RETHINKING DATA AUGMENTATION: SELF-SUPERVISION AND SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data augmentation techniques, e.g., flipping or cropping, which systematically enlarge the training dataset by explicitly generating more training samples, are effective in improving the generalization performance of deep neural networks. In the supervised setting, a common practice for data augmentation is to assign the same label to all augmented samples of the same source. However, if the augmentation results in large distributional discrepancy among them (e.g., rotations), forcing their label invariance may be too difficult to solve and often hurts the performance. To tackle this challenge, we suggest a simple yet effective idea of learning the joint distribution of the original and self-supervised labels of augmented samples. The joint learning framework is easier to train, and enables an aggregated inference combining the predictions from different augmented samples for improving the performance. Further, to speed up the aggregation process, we also propose a knowledge transfer technique, self-distillation, which transfers the knowledge of augmentation into the model itself. We demonstrate the effectiveness of our data augmentation framework on various fully-supervised settings including the few-shot and imbalanced classification scenarios.

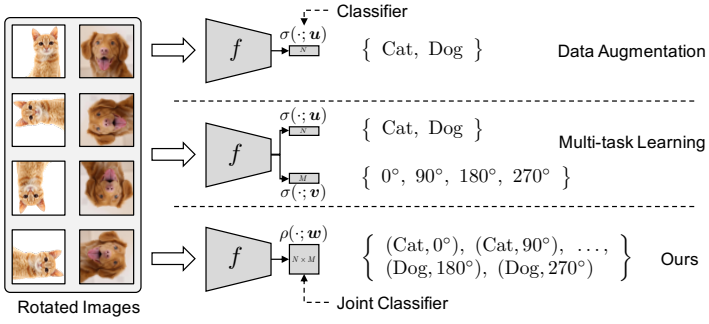
1 INTRODUCTION

Training deep neural networks (DNNs) generally requires a large number of training samples. When the number of training samples is small, DNNs become susceptible to overfitting, causing high generalization errors on the test samples. This overfitting problem is at the center of DNN research, where many regularization techniques have been investigated in the literature (Srivastava et al., 2014; Huang et al., 2016; Gastaldi, 2017). The most explicit and easy-to-use regularization technique is *data augmentation* (Zhong et al., 2017; DeVries & Taylor, 2017; Zhang et al., 2018; Cubuk et al., 2019), which aims to increase the volume of the training set by altering the existing training data.

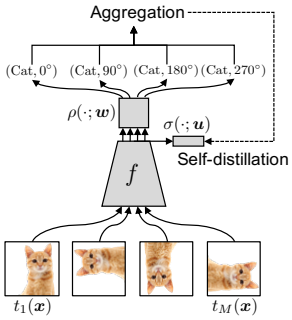
In supervised learning scenarios, data augmentation is done simply by augmenting each sample with multiple transformations that do not affect their semantics. Consequently, data augmentation during training forces DNNs to be invariant to the augmentation transformations. However, depending on the type of transformations, learning transformation-invariant properties may be difficult or compete with the original task, and thus could hurt the performance. For example, for certain fine-grained image classification tasks (e.g., species of birds), the color information could be crucial in class discrimination. In such a case, the data augmentation using color transformations should be avoided.

The evidence that learning invariance is not always helpful could be found in many recent works on *self-supervised learning* (Gidaris et al., 2018; Zhang et al., 2019; Doersch et al., 2015; Noroozi & Favaro, 2016), where the model is trained with artificial labels constructed by input transformations, e.g., the rotation degree for rotated images, without any human-annotated labels. These works have shown that it is possible to learn high-level representations just by learning to predict such transformations. This suggests that some meaningful information could be lost when trying to learn a transformation-invariant property under conventional data augmentation.

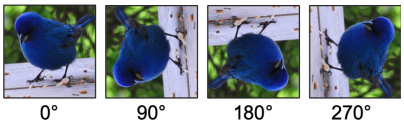
While self-supervision was originally developed for unsupervised learning, there are many recent attempts to use it for other related purposes, e.g., semi-supervised learning (Zhai et al., 2019), robustness (Hendrycks et al., 2019) and adversarial generative networks (Chen et al., 2019). They commonly maintain two separated classifiers (yet sharing common feature representations) for the original and self-supervised tasks. However, such a multi-task learning strategy also forces invari-



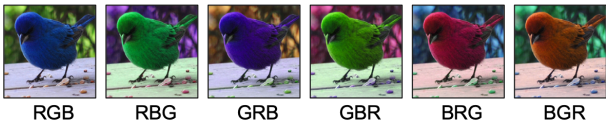
(a) Difference with previous approaches.



(b) Aggregation & self-distillation.



(c) Rotation ($M = 4$).



(d) Color permutation ($M = 6$).

Figure 1: (a) An overview of our self-supervised data augmentation and previous approaches with self-supervision. (b) Illustrations of our aggregation method utilizing all augmented samples and self-distillation method transferring the aggregated knowledge into itself. (c) Rotation-based augmentation. (d) Color-permutation-based augmentation.

ance with respect to self-supervision for the primary classifier performing the original task. Thus, utilizing the self-supervised labels in this way could hurt the performance, e.g., in fully supervised settings. This inspires us to revisit and explore data augmentation methods with self-supervision.

Contribution. We primarily focus on fully supervised learning setups, i.e., assume not only the original/primary labels for all training samples, but also self-supervised labels for the augmented samples. Our main idea is simple and intuitive (see Figure 1(a)): maintain a single joint classifier, instead of two separate classifiers typically used in the prior self-supervision literature. For example, if the original and self-supervised tasks are CIFAR10 (10 labels) and rotation (4 labels), respectively, we learn the joint probability distribution on all possible combinations of 40 labels. This approach assumes no relationship between the original and self-supervised labels, and consequently does not force any invariance to the transformations. Furthermore, since we assign different self-supervised labels for each transformation, it is possible to make a prediction by aggregating across all transformations at test time, as illustrated in Figure 1(b). This can provide an (implicit) ensemble effect using a single model. Finally, to speed up the evaluation process, we also propose a novel knowledge transfer technique, self-distillation, which transfers the knowledge of the aggregated prediction into the model itself.

In our experiments, we consider two types of transformations for self-supervised data augmentation, *rotation* (4 transformations) and *color permutation* (6 transformations), as illustrated in Figure 1(c) and Figure 1(d), respectively. We also consider composed transformations using both rotation and color permutation, i.e., up to 24 transformations. To show wide applicability of our method, we consider various image benchmark datasets and classification scenarios including the few-shot and imbalanced classification tasks. In all tested settings, our simple method improves the classification accuracy significantly and consistently. As desired, the gain tends to be larger when using more augmentation (or self-supervised) labels. We highlight some of our experimental results as follows:

- We show that the proposed self-supervised data augmentation methods with aggregation (SDA+AG) and self-distillation (SDA+SD) can provide significant improvements in the standard fully supervised settings. For example, by using 4 self-supervised labels of rotation, SDA+AG (or SDA+SD) achieves 8.60% (or 5.24%) and 18.8% (or 15.3%) relative gains on the CIFAR100 and CUB200 datasets, respectively, compared to the baseline.[1] The gain is increased up to 20.8% in CUB200 by using 12 composed transformations of the rotation and color permutation.

[1] In this case, SDA+SD is 4 times faster than SDA+AG, as the latter aggregates predictions of 4 rotations.

- We show that SDA+AG can improve the state-of-the-art methods, ProtoNet (Snell et al., 2017) and MetaOptNet (Lee et al., 2019), for few-shot classification, e.g., MetaOptNet with SDA+AG achieves 7.05% relative gain on 5-way 5-shot tasks on the FC100 dataset.
- We show that SDA+SD can improve the state-of-the-art methods, Class-Balanced (Cui et al., 2019) and LDAM (Cao et al., 2019), for imbalanced classification, e.g., LDAM with SDA+SD achieves 8.30% relative gain on an imbalanced version of the CIFAR100 dataset.

We remark that rotation and color permutation are rarely used in the literature for improving fully supervised learning. We think our results are useful to guide many interesting future directions.

2 Self-supervised data augmentation

In this section, we provide the details of our self-supervised data augmentation techniques. We first discuss conventional techniques on data augmentation and self-supervision with their limitations in Section 2.1. Then, we propose our learning framework for data augmentation that can fully utilize the power of self-supervision. In Section 2.2, we also introduce a *self-distillation* technique transferring the augmented knowledge into the model itself for accelerating the inference speed.

Notation. Let $\boldsymbol{x} \in \mathbb{R}^d$ be an input, $y \in \{1, \dots, N\}$ be its label where N is the number of classes, $\mathcal{L}_{\text{CE}}$ be the cross-entropy loss function, $\sigma(\cdot; \boldsymbol{u})$ be the softmax classifier, i.e., $\sigma_i(\boldsymbol{z}; \boldsymbol{u}) = \exp(\boldsymbol{u}_i^\top \boldsymbol{z}) / \sum_k \exp(\boldsymbol{u}_k^\top \boldsymbol{z})$, and $\boldsymbol{z} = f(\boldsymbol{x}; \boldsymbol{\theta})$ be an embedding vector of $\boldsymbol{x}$ where f is a neural network with the parameter $\boldsymbol{\theta}$. We also let $\tilde{\boldsymbol{x}} = t(\boldsymbol{x})$ denote an augmented sample using a transformation t, and $\tilde{\boldsymbol{z}} = f(\tilde{\boldsymbol{x}}; \boldsymbol{\theta})$ be the embedding of the augmented sample.

2.1 Data augmentation and self-supervision

Data augmentation. In a supervised setting, the conventional data augmentation aims to improve upon the generalization ability of the target neural network f by leveraging certain transformations that can preserve their semantics, e.g. cropping, contrast enhancement, and flipping. We can write the training objective $\mathcal{L}_{\text{DA}}$ with data augmentation as follows:

$$\mathcal{L}_{\text{DA}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{u}) = \mathbb{E}_{t \sim T}\Big[\mathcal{L}_{\text{CE}}(\sigma(f(\tilde{\boldsymbol{x}}; \boldsymbol{\theta}); \boldsymbol{u}), y)\Big] \tag{1}$$

where T is a distribution of the transformations for data augmentation. Optimizing the above loss forces the classifier $\sigma(f(\cdot; \boldsymbol{\theta}); \boldsymbol{u})$ to be invariant to the transformations. However, depending on the type of transformations, forcing such invariance may not make sense, as the statistical characteristics of the augmented training samples could become very different from those of original training samples (e.g., rotation). In such a case, enforcing invariance to those transformations would make the learning more difficult, and even degrade the performance (see Table 1 in Section 3.2).

Self-supervision. The recent self-supervised learning literature (Zhang et al., 2019; Doersch et al., 2015; Noroozi & Favaro, 2016; Larsson et al., 2017; Oord et al., 2018; Gidaris et al., 2018) has shown that high-level semantic representations can be learned by predicting labels that could be obtained from the input signals without any human annotations. In self-supervised learning, models learn to predict which transformation t is applied to an input $\boldsymbol{x}$ given a modified sample $\tilde{\boldsymbol{x}} = t(\boldsymbol{x})$. The common approach to utilize self-supervised labels is to optimize two losses of the original and self-supervised tasks, while sharing the feature space among them; that is, the two tasks are trained in a multi-task learning framework (Hendrycks et al., 2019; Zhai et al., 2019; Chen et al., 2019). Thus, in a fully supervised setting, one can formulate the multi-task objective $\mathcal{L}_{\text{MT}}$ with self-supervision as follows:

$$\mathcal{L}_{\text{MT}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{u}, \boldsymbol{v}) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}); \boldsymbol{u}), y) + \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}); \boldsymbol{v}), j) \tag{2}$$

where $\{t_j\}_{j=1}^M$ is a set of pre-defined transformations, M is the number of self-supervised labels, $\sigma(\cdot; \boldsymbol{v})$ is the classifier for self-supervision, and $\tilde{\boldsymbol{x}}_j = t_j(\boldsymbol{x})$. The above loss also forces the primary classifier $\sigma(f(\cdot; \boldsymbol{\theta}); \boldsymbol{u})$ to be invariant to the transformations $\{t_j\}$. Therefore, due to the aforementioned reason, the usage of additional self-supervised labels does not guarantee the performance improvement, in particular, for fully supervised settings (see Table 1 in Section 3.2).

2.2 ELIMINATING INVARINCE VIA JOINT-LABEL CLASSIFIER

Our key idea is to remove the unnecessary invariant property of the classifier $\sigma(f(\cdot;\boldsymbol{\theta});\boldsymbol{u})$ in (1) and (2) among the transformed samples. To this end, we use a joint softmax classifier $\rho(\cdot;\boldsymbol{w})$ which represents the joint probability as $P(i,j|\tilde{\boldsymbol{x}}) = \rho_{ij}(\tilde{\boldsymbol{z}};\boldsymbol{w}) = \exp(\boldsymbol{w}_{ij}^\top\tilde{\boldsymbol{z}})/\sum_{k,l}\exp(\boldsymbol{w}_{kl}^\top\tilde{\boldsymbol{z}})$. Then, our training objective can be written as

$$\mathcal{L}_{\text{SDA}}(\boldsymbol{x},y;\boldsymbol{\theta},\boldsymbol{w}) = \frac{1}{M}\sum_{j=1}^{M}\mathcal{L}_{\text{CE}}(\rho(f(\tilde{\boldsymbol{x}}_j;\boldsymbol{\theta});\boldsymbol{w}),(y,j)) \tag{3}$$

where $\mathcal{L}_{\text{CE}}(\rho(\tilde{\boldsymbol{z}};\boldsymbol{w}),(i,j)) = -\log\rho_{ij}(\tilde{\boldsymbol{z}};\boldsymbol{w})$. Note that the above objective can be reduced to the data augmentation objective (1) when $\boldsymbol{w}_{ij} = \boldsymbol{u}_i$ for all i, and the multi-task learning objective (2) when $\boldsymbol{w}_{ij} = \boldsymbol{u}_i + \boldsymbol{v}_j$ for all i,j. It means that (3) is easier to optimize than (2) since both consider the same set of multi-labels, but latter forces the additional constraint $\boldsymbol{w}_{ij} = \boldsymbol{u}_i + \boldsymbol{v}_j$. The difference between the conventional augmentation, multi-task learning, and ours is illustrated in Figure 1(a). During training, we feed all M augmented samples simultaneously for each iteration as Gidaris et al. (2018) did, i.e., we minimize $\frac{1}{|B|}\sum_{(\boldsymbol{x},y)\in B}\mathcal{L}_{\text{SDA}}(\boldsymbol{x},y;\boldsymbol{\theta},\boldsymbol{w})$ for each mini-batch B. We also assume the first transformation is the identity function, i.e., $\tilde{\boldsymbol{x}}_1 = t_1(\boldsymbol{x}) = \boldsymbol{x}$.

Aggregated inference. Given a test sample $\boldsymbol{x}$ or its augmented sample $\tilde{\boldsymbol{x}}_j = t_j(\boldsymbol{x})$ by a transformation t_j, we do not need to consider all $N\times M$ labels for the prediction of its original label, because we already know which transformation is applied. Therefore, we predict a label using the conditional probability $P(i|\tilde{\boldsymbol{x}}_j,j) = \exp(\boldsymbol{w}_{ij}^\top\tilde{\boldsymbol{z}}_j)/\sum_k\exp(\boldsymbol{w}_{kj}^\top\tilde{\boldsymbol{z}}_j)$ where $\tilde{\boldsymbol{z}}_j = f(\tilde{\boldsymbol{x}}_j;\boldsymbol{\theta})$. Furthermore, for all possible transformations $\{t_j\}$, we aggregate the corresponding conditional probabilities to improve the classification accuracy, i.e., we train a single model, which can perform inference as an ensemble model. To compute the probability of the *aggregated inference*, we first average pre-softmax activations, and then compute the softmax probability as follows:

$$P_{\text{aggregated}}(i|\boldsymbol{x}) = \frac{\exp(s_i)}{\sum_{k=1}^{N}\exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{w}_{ij}^\top\tilde{\boldsymbol{z}}_j.$$

Since we assign different labels for each transformation t_j, our aggregation scheme improves accuracy significantly. Somewhat surprisingly, it achieves comparable performance with the ensemble of multiple independent models in our experiments (see Table 2 in Section 3.2). We refer to the counterpart of the aggregation as *single inference*, which uses only the non-augmented or original sample $\tilde{\boldsymbol{x}}_1 = \boldsymbol{x}$, i.e., predicts a label using $P(i|\tilde{\boldsymbol{x}}_1,1)$.

Self-distillation from aggregation. Although the aforementioned aggregated inference achieves outstanding performance, it requires to compute $\tilde{\boldsymbol{z}}_j = f(\tilde{\boldsymbol{x}}_j;\boldsymbol{\theta})$ for all j, i.e., it requires M times higher computation cost than the single inference. To accelerate the inference, we perform self-distillation (Hinton et al., 2015; Lan et al., 2018) from the aggregated knowledge $P_{\text{aggregated}}(\cdot|\boldsymbol{x})$ to another classifier $\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u})$ parameterized by $\boldsymbol{u}$, as illustrated in Figure 1(b). Then, the classifier $\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u})$ can maintain the aggregated knowledge using only one embedding $\boldsymbol{z} = f(\boldsymbol{x};\boldsymbol{\theta})$. To this end, we optimize the following objective:

$$\begin{aligned}\mathcal{L}_{\text{SDA+SD}}(\boldsymbol{x},y;\boldsymbol{\theta},\boldsymbol{w},\boldsymbol{u}) &= \mathcal{L}_{\text{SDA}}(\boldsymbol{x},y;\boldsymbol{\theta},\boldsymbol{w})\\ &\quad + D_{\text{KL}}(P_{\text{aggregated}}(\cdot|\boldsymbol{x})\|\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u})) + \beta\mathcal{L}_{\text{CE}}(\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u}),y)\end{aligned} \tag{4}$$

where β is a hyperparameter and we simply choose $\beta\in\{0,1\}$. When computing the gradient of $\mathcal{L}_{\text{SDA+SD}}$, we consider $P_{\text{aggregated}}(\cdot|\boldsymbol{x})$ as a constant. After training, we use $\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u})$ for inference without aggregation.

3 EXPERIMENTS

We experimentally validate our self-supervised data augmentation techniques described in Section 2. Throughout this section, we refer to data augmentation (1) as DA, multi-task learning (2) as MT, and ours self-supervised data augmentation (3) as SDA for notational simplicity. After training with SDA, we consider two inference schemes: the single inference and the aggregated inference denoted by SDA+SI and SDA+AG, respectively. We also denote the self-distillation method (4) as SDA+SD which uses only the single inference $\sigma(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{u})$.

3.1 SETUP

Datasets and models. We evaluate our method on various classification datasets: CIFAR10/100 (Krizhevsky et al., 2009), Caltech-UCSD Birds or CUB200 (Wah et al., 2011), Indoor Scene Recognition or MIT67 (Quattoni & Torralba, 2009), Stanford Dogs (Khosla et al., 2011), and tiny-ImageNet[2] for standard or imbalanced image classification; mini-ImageNet (Vinyals et al., 2016), CIFAR-FS (Bertinetto et al., 2019), and FC100 (Oreshkin et al., 2018) for few-shot classification. Note that CUB200, MIT67, and Stanford Dogs are fine-grained datasets. We use residual networks (He et al., 2016) for all experiments: 32-layer ResNet for CIFAR, 18-layer ResNet for three fine-grained datasets and tiny-ImageNet, and ResNet-12 (Lee et al., 2019) for few-shot datasets.

Implementation details. For the standard image classification datasets, we use SGD with learning rate of 0.1, momentum of 0.9, and weight decay of 10^{-4}. We train for 80k iterations with batch size of 128. For the fine-grained datasets, we train for 30k iterations with batch size of 32 because they have a relatively smaller number of training samples. We decay the learning rate by the constant factor of 0.1 at 50% and 75% iterations. We report the average accuracy of 3 trials for all experiments. For few-shot learning and imbalance experiments, we use the publicly available codes of MetaOptNet (Lee et al., 2019) and LDAM (Cao et al., 2019), respectively.

Choices of transformation. Since using the entire input image during training is important in achieving improved classification accuracy, some self-supervision techniques are not suitable for our purpose. For example, the Jigsaw puzzle approach (Noroozi & Favaro, 2016) divides an input image to 3×3 patches, and computes their embedding separately. Prediction using the embedding performs worse than that using the entire image. To avoid this issue, we choose two transformations which use the entire input image without cropping: *rotation* (Gidaris et al., 2018) and *color permutation*. Rotation constructs $M = 4$ rotated images ($0°$, $90°$, $180°$, $270°$) as illustrated in Figure 1(c). This transformation is widely used for self-supervision due to its simplicity, e.g., Chen et al. (2019). Color permutation constructs $M = 3! = 6$ different images via swapping RGB channels as illustrated in Figure 1(d). This transformation can be useful when color information is important such as fine-grained classification datasets.

3.2 ABLATION STUDY

Comparison with DA and MT. We first verify that our proposed method can utilize self-supervision without loss of accuracy on fully supervised datasets while data augmentation and multi-task learning approaches cannot. To this end, we train models on generic classification datasets, CIFAR10/100 and tiny-ImageNet, using three different objectives: data augmentation $\mathcal{L}_{\text{DA}}$ (1), multi-task learning $\mathcal{L}_{\text{MT}}$ (2), and our self-supervised data augmentation $\mathcal{L}_{\text{SDA}}$ (3) with rotation. As reported in Table 1, $\mathcal{L}_{\text{DA}}$ and $\mathcal{L}_{\text{MT}}$ degrade the performance significantly compared to the baseline that does not use the rotation-based augmentation. However, when training with $\mathcal{L}_{\text{SDA}}$, the performance is improved. Figure 2 shows the classification accuracy on training and test samples of CIFAR100 while training. As shown in the figure, $\mathcal{L}_{\text{DA}}$ causes a higher generalization error than others because $\mathcal{L}_{\text{DA}}$ forces the unnecessary invariant property. Moreover, optimizing $\mathcal{L}_{\text{MT}}$ is harder than doing $\mathcal{L}_{\text{SDA}}$ as described in Section 2.2, thus the former achieves the lower accuracy on both training and test samples than the latter. These results show that learning invariance to some transformations, e.g., rotation, makes optimization harder and degrades the performance. Namely, such transformations should be carefully handled.

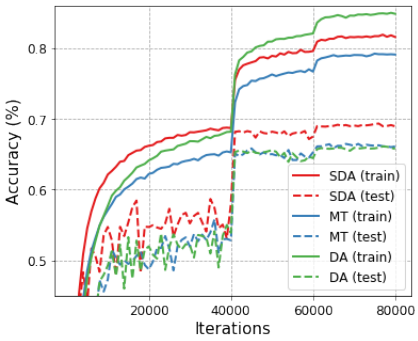


Figure 2: Training/test accuracy of DA, MT, SDA while training on CIFAR100.

Comparison with ten-crop and ensemble. Next, to evaluate the effect of aggregation in SDA-trained models, we compare the aggregation using rotation with other popular aggregation schemes: the ten-crop (Krizhevsky et al., 2012) method which aggregates the prediction scores over a number of cropped images, and the independent ensemble which aggregates the scores over four independently trained models. Note that the ensemble requires $4\times$ more parameters than ours and ten-crop.

[2] https://tiny-imagenet.herokuapp.com/

Table 1: Classification accuracy (%) of single inference using data augmentation (DA), multi-task learning (MT), and ours self-supervised data augmentation (SDA) with rotation. The best accuracy is indicated as bold, and the relative gain over the baseline is shown in brackets.

Dataset	Baseline	DA	MT	SDA+SI
CIFAR10	92.39	90.44 (-2.11%)	90.79 (-1.73%)	**92.50** (+0.12%)
CIFAR100	68.27	65.73 (-3.72%)	66.10 (-3.18%)	**68.68** (+0.60%)
tiny-ImageNet	63.11	60.21 (-4.60%)	58.04 (-8.03%)	**63.99** (+1.39%)

Table 2: Classification accuracy (%) of the ten-crop, indenpendent ensemble, and our aggregation using rotation (SDA+AG). The best accuracy is indicated as bold, and the relative gain over the baseline is shown in brackets.

	Single Model			4 Models	
Dataset	Baseline	ten-crop	SDA+AG	Ensemble	Enemble + SDA+AG
CIFAR10	92.39	93.33	**94.50** (+2.28%)	94.36	**95.10** (+2.93%)
CIFAR100	68.27	70.54	**74.14** (+8.60%)	74.82	**76.40** (+11.9%)
tiny-ImageNet	63.11	64.95	**66.95** (+6.08%)	68.18	**69.01** (+9.35%)

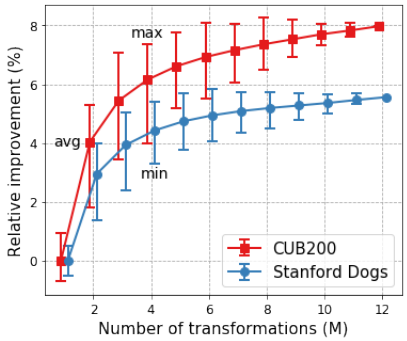


Figure 3: Relative improvements of aggregation versus the number of transformations.

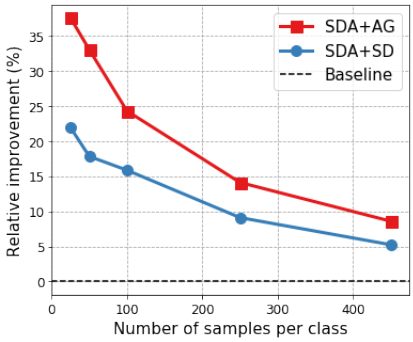


Figure 4: Relative improvements over baselines on subsets of CIFAR100.

Surprisingly, as reported in Table 2, the aggregation using rotation performs significantly better than ten-crop, and also achieves competitive performance compared to the ensemble with 4 independently trained models. When using both independent ensemble and aggregation with rotation, i.e., the same number of parameters as the ensemble, the accuracy is improved further.

3.3 Evaluation on standard setting

Basic transformations. We demonstrate the effectiveness of our self-supervised augmentation method on various image classification datasets: CIFAR10/100, CUB200, MIT67, Stanford Dogs, and tiny-ImageNet. We first evaluate the effect of aggregated inference $P_{\text{aggregated}}(\cdot|\boldsymbol{x})$ described in Section 2.2: see the SDA+AG column in Table 3. Using rotation as augmentation improves the classification accuracy on all datasets, e.g., 8.60% and 18.8% relative gain over baselines on CIFAR100 and CUB200, respectively. With color permutation, the performance improvements are less significant on CIFAR and tiny-ImageNet, but it still provides meaningful gains on fine-grained datasets, e.g., 12.6% and 10.6% relative gain on CUB200 and Stanford Dogs, respectively.

To maintain the performance of the aggregated inference and reduce its computation cost simultaneously, we apply the self-distillation method described in Section 2.2. As reported in the SDA+SD column in Table 3, it also significantly improves the performance of the single inference (without aggregation) up to 16.1% and 12.4% relatively based on rotation and color permutation, respectively.

Composed transformations. We now show one can improve the performance further by using various combinations of two transformations, rotation and color permutation, on CUB and Stanford Dogs datasets. To construct the combinations, we first choose two subsets T_r and T_c of rotation

Table 3: Classification accuracy (%) using self-supervised data augmentation with rotation and color permutation. SDA+SD and SDA+AG indicate the single inference trained by $\mathcal{L}_{\text{SDA+SD}}$, and the aggregated inference trained by $\mathcal{L}_{\text{SDA}}$, respectively. The relative gain is shown in brackets.

		Rotation		Color Permutation	
Dataset	Baseline	SDA+SD	SDA+AG	SDA+SD	SDA+AG
CIFAR10	92.39	93.26 (+0.94%)	94.50 (+2.28%)	91.51 (-0.95%)	92.51 (+0.13%)
CIFAR100	68.27	71.85 (+5.24%)	74.14 (+8.60%)	68.33 (+0.09%)	69.14 (+1.27%)
CUB200	54.24	62.54 (+15.3%)	64.41 (+18.8%)	60.95 (+12.4%)	61.10 (+12.6%)
MIT67	54.75	63.54 (+16.1%)	64.85 (+18.4%)	60.03 (+9.64%)	59.99 (+9.57%)
Stanford Dogs	60.62	66.55 (+9.78%)	68.70 (+13.3%)	65.92 (+8.74%)	67.03 (+10.6%)
tiny-ImageNet	63.11	65.53 (+3.83%)	66.95 (+6.08%)	63.98 (+1.38%)	64.15 (+1.65%)

Table 4: Classification accuracy (%) depending on the set of transformations. The best accuracy is indicated as bold.

			CUB200		Stanford Dogs	
Rotation	Color permutation	M	SDA+SI	SDA+AG	SDA+SI	SDA+AG
0°	RGB	1	54.24		60.62	
0°, 180°	RGB	2	56.62	58.92	63.57	65.65
0°, 90°, 180°, 270°	RGB	4	60.85	64.41	65.67	67.03
0°	RGB, GBR, BRG	3	52.91	56.47	63.26	65.87
0°	RGB, RBG, GRB, GBR, BRG, BGR	6	56.81	61.10	64.83	67.03
0°, 180°	RGB, GBR, BRG	6	56.14	60.87	65.45	68.75
0°, 90°, 180°, 270°	RGB, GBR, BRG	12	60.74	**65.53**	**66.40**	**69.95**
0°, 90°, 180°, 270°	RGB, RBG, GRB, GBR, BRG, BGR	24	**61.67**	65.43	64.71	67.80

and color permutation, respectively, e.g., $T_r = \{0°, 180°\}$ or $T_c = \{\text{RGB}, \text{GBR}, \text{BRG}\}$. Then, let $T = T_r \times T_c$ be a set of composed transformations of all $t_r \in T_r$ and $t_c \in T_c$. It means that $t = (t_r, t_c) \in T$ rotates an image by t_r and then swaps color channels by t_c. We first evaluate how the set of transformations T affects training. As reported in Table 4, using a larger set T achieves better performance on both the single and aggregated inference than a smaller set $T' \subset T$ in most cases. However, under too many transformations, the aggregation performance can be degraded since the optimization becomes too harder. When using $M = 12$ transformations, we achieve the best performance, 20.8% and 15.4% relatively higher than baselines on CUB200 and Stanford Dogs, respectively.

Next, we demonstrate the aggregation effect at test time depending on various combinations of transformations. To this end, we use the model trained with $M = 12$ transformations, and evaluate the aggregated performance using all possible 2^M combinations. Figure 3 shows the performance of the aggregated inference depending on the size of combinations. As shown in the figure, the average is consistently increasing as the number of transformations increases. We also observe that the maximum performance of $M = 4$ is similar to that of $M = 12$, e.g., the aggregation of $T = \{(0°, \text{GBR}), (90°, \text{BRG}), (180°, \text{RGB}), (270°, \text{RGB})\}$ achieves 65.1%, while that of all 12 transformations achieves 65.4% on CUB200. Namely, one can choose a proper subset of transformations for saving inference cost and maintaining the aggregation performance simultaneously.

3.4 Evaluation on limited-data setting

Limited-data regime. Our augmentation techniques are also effective when only few training samples are available. To evaluate the effectiveness, we first construct sub-datasets of CIFAR100 via randomly choosing $n \in \{25, 50, 100, 250\}$ samples for each class, and then train models with and without our rotation-based self-supervised data augmentation. As shown in Figure 4, our scheme improves the accuracy relatively up to 37.5% under aggregation and 21.9% without aggregation.

Few-shot classification. Motivated by the above results in the limited-data regime, we also apply our SDA+AG method to solve few-shot classification, combined with the state-of-the-art methods, ProtoNet (Snell et al., 2017) and MetaOptNet (Lee et al., 2019) specialized for this problem. Note that our method augments N-way K-shot tasks to NM-way K-shot when using M-way transfor-

Table 5: Average classification accuracy (%) with 95% confidence intervals of 1000 5-way few-shot tasks on mini-ImageNet, CIFAR-FS, and FC100. † and ‡ indicates 4-layer convolutional and 28-layer residual networks (Zagoruyko & Komodakis, 2016), respectively. Others use 12-layer residual networks as Lee et al. (2019). The best accuracy is indicated as bold.

Method	mini-ImageNet		CIFAR-FS		FC100	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML† (Finn et al., 2017)	48.70±1.84	63.11±0.92	58.9±1.9	71.5±1.0	-	-
R2D2† (Bertinetto et al., 2019)	-	-	65.3±0.2	79.4±0.1	-	-
RelationNet† (Sung et al., 2018)	50.44±0.82	65.32±0.70	55.0±1.0	69.3±0.8	-	-
SNAIL (Mishra et al., 2018)	55.71±0.99	68.88±0.92	-	-	-	-
TADAM (Oreshkin et al., 2018)	58.50±0.30	76.70±0.30	-	-	40.1±0.4	56.1±0.4
LEO‡ (Rusu et al., 2019)	61.76±0.08	77.59±0.12	-	-	-	-
MetaOptNet-SVM (Lee et al., 2019)	62.64±0.61	78.63±0.46	72.0±0.7	84.2±0.5	41.1±0.6	55.5±0.6
ProtoNet (Snell et al., 2017)	59.25±0.64	75.60±0.48	72.2±0.7	83.5±0.5	37.5±0.6	52.5±0.6
ProtoNet + **SDA+AG** (ours)	62.22±0.69	77.78±0.51	**74.6**±0.7	**86.8**±0.5	40.0±0.6	55.7±0.6
MetaOptNet-RR (Lee et al., 2019)	61.41±0.61	77.88±0.46	72.6±0.7	84.3±0.5	40.5±0.6	55.3±0.6
MetaOptNet-RR + **SDA+AG** (ours)	**62.93**±0.63	**79.63**±0.47	73.5±0.7	86.7±0.5	**42.2**±0.6	**59.2**±0.5

Table 6: Classification accuracy (%) on imbalance datasets of CIFAR10/100. Imbalance Ratio is the ratio between the numbers of samples of most and least frequent classes. The best accuracy is indicated as bold, and the relative gain is shown in brackets.

Imbalance Ratio ($N_{\text{max}}/N_{\text{min}}$)	Imbalanced CIFAR10		Imbalanced CIFAR100	
	100	10	100	10
Baseline	70.36	86.39	38.32	55.70
Baseline + **SDA+SD** (ours)	74.61 (+6.04%)	89.55 (+3.66%)	43.42 (+13.3%)	60.79 (+9.14%)
CB-RW (Cui et al., 2019)	72.37	86.54	33.99	57.12
CB-RW + **SDA+SD** (ours)	77.02 (+6.43%)	89.50 (+3.42%)	37.50 (+10.3%)	**61.00** (+6.79%)
LDAM-DRW (Cao et al., 2019)	77.03	88.16	42.04	58.71
LDAM-DRW + **SDA+SD** (ours)	**80.24** (+4.17%)	**89.58** (+1.61%)	**45.53** (+8.30%)	59.89 (+1.67%)

mations. As reported in Table 5, ours improves consistently 5-way 1/5-shot classification accuracy on mini-ImageNet, CIFAR-FS, and FC100. For example, we obtain 7.05% relative improvements on 5-shot tasks of FC100.

Imbalanced classification. Finally, we consider a setting where training datasets are imbalanced, where the number of instances per class largely differs and some classes have only a few training instances. For this experiment, we combine our SDA+SD method with two recent approaches, Class-Balanced (CB) loss (Cui et al., 2019) and LDAM (Cao et al., 2019), specialized for this problem. Under imbalanced datasets of CIFAR10/100 which have long-tailed label distributions, our approach consistently improves the classification accuracy as reported in Table 6 (13.3% relative gain on imbalanced CIFAR100 datasets). These results show that our self-supervised data augmentation can be useful for various scenarios of limited training data.

4 Conclusion

In this paper, we proposed a simple yet effective data augmentation approach, where we introduce self-supervised learning tasks as auxiliary tasks and train the model to jointly predict both the label for the original problem and the type of transformation, and validated it with extensive experiments on diverse datasets. We believe that our work could bring in many interesting directions for future research; for instance, one can revisit prior works on applications of self-supervision, e.g., training generative adversarial networks with self-supervision (Chen et al., 2019). Applying our joint learning framework to fully-supervised tasks other than the few-shot or imbalanced classification task, or learning to select tasks that are helpful toward improving the main task prediction accuracy, are other interesting research directions we could further explore.

REFERENCES

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12154–12163, 2019.

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

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

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

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, 2017.

Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

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*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, pp. 7528–7538. Curran Associates Inc., 2018.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6874–6883, 2017.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1DmUzWAW.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

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

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413–420. IEEE, 2009.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgklhAcK7.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12, 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S^4L: Self-supervised semi-supervised learning. *arXiv preprint arXiv:1905.03670*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2547–2555, 2019.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.