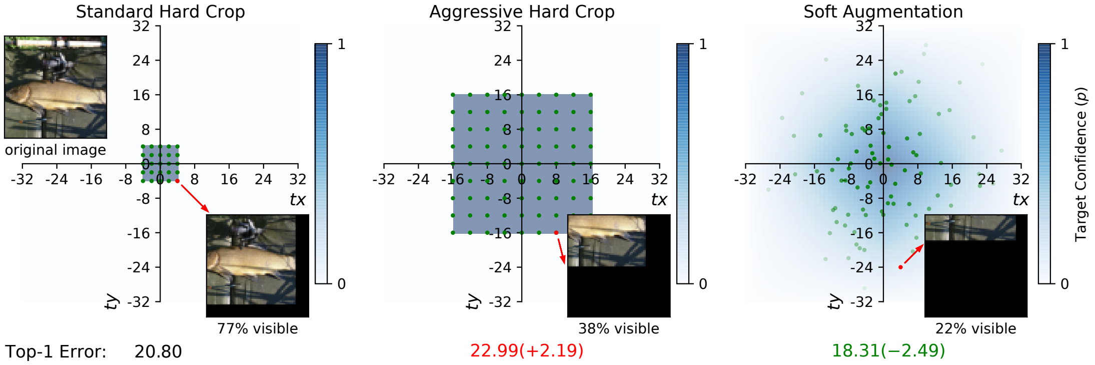


Figure 1. Traditional augmentation encourages invariance by requiring augmented samples to produce the same target label; we visualize the translational offset range (tx, ty) of Standard Hard Crop augmentations for 32×32 images from Cifar-100 on the **left**, reporting the top-1 error of a baseline ResNet-18. Naively increasing the augmentation range without reducing target confidence increases error (**middle**), but softening the target label by reducing the target confidence for extreme augmentations reduces the error (**right**), allowing for training with even more aggressive augmentations that may even produce *blank images*. Our work also shows that soft augmentations produce models that are more robust to occlusions (since they encounter larger occlusions during training) and models that are better calibrated (since they are trained to be less-confident on such occluded examples).

2. Related Work

2.1. Neural Networks for Vision

Since the seminal work from Krizhevsky *et al.* [24], neural networks have been the dominant class of high performing visual classifiers. Convolutional Neural Networks (CNNs) are a popular family of high performing neural models which borrows a simple idea of spatially local computations from biological vision [12, 18, 26]. With the help of architectural improvements [15], auxiliary loss [42], and improved computational power [13], deeper, larger, and more efficient neural nets have been developed in the past decade.

2.2. Data Augmentation

Data augmentation has been an essential regularizer for high performing neural networks in many domains including visual recognition. While many other regularization techniques such as weight decay [32] and batch normalization [4] are shown to be optional, we are aware of no competitive vision models that omit data augmentation.

Accompanying the influential AlexNet model, Krizhevsky *et al.* [24] proposed horizontal flipping and random cropping transforms which became the backbone of image data augmentation. Since the repertoire of invariant transformations has grown significantly in the past decade [42], choosing which subset to use and then finding the optimal hyperparameters for each transform has become computationally burdensome. This sparked a line of research [7, 28] which investigates optimal policies for data augmentation such as RandAugment [8] and TrivialAugment [37].

2.3. Learning from Soft Targets

While minimizing the cross entropy loss between model logits and hard one-hot targets remains the go-to recipe for supervised classification training, learning with *soft* targets has emerged in many lines of research. Label Smoothing [36, 43] is a straightforward method which applies a fixed smoothing (softening) factor α to the hard one-hot classification target. The motivation is that label smoothing prevents the model from becoming over-confident. Müller *et al.* [36] shows that label smoothing is related to knowledge distillation [17], where a student model learns the soft distribution of a (typically) larger teacher model.

A related line of research [49,53] focuses on regularizing how a model interpolates between samples by linearly mixing two or more samples and linearly softening the resulting learning targets. Mixing can be in the form of per-pixel blending [53] or patch-level recombination [49].

2.4. Robustness of Human Vision

Human visual classification is known to be robust against perturbations such as occlusion. In computer vision research, the robustness of human vision is often regarded as the gold standard for designing computer vision models [34, 54]. These findings indeed inspire development of robust vision models, such as compositional, recurrent, and occlusion aware models [22,46,47]. In addition to specialty models, much of the idea of using invariant transforms to augment training samples come from the intuition and observation that human vision are robust against these transforms such as object translation, scaling, occlusion, photometric distortions, etc.

Recent studies such as Tang *et al.* [44] indeed confirm the robustness of human visual recognition against mild to moderate perturbations. In a 5-class visual classification task, human subjects maintain high accuracy when up to approximately half of an object is occluded. However, the more interesting observation is that human performance starts to degenerate rapidly as occlusion increases and falls to chance level when the object is fully occluded (see Figure 2 right $k = 2, 3, 4$ for qualitative curves).

3. Soft Augmentation

In a typical supervised image classification setting, each training image x_i has a ground truth learning target y_i associated to it thus forming tuples:

$$(x_i, y_i), \tag{1}$$

where $x_i \in \mathbb{R}^{C\times W\times H}$ denotes the image and $y_i \in [0,1]^N$ denotes a N-dimensional one-hot vector representing the target label (Figure 2 left, "Hard Target"). As modern neural models have the capacity to memorize even large datasets [1], data augmentation mitigates the issue by hallucinating data points through transformations of existing training samples.

(Hard) data augmentation relies on the key underlying assumption that the augmented variant of x_i should maintain the original target label y_i:

$$(x_i, y_i) \Rightarrow (t_{\phi\sim S}(x_i), y_i)\,, \qquad \textbf{Hard Augmentation} \tag{2}$$

where $t_{\phi\sim S}(x_i)$ denotes the image transform applied to sample x_i, ϕ is a random sample from the fixed transform range S. Examples of image transforms include translation, rotation, crop, noise injection, etc. As shown by Tang *et al.* [44], transforms of x_i such as occlusion are approximately perceptually invariant only when ϕ is mild. Hence S often has to be carefully tuned in practice, since naively increasing it can lead to degraded performance (Figure 1). In the extreme case of 100% occlusion, total information loss occurs, making it detrimental for learning.

Label Smoothing applies a smoothing function g to the target label y_i parameterized by a handcrafted, fixed smoothing factor α. Specifically, label smoothing replaces the indicator value '1' (for the ground-truth class label) with $p = 1 - \alpha$, distributing the remaining α probability mass across all other class labels (Figure 2 left, "Soft Target"). One can interpret label smoothing as accounting for the *average* loss of information resulting from averaging over transforms from the range S. From this perspective, the smoothing factor α can be written as a function of the *fixed* transform range S:

$$(x_i, y_i) \Rightarrow \left(t_{\phi\sim S}(x_i), g_{\alpha(S)}(y_i)\right), \textbf{Label Smoothing} \tag{3}$$

Soft Augmentation, our proposed approach, can now be described succinctly as follows: replace the fixed smoothing factor $\alpha(S)$ with an adaptive smoothing factor $\alpha(\phi)$, that depends on the degree of the *specific* sampled augmentation ϕ applied to x_i:

$$(x_i, y_i) \Rightarrow \left(t_{\phi\sim S}(x_i), g_{\alpha(\phi)}(y_i)\right), \qquad \textbf{Soft Augmentation (Target)} \tag{4}$$

Crucially, conditioning on the information loss from a particular ϕ allows one to define far *larger* augmentation ranges S. We will show that such a strategy consistently produces robust performance improvements with little tuning across a variety of datasets, models, and augmentation strategies.

Extensions to Soft Augmentation may be proposed by also considering loss reweighting [40, 48], which is an alternative approach for softening the impact of an augmented example by down-weighting its contribution to the loss. To formalize this, let us write the training samples of a supervised dataset as triples including a weight factor w_i (that is typically initialized to all '1's). One can then re-purpose our smoothing function g to modify the weight instead of (or in addition to) the target label (Figure 2 left):

$$(x_i, y_i, w_i) \Rightarrow \left(t_{\phi\sim S}(x_i), y_i, g_{\alpha(\phi)}(w_i)\right), \qquad \textbf{Soft Augmentation (Weight)} \tag{5}$$

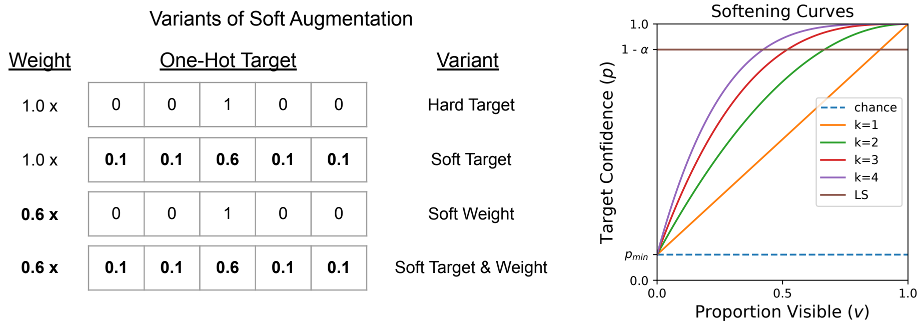
$$(x_i, y_i, w_i) \Rightarrow \left(t_{\phi\sim S}(x_i), g_{\alpha(\phi)}(y_i), g_{\alpha(\phi)}(w_i)\right). \qquad \textbf{Soft Augmentation (Target \& Weight)} \tag{6}$$

Finally, one may wish to soften targets by exploiting class-specific confusions when applying $\alpha(\phi)$; the smoothed target label of a highly-occluded truck example could place more probability mass on other vehicle classes, as opposed to distributing the remaining probability equally across all other classes. Such extensions are discussed in Section 5.

4. Experiments

4.1. Soft Augmentation with Crop

As a concrete example of the proposed Soft Augmentation, we consider the crop transform $t_{(tx,ty,w,h)}(x)$. In the case of 32×32 pixel Cifar images [23], the cropped images typically have a constant size $w = h = 32$, and $t(x)$ is fully parameterized by tx and ty, which are translational offsets between the cropped and the original image. As shown in



Variants of Soft Augmentation

Weight	One-Hot Target					Variant
1.0 x	0	0	1	0	0	Hard Target
1.0 x	**0.1**	**0.1**	**0.6**	**0.1**	**0.1**	Soft Target
0.6 x	0	0	1	0	0	Soft Weight
0.6 x	**0.1**	**0.1**	**0.6**	**0.1**	**0.1**	Soft Target & Weight

Figure 2. Variants of Soft Augmentation as prescribed by Equations 4 (*Soft Target*), 5 (*Soft Weight*), 6 (*Soft Target & Weight*) with example target confidence $p = 0.6$ (left). Soft Augmentation applies non-linear ($k = 2, 3, 4, \ldots$) softening to learning targets based on **the specific degree of occlusion of a cropped image** (Equation 7), which qualitatively captures the degradation of human visual recognition under occlusion [44]. Label Smoothing applies a **fixed** softening factor α to the one-hot classification target.

Figure 1 (left), the standard hard crop augmentation for the Cifar-10/100 classification tasks draws tx, ty independently from a uniform distribution of a modest range $U(-4, 4)$. Under this distribution, the minimal visibility of an image is 77% and ResNet-18 models trained on the Cifar-100 task achieve mean top-1 validation error of 20.80% across three independent runs (Figure 1 left). Naively applying aggressive hard crop augmentation drawn from a more aggressive range $U(-16, 16)$ *increases* top-1 error by 2.19% (Figure 1 middle). We make two changes to the standard crop augmentation.

We first propose drawing tx, ty independently from a scaled normal distribution $S* \sim N(0, \sigma L)$ (with clipping such that $|tx| < L, |ty| < L$), where L is the length of the longer edge of the image ($L = 32$ for Cifar). The distribution has zero mean and σ controls the relative spread of the distribution hence the mean occlusion level. Following the 3σ rule of normal distribution, an intuitive tuning-free choice is to set $\sigma \approx 0.3$, where $\sim 99\%$ of cropped samples have visibility ≥ 0. Figure 3 (left, $\alpha = 0$) shows that changing the distribution alone without target softening provides a moderate $\sim 0.4\%$ performance boost across crop strength σ.

Directly borrowing the findings from human vision reseach [44], one can define an adaptive softening $\alpha(tx, ty, k)$ that softens the ground truth learning target. Similar to Label Smoothing [43], a hard target can be softened to confidence $p \in [0, 1]$. Instead of a fixed α, consider a family of power functions that produces target hardness p given crop parameters tx, ty and curve shape k:

$$p = 1 - \alpha(tx, ty, k) = 1 - (1 - p_{min})(1 - v_{(tx,ty)})^k, \quad (7)$$

where $v_{(tx,ty)} \in [0, 1]$ is the image visibility which is a function of tx and ty. The power function family is a simple one-parameter formulation that allows us to test both linear ($k = 1$) and non-linear ($k \neq 1$) softening: higher k provides flatter plateaus in high visibility regime (see Figure 2 right). As seen in Label Smoothing, p can be interpreted as ground

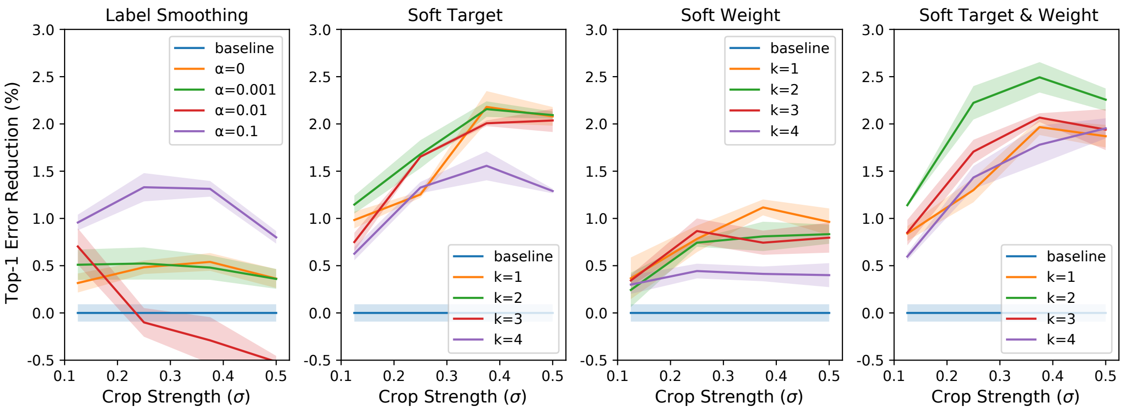


Figure 3. Soft Augmentation reduces the top-1 validation error of ResNet-18 on Cifar-100 by up to 2.5% via combining both target and weight softening (Equation 6). Applying target softening alone (Equation 4) can boost performance by $\sim 2\%$. Crop parameters tx, ty are independently drawn from $N(0, \sigma L)$ ($L = 32$). Higher error reductions indicate better performance over baseline. All results are the means and standard errors across 3 independent runs.

truth class probability of the one-hot learning target. p_{min} is the chance probability depending on the task prior. For example, for Cifar-100, $p_{min} = \frac{1}{\#classes} = 0.01$.

Equation 7 has three assumptions: 1) the information loss is a function of image visibility and all information is lost only when the image is fully occluded, 2) the original label of a training image has a confidence of 100%, which suggests that there is no uncertainty in the class of the label, and 3) the information loss of all images can be approximated by a single confidence-visibility curve. While the first assumption is supported by observations of human visual classification research [44], and empirical evidence in Sections 4.2 and 4.3 suggests that the second and the third assumptions approximately hold, the limitations to these assumptions will be discussed in Section 5.

4.2. How to Soften Targets

As prescribed by Equations 4, 5, and 6, three versions of Soft Augmentation are compared with Label Smoothing across a range of crop strength σ. The popular ResNet-18 models [16] are trained on the 100-class classification Cifar-100 training set. Top-1 error reductions on the validation set are reported (details in Appendix B). Consistent with prior studies, label smoothing can boost model performance by $\sim 1.3\%$ when the smoothing factor α is properly tuned (Figure 3 left).

Combining both target and weight softening (Equation 6) with $k = 2$ and $\sigma \approx 0.3$ boosts model performance by 2.5% (Figure 3 right). Note that $k = 2$ qualitatively resembles the shape of the curve of human visual confidence degradation under occlusion reported by Tang *et al.* [44]. Interestingly, the optimal $\sigma \approx 0.3$ fits the intuitive 3-σ rule. In the next section we freeze $k = 2$ and $\sigma = 0.3$ and show robust improvements that generalize to Cifar-10 [23], ImageNet-1K [9], and ImageNet-V2 tasks [39].

4.3. Supervised Classification

4.3.1 Comparison with Related Methods

As mentioned in Section 2, many approaches similar to soft augmentation have demonstrated empirical performance gains, including additional data augmentation transforms [10], learning augmentation policies [8], softening learning targets [43], and modifying loss functions [29]. However, as training recipes continued to evolve over the past decade, baseline model performance has improved accordingly. As seen in Table 1 (Baseline), our baseline ResNet-18 models with a 500-epoch schedule and cosine learning rate decay [33] not only outperform many recent baseline models of the same architecture, but also beat various published results of Mixup and Cutout. To ensure fair comparisons, we reproduce 6 popular methods: Mixup, Cutout, Label Smoothing, Online Label Smoothing, Focal Loss, and RandAugment, and report the Top-1 Error on Cifar-100 in Table 1. Additional comparisons with the self-reported results are available in Appendix Table 5.

Table 1. Soft augmentation outperforms related methods. Optimal hyperparameters for Mixup [53], Cutout [10], and Online Label Smoothing [52] were applied. α of Focal Loss is tuned as [29] did not prescribe an optimal α for Cifar classification. It is worth noting that our baseline model (20.80%) not only outperforms other published baseline models by 1.5% to 4.8%, but also beat various implementations of Mixup and Cutout. Top-1 errors of ResNet-18 on Cifar-100 are reported.

ResNet-18		Top-1 Error
Baseline	Zhang *et al.* [53]	25.6
	DeVries and Taylor [10]	$22.46_{\pm 0.31}$
	Kim *et al.* [20]	23.59
	Ours	$\mathbf{20.80}_{\pm \mathbf{0.11}}$
Mixup	Zhang *et al.* [53]	21.1
	Kim *et al.* [20]	22.43
	Ours	$\mathbf{19.88}_{\pm \mathbf{0.38}}$
Cutout	DeVries and Taylor [10]	$21.96_{\pm 0.24}$
	Ours	$\mathbf{20.51}_{\pm \mathbf{0.02}}$
Label Smoothing	Ours	$19.47_{\pm 0.18}$
Online Label Smoothing		$20.12_{\pm 0.05}$
Focal Loss ($\alpha = 1$)		$20.45_{\pm 0.08}$
Focal Loss ($\alpha = 2$)		$20.38_{\pm 0.08}$
Focal Loss ($\alpha = 5$)		$20.69_{\pm 0.17}$
RandAugment		$20.99_{\pm 0.11}$
Soft Augmentation		$\mathbf{18.31}_{\pm \mathbf{0.17}}$

Table 1 shows that Soft Augmentation outperforms all other single methods. It is worth noting that although focal loss [29] proposed for detection tasks, it can be tuned to slightly improve classification model performance.

4.3.2 Soft Augmentation Compliments RandAugment

This section investigates the robustness of Soft Augmentation across models and tasks, and how well it compares or complements more sophisticated augmentation policies such as RandAugment [8]. The ImageNet-1K dataset [9] has larger and variable-sized images compared to the Cifar [23] datasets. In contrast with the fixed-sized crop augmentation for Cifar, a crop-and-resize augmentation $t_{(tx,ty,w,h)}(x)$ with random location tx, ty and random size w, h is standard for ImageNet training recipes [7,8,42]. The resizing step is necessary to produce fixed-sized training images (e.g. 224×224). We follow the same $\sigma = 0.3$ principle for drawing tx, ty and w, h (details in Appendix B).

Comparing single methods, Soft Augmentation with *crop only* consistently outperforms the more sophisticated RandAugment with 14 transforms (Table 2). The small ResNet-18 models trained with SA on Cifar-10/100 even outperforms much larger baseline ResNet-50 [39] and WideResNet-28 [50] models.

Table 2. Soft Augmentation (SA) with a **fixed** softening curve of $k = 2$ **doubles** the top-1 error reduction of RandAugment (RA) across datasets and models. Note that the ResNet-18 models trained with SA on Cifar-10/100 even outperform larger baseline ResNet-50 and WideResNet-28 models. All results are mean ± standard error of top-1 validation error in percentage. Best results are shown in bold, runners-up are underlined, and results in parentheses indicate improvement over baseline. Statistics are computed from three runs.

Dataset	Model	Baseline	SA	RA	SA+RA
Cifar100	EfficientNet-B0	$49.70_{\pm 1.55}$	$\underline{42.13_{\pm 0.45}(-7.57)}$	$46.68_{\pm 1.52}(-3.02)$	$\mathbf{38.72_{\pm 0.71}(-10.98)}$
	ResNet-18	$20.80_{\pm 0.11}$	$\underline{18.31_{\pm 0.17}(-2.49)}$	$20.99_{\pm 0.11}(+0.19)$	$\mathbf{18.10_{\pm 0.20}(-2.70)}$
	ResNet-50	$20.18_{\pm 0.30}$	$\underline{18.06_{\pm 0.24}(-2.12)}$	$18.57_{\pm 0.09}(-1.61)$	$\mathbf{16.72_{\pm 0.06}(-3.46)}$
	WideResNet-28	$18.60_{\pm 0.19}$	$\underline{16.47_{\pm 0.18}(-2.13)}$	$17.65_{\pm 0.14}(-0.95)$	$\mathbf{15.37_{\pm 0.17}(-3.23)}$
	PyramidNet + ShakeDrop	$15.77_{\pm 0.17}$	$14.03_{\pm 0.05}(-1.75)$	$\underline{14.02_{\pm 0.28}(-1.76)}$	$\mathbf{12.78_{\pm 0.16}(-2.99)}$
Cifar10	EfficientNet-B0	$17.73_{\pm 0.69}$	$\underline{12.21_{\pm 0.22}(-5.52)}$	$14.54_{\pm 0.47}(-3.19)$	$\mathbf{11.67_{\pm 0.26}(-6.06)}$
	ResNet-18	$4.38_{\pm 0.05}$	$\underline{3.51_{\pm 0.08}(-0.87)}$	$3.89_{\pm 0.06}(-0.49)$	$\mathbf{3.27_{\pm 0.08}(-1.11)}$
	ResNet-50	$4.34_{\pm 0.14}$	$\underline{3.67_{\pm 0.08}(-0.67)}$	$3.91_{\pm 0.14}(-0.43)$	$\mathbf{3.01_{\pm 0.02}(-1.33)}$
	WideResNet-28	$3.67_{\pm 0.08}$	$\underline{2.85_{\pm 0.02}(-0.82)}$	$3.26_{\pm 0.04}(-0.41)$	$\mathbf{2.45_{\pm 0.03}(-1.20)}$
	PyramidNet + ShakeDrop	$2.86_{\pm 0.03}$	$\underline{2.26_{\pm 0.02}(-0.60)}$	$2.32_{\pm 0.08}(-0.54)$	$\mathbf{2.02_{\pm 0.01}(-0.84)}$
ImageNet-1K	ResNet-50	$22.62_{\pm <0.01}$	$\underline{21.66_{\pm 0.02}(-0.96)}$	$22.02_{\pm 0.02}(-0.60)$	$\mathbf{21.27_{\pm 0.05}(-1.35)}$
	ResNet-101	$20.91_{\pm 0.04}$	$20.63_{\pm 0.03}(-0.28)$	$\underline{20.39_{\pm 0.07}(-0.52)}$	$\mathbf{19.86_{\pm 0.03}(-1.05)}$
ImageNet-V2	ResNet-50	$34.97_{\pm 0.03}$	$\underline{33.32_{\pm 0.10}(-1.65)}$	$34.16_{\pm 0.21}(-0.81)$	$\mathbf{32.38_{\pm 0.16}(-2.59)}$
	ResNet-101	$32.68_{\pm 0.04}$	$\underline{31.81_{\pm 0.16}(-0.87)}$	$32.08_{\pm 0.19}(-0.60)$	$\mathbf{31.26_{\pm 0.12}(-1.42)}$

Because RandAugment is a searched policy that is originally prescribed to be applied in addition to the standard crop augmentation [8], one can easily replace the standard crop with soft crop and combine Soft Augmentation and RandAugment. As shown in Table 2, Soft Augmentation complements RandAugment by doubling its top-1 error reduction across tasks and models.

Note that for the small ResNet-18 model trained on Cifar-100, the fixed RandAugment method slightly degrades its performance. Consistent with observations from Cubuk *et al.* [8], the optimal hyperparameters for RandAugment depend on the combination of model capacity and task complexity. Despite the loss of performance of applying RandAugment alone, adding Soft Augmentation reverses the effect and boosts performance by 2.7%.

For the preceding experiments, a fixed $k = 2$ is used for Soft Augmentation and the official PyTorch RandomAugment [38] is implemented to ensure a fair comparison and to evaluate robustness. It is possible to fine-tune the hyperparameters for each model and task to achieve better empirical performance.

4.3.3 Occlusion Robustness

As discussed in Section 2, occlusion robustness in both human vision [34,44,54] and computer vision [22,46,47] have been an important property for real world applications of vision models as objects. To assess the effect of soft augmentation on occlusion robustness of computer vision models, ResNet-50 models are tested with occluded ImageNet validation images (Figure 4 and Appendix Figure 7). 224×224 validation images of ImageNet are occluded with randomly placed square patches that cover λ of the image area. λ is set to $\{0\%, 20\%, 40\%, 60\%, 80\%\}$ to create a range of occlusion levels.

As shown in Figure 5, both RandAugment (RA) and Soft Augmentation (SA) improve occlusion robustness independently across occlusion levels. Combining RA with SA reduces Top-1 error by up to 17%. At 80% occlusion level, SA+RA achieves more than 4× **accuracy improvement** over the baseline (18.98% vs 3.42%).

4.3.4 Confidence Calibration

In addition to top-1 errors, reliability is yet another important aspect of model performance. It measures how close a model's predicted probability (confidence) tracks the true correctness likelihood (accuracy). Expected Calibration Error (ECE) is a popular metric [14, 25, 35] to measure confidence calibration by dividing model predictions into M confidence bins (B_m) and compute a weighted average error between accuracy and confidence:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|, \tag{8}$$

where n is the number of samples, $acc(B_m)$ denotes the accuracy of bin m, and $conf(B_m)$ denotes mean model confidence of bin m. Consistent with Guo *et al.* [14], we set $M = 10$ and compute ECE for Cifar-10 and Cifar-100 tasks.

As shown in Table 3, many methods [25,30,35,45] have been proposed to improve confidence calibration, sometimes at the cost of drastically increased computational overhead [25], or degraded raw performance [30]. We show in Table 3 (and Appendix Table 7) that it is possible to further reduce model top-1 error and expected calibration error simultaneously.

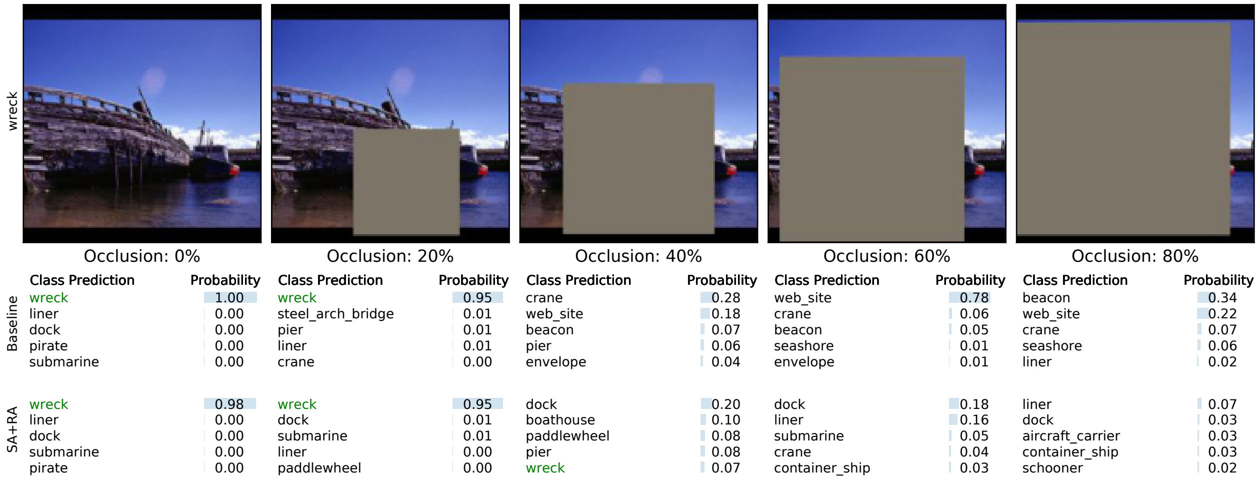


Figure 4. Examples of occluded ImageNet validation images and model predictions of ResNet-50. 224 × 224 validation images of ImageNet are occluded with randomly placed square patches that cover λ of the image area. λ is set to $\{0\%, 20\%, 40\%, 60\%, 80\%\}$ to create a range of occlusion levels.

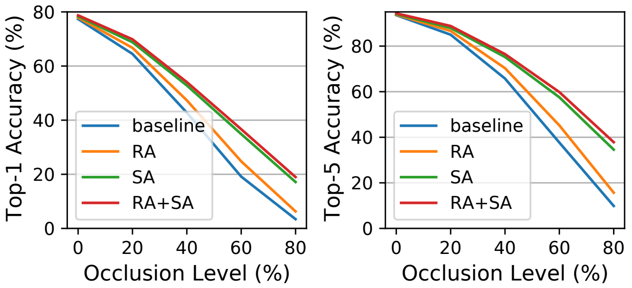


Figure 5. Soft Augmentation improves occlusion robustness of ResNet-50 on ImageNet. Both RandAugment (RA) and Soft Augmentation (SA) improve occlusion robustness independently. Combining RA with SA reduces Top-1 error by up to 17%. At 80% occlusion level, compared with baseline accuracy (3.42%), SA+RA achieves more than **4× accuracy** (18.98%).

Table 3. Soft Augmentation improves both accuracy and calibration. We report mean and standard error of three WideResNet-28 runs per configuration (bottom two rows). On the more challenging Cifar-100 benchmark, our Baseline already outperforms much of prior work in terms of Top-1 error, but has worse calibration error (ECE). Applying Soft Augment + Trivial Augment (SA+TA) reduces Top-1 error by 4.36% and reduces ECE by **more than half**, outperforming even compute-heavy models such as the 5-Ensemble [25]. Similar trends hold for Cifar-10.

Method	Cifar-100		Cifar-10	
	Top-1 Error	ECE	Top-1 Error	ECE
Energy-based [31]	19.74	4.62	4.02	0.85
DUQ [45]	–	–	5.40	1.55
SNGP [30]	20.00	4.33	3.96	1.80
DDU [35]	19.02	4.10	4.03	0.85
5-Ensemble [25]	17.21	3.32	3.41	0.76
Our Baseline	$18.60_{\pm 0.16}$	$4.86_{\pm 0.10}$	$3.67_{\pm 0.07}$	$2.22_{\pm 0.03}$
SA+TA	$\mathbf{14.24}_{\pm \mathbf{0.11}}$	$\mathbf{1.76}_{\pm \mathbf{0.15}}$	$\mathbf{2.23}_{\pm \mathbf{0.06}}$	$\mathbf{0.61}_{\pm \mathbf{0.10}}$

Compared to previous single-model methods, our strong baseline WideResNet-28 models achieves lower top-1 error at the cost of higher ECE. Combining Soft Augmentation with more recently developed augmentation policies such as TrivialAugment [37] (SA+TA) reduces top-1 error by 4.36% and reduces ECE by more than half on Cifar-100, outperforming the 4× more computationally expensive 5-ensemble model [25]. To the best of our knowledge, this is state of the art ECE performance for WideResNet-28 on Cifar without post-hoc calibration.

4.4. Soft Augmentation Boosts Self-Supervised Learning

In contrast with supervised classification tasks where the learning target y_i is usually a one-hot vector, many self-supervised methods such as SimSiam [6] and Barlow Twins [51] learn visual feature representations without class labels by encouraging augmentation invariant feature representations. This section investigates whether Soft Augmentation generalizes to learning settings where no one-hot style labeling is provided.

In a typical setting, two random crops of the same image are fed into a pair of identical twin networks (e.g., ResNet-18) with shared weights and architecture. The learning target can be the maximization of similarity between the feature representations of the two crops [6], or minimization of redundancy [51]. By default, all randomly cropped pairs have equal weights. We propose and test two alternative hypotheses for weight softening with SimSiam. To accommodate self-supervised learning, Equation 7 is modified by replacing visibility $v_{tx,ty}$ with intersection over union IoU of two crops of an image:

$$p = 1 - \alpha(\phi_1, \phi_2, k) = 1 - (1 - p_{min})(1 - IoU_{\phi_1,\phi_2})^k, \quad \textbf{SA\#1} \quad (9)$$

where $\phi_1 = (tx_1, ty_1, w_1, h_1)$ and $\phi_2 = (tx_2, ty_2, w_2, h_2)$ are crop parameters for the first and second sample in a pair.

Table 4. Soft Augmentation (SA#1) improves self supervised learning with SimSiam (ResNet-18) on Cifar-100 by down-weighting sample pairs with *small* intersection over union (IoU), outperforming the opposite hypothesis (SA#2) of down-weighting pairs with *large* IoU. For each configuration, we report means and standard errors of 3 runs with best learning rates (LR) found for Cifar-100. The effect of SA#1 (with a fixed $k = 4$) generalizes to Cifar-10 without re-tuning.

Task	LR	Baseline	LR	SA#1	LR	SA#2
Cifar100	0.2	$37.64_{\pm 0.06}$	0.2	$\mathbf{36.61}_{\pm 0.05}$	0.1	$37.39_{\pm 0.06}$
Cifar10	0.2	$9.87_{\pm 0.03}$	0.2	$\mathbf{9.31_{\pm 0.01}}$	-	-

p is used to soften weights only as no one-hot classification vector is available in this learning setting. With this hypothesis (SA#1), “hard" sample pairs with low IoUs are assigned low weights. Alternatively, one can assign lower weights to “easy" sample pairs with higher IoUs (SA#2), as prescribed by Equation 10:

$$p = 1 - \alpha(\phi_1, \phi_2, k) = 1 - (1 - p_{min})(IoU_{\phi_1,\phi_2})^k. \quad \textbf{SA\#2} \quad (10)$$

We first test all three hypotheses (baseline, SA#1, and SA#2) on Cifar-100 with the SimSiam-ResNet-18 models. Table 4 (top) shows that SA#1 outperform both baseline and SA#2 (details in Appendix B.4). Additional experiments show that models trained with the same SA#1 configuration also generalize to Cifar-10 (Table 4 bottom).

5. Discussion

Other augmentations. While we focus on crop augmentations as an illustrative example, Soft Augmentation can be easily extended to a larger repertoire of transforms such as affine transforms and photometric distortions, as seen in the more sophisticated augmentation policies such as RandAugment. As the formulation of Equation 7 (and Figure 2 right) is directly inspired by the qualitative shape of human vision experiments from Tang *et al*. [44], optimal softening curves for other transforms may be discovered by similar human experiments. However, results with a second transform in Appendix Table 6 suggest that Equation 7 generalizes to additive noise augmentation as well. A potential challenge is determining the optimal softening strategy when a combination of several transforms are applied to an image since the cost of a naive grid search increases exponentially with the number of hyperparameters. Perhaps reinforcement learning methods as seen in RandAugment can be used to speed up the search.

Other tasks. While we limit the scope of Soft Augmentation to image classification as it is directly inspired by human visual classification research, the idea can be generalized to other types of tasks such as natural language modeling and object detection. Recent studies have shown that detection models benefit from soft learning targets in the final stages [3,27], Soft Augment has the potential to complement these methods by modeling information loss of image transform in the models' input stage.

Class-dependant augmentations. As pointed out by Balestriero *et al*. [2], the effects of data augmentation are class-dependent. Thus assumption 3 of Equation 7 does not exactly hold. One can loosen it by adaptively determining the range of transform and softening curve on a per class or per sample basis. As shown in Equation 11,

$$(x_i, y_i) \Rightarrow \left(t_{\phi \sim S(x_i,y_i)}(x_i), g_{\alpha(\phi,x_i,y_i)}(y_i)\right), \qquad (11)$$

two adaptive improvements can be made: 1) the transformation range S where ϕ is drawn from can be made a function of sample (x_i, y_i), 2) the softening factor α can also adapt to (x_i, y_i). Intuitively, the formulation recognizes the heterogeneity of training samples of images at two levels. Firstly, the object of interest can occupy different proportions of an image. For instance, a high-resolution training image with a small object located at the center can allow more aggressive crop transforms without losing its class invariance. Secondly, texture and shape may contribute differently depending on the visual class. A heavily occluded tiger may be recognized solely by its distinctive stripes; in contrast, a minimally visible cloak can be mistaken as almost any clothing.

6. Conclusion

In summary, we draw inspiration from human vision research, specifically how human visual classification performance degrades non-linearly as a function of image occlusion. We propose generalizing data augmentation with invariant transforms to Soft Augmentation where the learning target (e.g. one-hot vector and/or sample weight) softens non-linearly as a function of the degree of the transform applied to the sample.

Using cropping transformations as an example, we empirically show that Soft Augmentation offers robust top-1 error reduction across Cifar-10, Cifar-100, ImageNet-1K, and ImageNet-V2. With a fixed softening curve, Soft Augmentation doubles the top-1 accuracy boost of the popular RandAugment method across models and datasets, and improves performance under occlusion by up to $4\times$. Combining Soft Augment with the more recently developed TrivialAugment further improves model accuracy and calibration simultaneously, outperforming even compute-heavy 5-ensemble models. Finally, self-supervised learning experiments demonstrate that Soft Augmentation also generalizes beyond the popular supervised one-hot classification setting.

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017. 3

[2] Randall Balestriero, Leon Bottou, and Yann LeCun. The effects of regularization and data augmentation are class dependent. *arXiv preprint arXiv:2204.03632*, 2022. 8

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 8

[4] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021. 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 7, 14

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 2, 5

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 1, 2, 5, 6, 12

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 5

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 12

[12] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982. 2

[13] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5, 11

[17] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2

[18] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959. 2

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 1

[20] Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. *arXiv preprint arXiv:2102.03065*, 2021. 5, 12

[21] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020. 12

[22] Adam Kortylewski, Ju He, Qing Liu, and Alan L Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8940–8949, 2020. 2, 6

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 5

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[25] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 1, 6, 7

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[27] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020. 8

[28] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5, 12

[30] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020. 6, 7

[31] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 7

[32] Yang Liu, Jeremy Bernstein, Markus Meister, and Yisong Yue. Learning by turning: Neural architecture aware optimisation. In *International Conference on Machine Learning*, pages 6748–6758. PMLR, 2021. 2

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[34] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2, 6

[35] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deep deterministic uncertainty: A simple baseline, 2021. 6, 7

[36] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 2

[37] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 774–782, 2021. 1, 2, 7

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6, 12

[39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 5, 12

[40] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018. 3

[41] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020. 1

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1, 2, 5

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2, 4, 5

[44] Hanlin Tang, Martin Schrimpf, William Lotter, Charlotte Moerman, Ana Paredes, Josue Ortega Caro, Walter Hardesty, David Cox, and Gabriel Kreiman. Recurrent computations for visual pattern completion. *Proceedings of the National Academy of Sciences*, 115(35):8835–8840, 2018. 1, 3, 4, 5, 6, 8

[45] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020. 6, 7

[46] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654, 2020. 2, 6

[47] Dean Wyatte, Tim Curran, and Randall O'Reilly. The limits of feedforward vision: Recurrent processing promotes robust object recognition when objects are degraded. *Journal of Cognitive Neuroscience*, 24(11):2248–2261, 2012. 1, 2, 6

[48] Mingyang Yi, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Zhi-Ming Ma. Reweighting augmented samples by minimizing the maximal expected loss. *arXiv preprint arXiv:2103.08933*, 2021. 3

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5, 11

[51] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 7

[52] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996, 2021. 5, 12

[53] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 5, 12

[54] Hongru Zhu, Peng Tang, Jeongho Park, Soojin Park, and Alan Yuille. Robustness of object recognition under extreme occlusion in humans and computational models. *arXiv preprint arXiv:1905.04598*, 2019. 1, 2, 6

Appendix A. Implementation

```python
import torch

class SoftCropAugmentation:
    def __init__(self, n_class, sigma=0.3, k=2):
        self.chance = 1/n_class
        self.sigma = sigma
        self.k = k

    def draw_offset(self, limit, sigma=0.3, n=100):
        # draw an integer from a (clipped) Gaussian
        for d in range(n):
            x = torch.randn((1))*sigma
            if abs(x) <= limit:
                return int(x)
        return int(0)

    def __call__(self, image, label):
        # typically, dim1 = dim2 = 32 for Cifar
        dim1, dim2 = image.size(1), image.size(2)
        # pad image
        image_padded = torch.zeros((3, dim1 * 3, dim2 * 3))
        image_padded[:, dim1:2*dim1, dim2:2*dim2] = image
        # draw tx, ty
        tx = self.draw_offset(dim1, self.sigma_crop * dim1)
        ty = self.draw_offset(dim2, self.sigma_crop * dim2)
        # crop image
        left, right = tx + dim1, tx + dim1 * 2
        top, bottom = ty + dim2, ty + dim2 * 2
        new_image = image_padded[:, left: right, top: bottom]
        # compute transformed image visibility and confidence
        v = (dim1 - abs(tx)) * (dim2 - abs(ty)) / (dim1 * dim2)
        confidence = 1 - (1 - self.chance) * (1 - v) ** self.k
        return new_image, label, confidence
```

Listing 1. Pytorch implementation of Soft Crop Augmentation for Cifar.

```python
import torch
import torch.nn.functional as F

def soft_target(pred, label, confidence):
    log_prob = F.log_softmax(pred, dim=1)
    n_class = pred.size(1)
    # make soft one-hot target
    one_hot = torch.ones_like(pred) * (1 - confidence) / (n_class - 1)
    one_hot.scatter_(dim=1, index=label, src=confidence)
    # compute weighted KL loss
    kl = confidence * F.kl_div(input=log_prob,
                               target=one_hot,
                               reduction=’none’).sum(-1)
    return kl.mean()
```

Listing 2. Pytorch implementation of Soft Target loss function.

Appendix B. Experiment Details

Appendix B.1. Supervised Cifar-10/100

For Cifar-100 experiments, we train all ResNet-like models with a batch size 128 on a single Nvidia V100 16GB GPU on Amazon Web Services (AWS) and with an intial learning rate 0.1 with cosine learning rate decay over 500 epochs. EfficientNet-B0 is trained with an initial learning rate of 0.025, PyramidNet-272 is trained with 2 GPUs. We use the Conv-BatchNorm-ReLU configuration of ResNet models [16] and WideResNet-28 with a widening factor of 10 [50]. Horizontal flip is used in all experiments as it is considered a lossless transformation in the context of Cifar images. We find decaying crop aggressiveness towards the end of training (e.g., last 20 epochs) by a large factor (e.g., reducing σ by 1000×) marginally improve performance on Cifar-100, but slightly hurts performance on Cifar-10. Accordingly, we only apply σ decay for all Cifar-100 experiments. A single run of ResNet-18, ResNet-50, and WideResNet-28 takes ~ 2.5, ~ 7, ~ 9 GPU hours on Cifar-10/100, respectively.

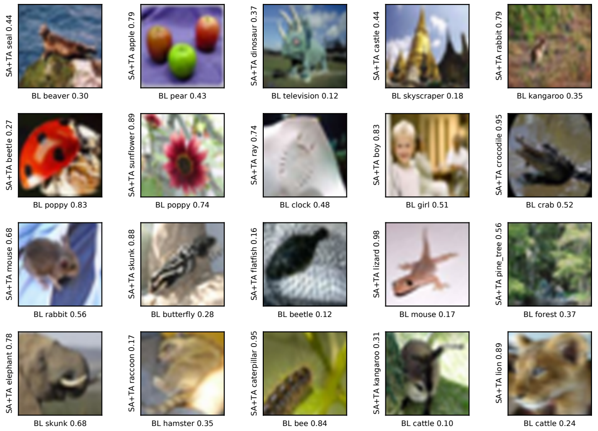


Figure 6. Example images of the Cifar-100 validation set and predictions of WideResNet-28. Predicted classes and confidence levels of models trained with Soft Augmentation + Trivial Augment (SA+TA) and baseline (BL) augmentation are reported. In many cases, SA+TA not only corrects the class prediction, but also improves the model confidence. For instance, BL mistakes “seal” for “beaver” (top-left, both classes belong to the same “aquatic mammal” superclass), and SA+TA makes a correct class prediction with higher confidence.

Appendix B.2. Additional Results

Table 5. Comparing SA with other methods. Recommended hyperparameters for Mixup [53], Cutout [11], and Online Label Smoothing [52]. α of Focal Loss is tuned as Lin *et al.* [29] did not prescribe an optimal α for Cifar classification. Top-1 errors of ResNet-18 on Cifar-100 are reported.

ResNet-18	Top-1 Error
Zhang *et al.* [53]	
Baseline	25.6
Mixup	21.1
Kim *et al.* [21]	
Baseline	23.67
Mixup	23.16
Manifold Mixup	20.98
Puzzle Mix	19.62
Kim *et al.* [20]	
Baseline	23.59
Mixup	22.43
Manifold Mixup	21.64
Puzzle Mix	20.62
Co-Mixup	19.87
Our Baseline	20.80$_{\pm 0.11}$
Label Smoothing	19.47$_{\pm 0.18}$
Online Label Smoothing	20.12$_{\pm 0.05}$
Focal Loss ($\alpha = 1$)	20.45$_{\pm 0.08}$
Focal Loss ($\alpha = 2$)	20.38$_{\pm 0.08}$
Focal Loss ($\alpha = 5$)	20.69$_{\pm 0.17}$
Mixup ($\alpha = 1.0$)	19.88$_{\pm 0.38}$
Cutout ($L = 8$)	20.51$_{\pm 0.02}$
SA	18.31$_{\pm 0.17}$
RA	20.99$_{\pm 0.11}$
SA + RA	**18.10$_{\pm 0.20}$**

Table 6. Soft Augmentation with additive noise improves ResNet-18 performance on Cifar-100. Given an image X and a random noise pattern X_{noise}, and augmented image is given by $X_{aug} = X + \alpha X_{noise}$, where α is drawn from $N(0, 0.1)$ and pixel values of X_{noise} are also independently drawn from $N(0, 0.1)$. Applying Soft Augmentation to additive noise boost performance over baseline as well as Soft Augmentation Crop + RandAugment.

ResNet-18	Top-1 Error
Baseline	20.80±0.11
RA	20.99$_{\pm 0.11}$
Hard Crop	20.26$_{\pm 0.12}$
SA-Crop (k=2)	18.31$_{\pm 0.17}$
Hard Noise	20.68$_{\pm 0.05}$
SA-Noise (k=1)	19.20±0.20
SA-Crop (k=2) + RA	18.10$_{\pm 0.20}$
SA-Noise (k=1) + SA-Crop (k=2) + RA	**17.87±0.17**

Table 7. Soft Augmentation reduces expected calibration error (ECE) of ResNet-50 on ImageNet.

Dataset	Baseline	RA	SA	RA+SA
ImageNet-1K	5.11	4.09	3.17	2.78
ImageNet-V2	9.91	8.84	3.24	3.18

Appendix B.3. ImageNet

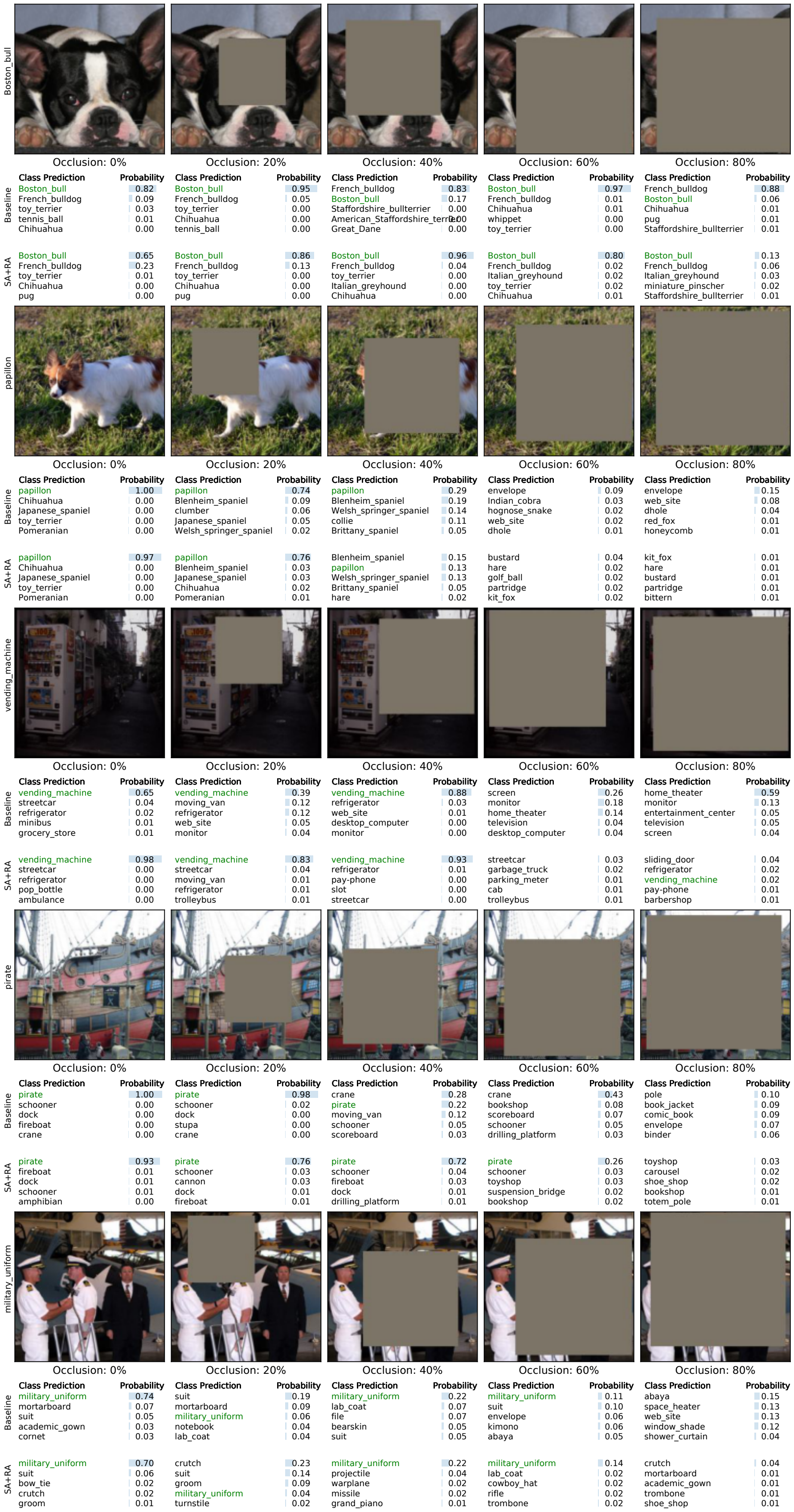
All ImageNet-1k experiments are conducted with a batch size of 256 distributed across 4 Nvidia V100 16GB GPUs on AWS. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset (BSD 3-Clause License) is downloaded from the official website (https://www.image-net.org/). Horizontal flip is used in all experiments as an additional lossless base augmentation. The base learning rate is set to 0.1 with a 5-epoch linear warmup and cosine decay over 270 epochs. A single run of ResNet-50 training takes $\sim 4 \times 4 = 16$ GPU days and ImageNet experiments take a total of 600 GPU days.

We use the official PyTorch [38] implementation of RandAugment and ResNet-50/101 (BSD-style license) and run all experiments with the standard square input $L_{input} = W = H = 224$. Note that the original RandAugment [8] uses a larger input size of $H = 224, W = 244$, but our re-implemention improved top-1 error (22.02 vs 22.4) of ResNet-50 despite using a smaller input size. ImageNet-V2 is a validation set proposed by He *et al.* [39].

For training, the standard crop transform has 4 hyperparameters: $(scale_{min}, scale_{max})$ define the range of the relative size of a cropped image to the original one, $(ratio_{min}, ratio_{max})$ determine lower and upper bound of the aspect ratio of the cropped patch before the final resize step. In practice, a scale is drawn from a uniform distribution $U(scale_{min}, scale_{max})$, then the logarithm of the aspect ratio is drawn from a uniform distribution $U(log(ratio_{min}), log(ratio_{max}))$. Default values are $scale_{min} = 0.08, scale_{max} = 1.0, ratio_{min} = 3/4, ratio_{max} = 4/3$.

Similar to our Cifar crop augmentation, we propose a simplified ImageNet crop augmentation with only 2 hyperparameters $\sigma, L_{min}.$. First, we draw $\Delta w, \Delta h$ from a clipped rectified normal distribution $N^R(0, \sigma(L - L_{min}))$ and get $w = W - \Delta w, h = H - \Delta h$ L_{min} is the minimum resolution of a cropped image and set to half of input resolution 224. tx, ty are then independently drawn from $N(0, \sigma(W + w), N(0, \sigma(H + h)$. Note that we use a fixed set of intuitive values $\sigma = 0.3, L_{min} = 1/2 L_{input} = 112$ for all the experiments.

For model validation, standard augmentation practice first resizes an image so that its short edge has length $L_{input} = 256$, then a center 224×224 crop is applied. Note that L_{input} is an additional hyperparameter introduced by the test augmentation. In contrast, we simplify this by setting L_{input} to the final input size 224 and use this configuration for all ImageNet model evaluation.



Boston_bull

	Occlusion: 0%		Occlusion: 20%		Occlusion: 40%		Occlusion: 60%		Occlusion: 80%	
	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability
Baseline	Boston_bull	0.82	Boston_bull	0.95	French_bulldog	0.83	Boston_bull	0.97	French_bulldog	0.88
	French_bulldog	0.09	French_bulldog	0.05	Boston_bull	0.17	French_bulldog	0.01	Boston_bull	0.06
	toy_terrier	0.03	toy_terrier	0.00	Staffordshire_bullterrier	0.00	Chihuahua	0.01	Chihuahua	0.01
	tennis_ball	0.01	Chihuahua	0.00	American_Staffordshire_terrier	0.00	whippet	0.00	pug	0.01
	Chihuahua	0.00	tennis_ball	0.00	Great_Dane	0.00	toy_terrier	0.00	Staffordshire_bullterrier	0.01
SA+RA	Boston_bull	0.65	Boston_bull	0.86	Boston_bull	0.96	Boston_bull	0.80	Boston_bull	0.13
	French_bulldog	0.23	French_bulldog	0.13	French_bulldog	0.04	French_bulldog	0.02	French_bulldog	0.06
	toy_terrier	0.01	toy_terrier	0.00	toy_terrier	0.00	Italian_greyhound	0.02	Italian_greyhound	0.03
	Chihuahua	0.00	Chihuahua	0.00	Italian_greyhound	0.00	toy_terrier	0.02	miniature_pinscher	0.02
	pug	0.00	pug	0.00	Chihuahua	0.00	Chihuahua	0.01	Staffordshire_bullterrier	0.01

papillon

	Occlusion: 0%		Occlusion: 20%		Occlusion: 40%		Occlusion: 60%		Occlusion: 80%	
	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability
Baseline	papillon	1.00	papillon	0.74	papillon	0.29	envelope	0.09	envelope	0.15
	Chihuahua	0.00	Blenheim_spaniel	0.09	Blenheim_spaniel	0.19	Indian_cobra	0.03	web_site	0.08
	Japanese_spaniel	0.00	clumber	0.06	Welsh_springer_spaniel	0.14	hognose_snake	0.02	dhole	0.04
	toy_terrier	0.00	Japanese_spaniel	0.05	collie	0.11	web_site	0.02	red_fox	0.01
	Pomeranian	0.00	Welsh_springer_spaniel	0.02	Brittany_spaniel	0.05	dhole	0.01	honeycomb	0.01
SA+RA	papillon	0.97	papillon	0.76	Blenheim_spaniel	0.15	bustard	0.04	kit_fox	0.01
	Chihuahua	0.00	Blenheim_spaniel	0.03	papillon	0.13	hare	0.02	hare	0.01
	Japanese_spaniel	0.00	Japanese_spaniel	0.03	Welsh_springer_spaniel	0.13	golf_ball	0.02	bustard	0.01
	toy_terrier	0.00	Chihuahua	0.02	Brittany_spaniel	0.05	partridge	0.02	partridge	0.01
	Pomeranian	0.00	Pomeranian	0.01	hare	0.02	kit_fox	0.02	bittern	0.01

vending_machine

	Occlusion: 0%		Occlusion: 20%		Occlusion: 40%		Occlusion: 60%		Occlusion: 80%	
	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability
Baseline	vending_machine	0.65	vending_machine	0.39	vending_machine	0.88	screen	0.26	home_theater	0.59
	streetcar	0.04	moving_van	0.12	refrigerator	0.03	monitor	0.18	monitor	0.13
	refrigerator	0.02	refrigerator	0.12	web_site	0.01	home_theater	0.14	entertainment_center	0.05
	minibus	0.01	web_site	0.05	desktop_computer	0.00	television	0.04	television	0.05
	grocery_store	0.01	monitor	0.04	monitor	0.00	desktop_computer	0.04	screen	0.04
SA+RA	vending_machine	0.98	vending_machine	0.83	vending_machine	0.93	streetcar	0.03	sliding_door	0.04
	streetcar	0.00	streetcar	0.04	refrigerator	0.01	garbage_truck	0.02	refrigerator	0.02
	refrigerator	0.00	moving_van	0.01	pay-phone	0.00	parking_meter	0.01	vending_machine	0.02
	pop_bottle	0.00	refrigerator	0.01	slot	0.00	cab	0.01	pay-phone	0.01
	ambulance	0.00	trolleybus	0.01	streetcar	0.00	trolleybus	0.01	barbershop	0.01

pirate

	Occlusion: 0%		Occlusion: 20%		Occlusion: 40%		Occlusion: 60%		Occlusion: 80%	
	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability
Baseline	pirate	1.00	pirate	0.98	crane	0.28	crane	0.43	pole	0.10
	schooner	0.00	schooner	0.02	pirate	0.22	bookshop	0.08	book_jacket	0.09
	dock	0.00	dock	0.00	moving_van	0.12	scoreboard	0.07	comic_book	0.09
	fireboat	0.00	stupa	0.00	schooner	0.05	schooner	0.05	envelope	0.07
	crane	0.00	crane	0.00	scoreboard	0.03	drilling_platform	0.03	binder	0.06
SA+RA	pirate	0.93	pirate	0.76	pirate	0.72	pirate	0.26	toyshop	0.03
	fireboat	0.01	schooner	0.03	schooner	0.04	schooner	0.03	carousel	0.02
	dock	0.01	cannon	0.03	fireboat	0.03	toyshop	0.03	shoe_shop	0.02
	schooner	0.01	dock	0.01	dock	0.01	suspension_bridge	0.02	bookshop	0.01
	amphibian	0.00	fireboat	0.01	drilling_platform	0.01	bookshop	0.02	totem_pole	0.01

military_uniform

	Occlusion: 0%		Occlusion: 20%		Occlusion: 40%		Occlusion: 60%		Occlusion: 80%	
	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability	Class Prediction	Probability
Baseline	military_uniform	0.74	suit	0.19	military_uniform	0.22	military_uniform	0.11	abaya	0.15
	mortarboard	0.07	mortarboard	0.09	lab_coat	0.07	suit	0.10	space_heater	0.13
	suit	0.05	military_uniform	0.06	file	0.07	envelope	0.06	web_site	0.13
	academic_gown	0.03	notebook	0.04	bearskin	0.05	kimono	0.06	window_shade	0.12
	cornet	0.03	lab_coat	0.04	suit	0.05	abaya	0.05	shower_curtain	0.04
SA+RA	military_uniform	0.70	crutch	0.23	military_uniform	0.22	military_uniform	0.14	crutch	0.04
	suit	0.06	suit	0.14	projectile	0.04	lab_coat	0.02	mortarboard	0.01
	bow_tie	0.02	groom	0.09	warplane	0.02	cowboy_hat	0.02	academic_gown	0.01
	crutch	0.02	military_uniform	0.04	missile	0.02	rifle	0.02	trombone	0.01
	groom	0.01	turnstile	0.02	grand_piano	0.01	trombone	0.02	shoe_shop	0.01

Figure 7. Examples of occluded ImageNet validation images and model predictions of ResNet-50.

Appendix B.4. Self-Supervised Cifar-10/100

Self-supervised SimSiam experiments are run on a single Nvidia A6000 GPU. We follow the standard two-step training recipe [6]. 1) We first train the Siamese network in a self-supervised manner to learn visual features for 500 epochs with a cosine decay schedule and a batch size of 512. We apply Soft Augmentation only during this step. 2) The linear layer is then tuned with ground-truth labels for 100 epochs with an initial learning rate of 10 and 10× decay at epochs 60 and 80. Following [6], we set $scale_{min} = 0.2, scale_{max} = 1.0, ratio_{min} = 3/4, ratio_{max} = 4/3$. Since down-weighting training samples in a batch effectively reduces learning rate and SimSiam is sensitive to it, we normalized the weight in a batch so that the mean remains 1 and re-tuned the learning rate (Table 8).

Table 8. Soft Augmentation improve self supervised learning with SimSiam. Mean ± standard error of top-1 validation errors of three runs of ResNet-18 are reported.

Task	lr	baseline	SA#1	Δ #1	SA#2	Δ #2
Cifar100	0.1	$39.50_{\pm 0.13}$	$40.21_{\pm 0.03}$	$+0.71$	$37.39_{\pm 0.06}$	-2.11
	0.2	$37.64_{\pm 0.06}$	$\mathbf{36.61}_{\pm 0.05}$	-1.03	$39.20_{\pm 0.42}$	$+1.56$
	0.4	$40.28_{\pm 2.49}$	$37.68_{\pm 0.06}$	-2.60	Diverged	-
	0.5	$43.26_{\pm 3.03}$	$41.94_{\pm 0.04}$	-1.32	Diverged	-
	0.8	$78.88_{\pm 9.05}$	$55.44_{\pm 4.15}$	-23.44	Diverged	-
Cifar10	0.2	$9.87_{\pm 0.03}$	$\mathbf{9.31_{\pm 0.01}}$	-0.56	-	-

Table 9. SimSiam k tuning on Cifar-100 (single run)

learning rate	k	Top-1 Error
0.2	1	37.78
	2	37.27
	3	36.34
	4	**36.31**

Appendix C. Effects of Target Smoothing and Loss Reweighting on Loss Functions

Consider the KL divergence loss of a single learning sample with a one-hot ground truth vector $\boldsymbol{y^{true}}$, and the softmax prediction vector of a model is denoted by $\boldsymbol{y^{pred}}$:

$$\begin{aligned} L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) &= w * D_{KL}(\boldsymbol{y^{true}} || \boldsymbol{y^{pred}}) \\ &= w * \sum_{n=1}^{N} y_n^{true} * log(\frac{y_n^{true}}{y_n^{pred}}), \end{aligned} \tag{12}$$

let $n*$ be the ground truth class of an N-class classification task, Equation 12 can be re-written as:

$$\begin{aligned} L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) &= -w * y_{n*}^{true} * log(y_{n*}^{pred}) \\ &+ w * \left[y_{n*}^{true} * log(y_{n*}^{true}) + \sum_{n \neq n*} y_n^{true} * log(\frac{y_n^{true}}{y_n^{pred}}) \right]. \end{aligned} \tag{13}$$

In the case of hard one-hot ground truth target where $y_{n*}^{true} = 1$ and $y_n^{true} = 0, n \neq n*$, with the default weight $w = 1$ it degenerates to cross entropy loss:

$$L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) = -log(y_{n*}^{pred}), \tag{14}$$

Now we apply label smoothing style softening to the one-hot target y^{true} so that $y_{n*}^{true} = p$ and $y_n^{true} = (1 - p)/(N-1) = q, n \neq n*$:

$$\begin{aligned} L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) &= -p * log(y_{n*}^{pred}) \\ &+ \left[p * log(p) + \sum_{n \neq n*} q * log(\frac{q}{y_n^{pred}}) \right]. \end{aligned} \tag{15}$$

If q is not distributed, and $y_n^{true} = 0, n \neq n*$ (This configuration does not correspond to any of our experiments):

$$L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) = -p * log(y_{n*}^{pred}) + p * log(p), \tag{16}$$

When only weight w is softened to $w = p$:

$$L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) = -p * log(y_{n*}^{pred}). \tag{17}$$

Note that p is not a function of model weights, so when we take the derivative w.r.t. model weights to compute gradient, Equations 16 and 17 yield the same gradient.

When both the one-hot label and weight are softened with p:

$$\begin{aligned} L(\boldsymbol{y^{pred}}, \boldsymbol{y^{true}}) &= -p^2 * log(y_{n*}^{pred}) \\ &+ p * \left[p * log(p) + \sum_{n \neq n*} q * log(\frac{q}{y_n^{pred}}) \right]. \end{aligned} \tag{18}$$

The three types of softening in Section 4 are unique as suggested by Equations 15, 17, and 18.