TeachAugment: Data Augmentation Optimization Using Teacher Knowledge

Teppei Suzuki

Denso IT Laboratory, Inc.

suzuki.teppei@core.d-itlab.co.jp

Abstract

Optimization of image transformation functions for the purpose of data augmentation has been intensively studied. In particular, adversarial data augmentation strategies, which search augmentation maximizing task loss, show significant improvement in the model generalization for many tasks. However, the existing methods require careful parameter tuning to avoid excessively strong deformations that take away image features critical for acquiring generalization. In this paper, we propose a data augmentation optimization method based on the adversarial strategy called TeachAugment, which can produce informative transformed images to the model without requiring careful tuning by leveraging a teacher model. Specifically, the augmentation is searched so that augmented images are adversarial for the target model and recognizable for the teacher model. We also propose data augmentation using neural networks, which simplifies the search space design and allows for updating of the data augmentation using the gradient method. We show that TeachAugment outperforms existing methods in experiments of image classification, semantic segmentation, and unsupervised representation learning tasks.

1. Introduction

Data augmentation is an important technique used to improve model generalization. To automatically search efficient augmentation strategies for model generalization, AutoAugment [9] has been proposed. Searched data augmentation policies lead to significant generalization improvements. However, AutoAugment requires thousands of GPU hours to search for efficient data augmentation.

Recent studies [17, 22, 30] have demonstrated methods leading to dramatic reductions in search costs with AutoAugment, meaning that computational costs are no longer a problem. In particular, online data augmentation optimization frameworks [18, 28, 49, 58] that alternately update augmentation policies and a target network have not only reduced the computational costs but also simplified the data augmentation search pipeline by unifying the search and training processes.

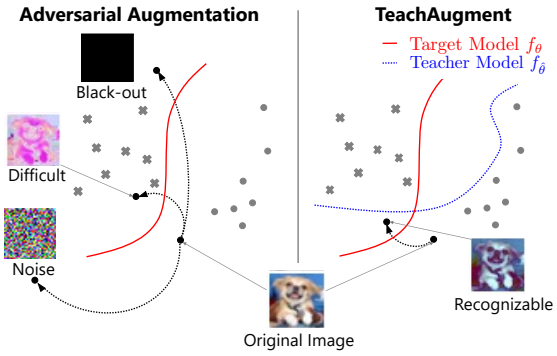


Figure 1. Concept of TeachAugment. Adversarial data augmentation, a baseline method, transforms data to increase loss for the target model f_θ. The augmented data are often meaningless (e.g., black-out and noise images) or difficult to recognize without any constraints. TeachAugment, the proposed method, transforms data so that they are adversarial for the target model but they are recognizable for the teacher model $f_{\hat{\theta}}$. As a result, the augmented images will be more informative than the adversarial data augmentation.

Many online optimization methods are based on an adversarial strategy that searches augmentation maximizing task loss for the target model, which is empirically known to improve model generalization [28, 37, 41, 48, 58]. However, the adversarial data augmentation is unstable without any constraint because maximizing the loss can be achieved by collapsing the inherent meanings of images, as shown in Fig. 1. To avoid the collapse, previous methods regularize augmentation based on prior knowledge and/or restrict the search range of the magnitude parameters of functions in the search space, but there are many tuned parameters that will annoy practitioners.

To alleviate the parameter tuning problem, we propose an online data augmentation optimization method using teacher knowledge, called TeachAugment. TeachAugment

Code: https://github.com/DensoITLab/TeachAugment

is also based on the adversarial data augmentation strategy, but it searches augmentation in the range where the transformed image can be recognizable for a teacher model, as shown in Fig. 1. Unlike previous adversarial data augmentation methods [37, 48, 49, 58], thanks to the teacher model, TeachAugment does not require priors and hyperparameters to avoid the excessively strong augmentation that collapses the inherent meanings of images. As a result, TeachAugment does not require parameter tuning to ensure that the transformed images are recognizable.

Moreover, we propose data augmentation using neural networks that represent two functions, geometric augmentation and color augmentation. Our augmentation model applies only two transformations to data but they can represent most functions included in the search space of AutoAugment and their composite functions. The use of the neural networks has two advantages compared to conventional augmentation functions: (i) we can update the augmentation parameters using the gradient method with backpropagation, and (ii) we can reduce the number of functions in the search space from tens of functions to two functions. In particular, because of the latter advantage, practitioners only need to consider the range of the magnitude parameters for the two functions when they adjust the size of the search space for better convergence.

Contribution. Our contribution is summarized as follows: (1) We propose an online data augmentation optimization framework based on the adversarial strategy using teacher knowledge called TeachAugment. TeachAugment makes the adversarial augmentation more informative without careful parameter tuning by leveraging the teacher model to avoid collapsing the inherent meaning of images. (2) We also propose data augmentation using neural networks. The proposed augmentation simplifies the search space design and enables updating of its parameters by the gradient method in TeachAugment. (3) We show that TeachAugment outperforms previous methods, including online data augmentation [28, 58] and state-of-the-art augmentation strategies [10, 39] in classification, semantic segmentation, and unsupervised representation learning tasks without adjusting the hyperparameters and size of the search space for each task.

2. Related work

As conventional data augmentation for image data, geometric and color transformations are widely used in deep learning. Besides, advanced data augmentations [12, 21, 23, 52, 55, 57, 60] have been recently developed and they have improved accuracy on image recognition tasks. Data augmentation not only enhances the image recognition accuracy, but also plays an important role in recent unsupervised representation learning [5–7, 15, 19] and semi-supervised learning [37, 45, 48]. While the data augmentation usually improves model generalization, it sometimes hurts performance or induces unexpected biases. Thus, one needs to manually find the effective augmentation policies based on domain knowledge to enhance model generalization.

Cubuk et al. [9] proposed a method to automatically search for effective data augmentation, called AutoAugment. AutoAugment outperforms hand-designed augmentation strategies and shows state-of-the-art performances on various benchmarks. Data augmentation search has become a research trend, and many methods have been proposed [10, 17, 18, 28–31, 33, 39, 43, 49, 51, 53, 58].

We roughly categorize them into two types: a *proxy task based* method and a *proxy task free* method. The proxy task based methods [17, 29, 30, 51, 58] search data augmentation strategies on *proxy tasks* that use subsets of the training data and/or small models to reduce computational costs. Thus, policy searches using the proxy task based method might be sub-optimal. The proxy task free methods [10, 28, 33, 39, 58] directly search data augmentation strategies on the target network with all training data. Thus, policies obtained with this method are potentially optimal.

In proxy task free methods, several approaches, such as RandAugment [10] and TrivialAugment [39], randomize the parameters searched and reduce the size of the search space. Other methods, such as Adversarial AutoAugment [58] and PointAugment [28], update augmentation policies in an online manner, meaning that they alternately update a target network and augmentation policies. The online optimization methods simplify data augmentation optimization frameworks by unifying the search and training processes. However, these methods are fraught with minor problems. For example, PointAugment [28] unjustly binds the difficulty of augmented images, Adversarial AutoAugment [58] manually restricts the search space to guarantee convergence, and OnlineAugment [49] has many hyperparamters for regularization. These problems stem from reliance on the adversarial strategy that searches augmentation maximizing task loss. In other words, these problems are induced to ensure that the transformed images are recognizable.

In this work, we focus on proxy task free methods updating the policies in an online manner for two reasons: (i) they can directly search data augmentation strategies on the target network with all training data, and (ii) they unify the search and training processes, simplifying the framework.

3. Data augmentation optimization using teacher knowledge

3.1. Preliminaries

Let $x \sim \mathcal{X}$ and a_ϕ be an image sampled from dataset $\mathcal{X}$ and an augmentation function parameterized by ϕ, respectively. In conventional data augmentation, ϕ corresponds

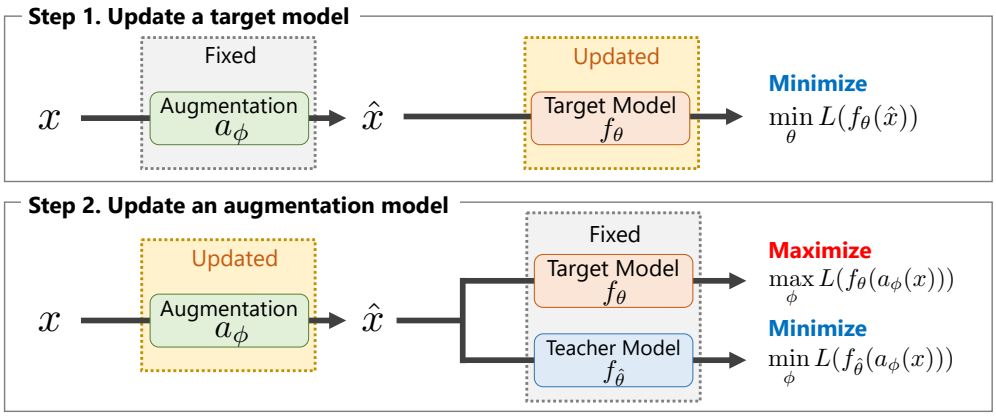


Figure 2. Overview of training procedure for TeachAugment. Our method alternately updates the target model f_θ and the augmentation model a_ϕ (i.e., step 1 and step 2 are repeated).

to the magnitude of augmentation, while in this work, it corresponds to the neural network's parameters. In general training procedures using data augmentation, the mini-batch samples are transformed by a_ϕ and fed into a target network f_θ (i.e., $f_\theta(a_\phi(x))$). Then, the parameters of the target network are updated to minimize task loss L in the stochastic gradient descent. This training procedure is represented as $\min_\theta \mathbb{E}_{x\sim\mathcal{X}} L(f_\theta(a_\phi(x))$. Note that we omitted the target label because we consider not only supervised learning but also unsupervised representation learning in this work.

In addition, adversarial data augmentation searches the parameters ϕ maximizing the loss. The objective is defined as $\max_\phi \min_\theta \mathbb{E}_{x\sim\mathcal{X}} L(f_\theta(a_\phi(x))$. This objective is sometimes solved by alternately updating ϕ and θ [28, 58]. The adversarial data augmentation is empirically known to improve model generalization [41, 58]. However, it does not work without some regularization or restriction in the size of the search space because the maximization with respect to ϕ can be achieved by collapsing the inherent meanings of x. Thus, instead of using regularization based on prior knowledge, we utilized a teacher model to avoid the collapse.

3.2. TeachAugment

Let $f_{\hat{\theta}}$ be the teacher model, which allows for any model as long as it is different from the target model f_θ.[1] In this work, we suggest two types of the teacher model, a pretrained teacher and an EMA teacher whose weights are updated as an exponential moving average of the target model's weights. We provide detailed definitions and evaluate the effect of teacher model choices in Sec. 5.2.

[1] In most of our experiments, we define the teacher model as the same model as the target model but with different parameters. Thus, the same symbol f is used for the teacher model and the target model although it may be a little complicated.

The proposed objective is defined as follows:

$$\max_\phi \min_\theta \mathbb{E}_{x\sim\mathcal{X}} \left[L(f_\theta(a_\phi(x))) - L(f_{\hat{\theta}}(a_\phi(x))) \right]. \quad (1)$$

For the target model, this objective has the same properties as the adversarial data augmentation, but the augmentation function needs to minimize the loss for the teacher model, in addition to maximizing the loss for the target model. The augmentation that is obtained in this manner avoids collapsing the inherent meanings of images because the loss for the teacher model will explode when the transformed images are unrecognizable. In other words, the introduced teacher loss requires the augmentation function to transform images so that they are adversarial for the target model in the range where they are recognizable for the teacher model.

As shown in Fig. 2, the objective is solved by alternately updating the augmentation function and the target model in the stochastic gradient descent, similar to previous methods [18, 28, 58]. We first update the target network for n_{inner} steps, and then update the augmentation function. A pseudo-code can be found in the appendix. Note that the augmentation function is updated by the gradient method because our proposed augmentation using the neural networks introduced in Sec. 4 is differentiable with respect to ϕ. We refer to the augmentation strategy following Eq. 1 as *TeachAugment*.

Unlike previous methods [28, 49], TeachAugment does not regularize augmentation functions based on the domain knowledge, such as cycle consistency and smoothness [49]. It also does not bind the difficulty of the transformed images as in [28] to ensure that the transformed images are recognizable.

3.3. Improvement techniques

The training procedure for TeachAugment is similar to that for generative adversarial networks (GANs) [14]

and actor-critic methods in reinforcement learning [26, 47], which alternately update two networks. Practitioners in both fields have amassed a large number of strategies to mitigate instabilities and improve training [42]. TeachAugment also benefits from three techniques used in both fields, *experience replay*, *non-saturating loss*, and *label smoothing*. Moreover, we introduced color regularization to mitigate inconsistency between color distributions of images before and after data augmentation. The techniques introduced here are also applicable for other online methods [28, 49, 58].

Non-saturating loss. For classification tasks, the loss function L is usually defined as the cross-entropy loss, $L(f_\theta(a_\phi(x))) = \sum_{k=1}^{K} -y_k \log f_\theta(a_\phi(x))_k$, where $y \in \{0,1\}^K$ and K denote the one-hot ground truth label and the number of classes, respectively. In this case, the gradient of the first term in Eq. (1) often saturates in the maximization problem with respect to ϕ when the target model's predictions are very confident. Thus, we use $\sum_{k=1}^{K} y_k \log(1 - f_\theta(a_\phi(x))_k)$ when updating the augmentation model rather than $\sum_{k=1}^{K} -y_k \log f_\theta(a_\phi(x))_k$. This technique has been used in GANs [14].

The non-saturating loss is a key factor for TeachAugment; it improves error rates of WideResNet-28-10 [56] on CIFAR-100 [27] from 18.7% to 17.4% (a baseline's error rate is 18.4%). Thus, we basically use the non-saturating loss for updating augmentation models in our experiments.

Experience replay. In reinforcement learning, experience replay [32, 38] stores actions chosen by the actor in the past and reuses them to update the critic. We apply this technique to our method by storing the augmentation models and prioritizing them in a manner similar to prioritized experience replay [44]. Then, the target network is updated using the augmentation model randomly sampled from the buffer following their priorities.

Let p_i be a priority of the i-th stored augmentation model. We compute the priority as $p_i = \gamma^{S-i}$, where S denotes the number of augmentation models stored in the buffer, γ denotes a decay rate, and we set γ to 0.99 for models with a long training schedule (epochs > 1000) and 0.9 otherwise. In our experiments, we stored the augmentation model every n_{buffer} epochs. For image classification on CIFAR-10, CIFAR-100, and semantic segmentation, n_{buffer} was set to 10, and for other tasks and datasets, it was set to 1.

Label smoothing. Label smoothing is a technique that replaces the one-hot labels with $\hat{y}_k = (1-\epsilon)y_k + \epsilon/K$, where $\epsilon \in [0, 1)$ is a smoothing parameter. For our method, the label smoothing prevents gradients from exploding when the target model's predictions are very confident under the non-saturating loss. In particular, such a situation tends to occur for easy tasks or strong target models. To mitigate exploding gradients, we used the smoothed labels for the first term in Eq. (1) when updating the augmentation model. Note that for a fair comparison, we did not apply it when updating the target model.

Color regularization. In practice, the color augmentation model tends to transform the pixel colors outside the color distribution of the training data. As a result, the augmented images will be out-of-distribution data, which may hurt the recognition accuracy for the in-distribution samples. To align the color distributions before and after augmentation, we regularized the color augmentation model by introducing sliced Wasserstein distance (SWD) [3] between pixel colors before and after the augmentation. SWD is a variant of Wasserstein distance that represents a distance between two distributions.

We define the color regularization term as follows:

$$L_{\text{color}}(\{x^b\}_b^B, \{\tilde{x}^b\}_b^B) = \sum_i \text{SWD}(\{x_i^b\}_b^B, \{\tilde{x}_i^b\}_b^B), \quad (2)$$

where $\{x_i^b\}_b^B$ denotes a set of i-th pixels of images in a mini-batch with a batch size of B, and $\{\tilde{x}^b\}_b^B$ denotes color-augmented images defined in Eq. (4). Because costs of computing SWD at each pixel position depends linearly on the image resolution, we can compute SWD for high-resolution images handled in semantic segmentation with low computational resources. Then, in the stochastic gradient descent, the gradient with respect to ϕ in each iteration is represented as follows:

$$\frac{\partial}{\partial \phi} \frac{1}{B} \sum_b^B \left[L(f_\theta(a_\phi(x^b))) - L(f_{\tilde{\theta}}(a_\phi(x^b))) \right] - \lambda L_{\text{color}}(\{x^b\}_b^B, \{\tilde{x}^b\}_b^B), \quad (3)$$

where λ is a hyperparameter controlling the effect of the regularization that was set to 10 in our experiments.

4. Data augmentation using neural networks

We propose data augmentation using neural networks parameterized by ϕ, which consists of two models, a color augmentation model c_{ϕ_c} and a geometric augmentation model g_{ϕ_g}. Thus, a_ϕ is defined as the composite function of c_{ϕ_c} and g_{ϕ_g}, $a_\phi = g_{\phi_g} \circ c_{\phi_c}$, and the parameter ϕ is a set of ϕ_c and ϕ_g, $\phi = \{\phi_c, \phi_g\}$. Our augmentation model enables updating of its parameters by the gradient method and construction of the search space with only two functions.

The augmentation procedure is illustrated in Fig. 3. Given an image $x \in \mathbb{R}^{M \times 3}$, where M denotes the number of pixels and 3 corresponds to the RGB color channels, the color augmentation is applied with probability of $p_c \in (0, 1)$, and then the geometric augmentation is applied with probability of $p_g \in (0, 1)$.

The color augmentation is defined as follows:

$$\tilde{x}_i = t(\alpha_i \odot x_i + \beta_i), \ (\alpha_i, \beta_i) = c_{\phi_c}(x_i, z, c), \quad (4)$$

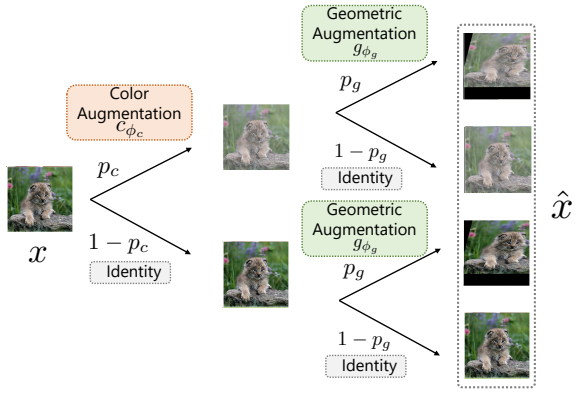


Figure 3. Data augmentation procedure. Our data augmentation consists of the color augmentation c_{ϕ_c} and the geometric augmentation g_{ϕ_g}. Each augmentation is applied to an input image x with probabilities p_c and p_g.

where $\alpha_i,\ \beta_i \in \mathbb{R}^3$ denote scale and shift parameters; $\odot$ denotes the element-wise multiplication between vectors; $t(\cdot)$ denotes the triangle wave, $t(x) = \arccos(\cos(x\pi))/\pi$, which ensures $\tilde{x}_i \in [0, 1]$; $z \sim \mathcal{N}(0, I_N)$, where $\mathcal{N}(0, I_N)$ denotes N-dimensional unit Gaussian distribution, and c is an optional context vector. In our experiments, we used a one-hot ground truth label as c for image classification and omitted it for other tasks. The color augmentation model can transform the input image into any image *in principle* when the augmentation model is sufficiently large because of the universal approximation theorem.

The geometric augmentation is defined as follows:

$$\hat{x} = \mathrm{Affine}(\tilde{x}, A + I),\ A = g_{\phi_g}(z, c), \tag{5}$$

where $\mathrm{Affine}(\tilde{x}, A + I)$ denotes an affine transformation of $\tilde{x}$ with a parameter $A + I$; $I \in \mathbb{R}^{2\times3}$ denotes a matrix where $\forall i, I_{ii} = 1$ and 0 otherwise, which makes the affine transformation the identity mapping $\mathrm{Affine}(x, I) = x$; $A \in \mathbb{R}^{2\times3}$ denotes the residual parameter, and c and z are the same vectors used in Eq. (4).

The geometric augmentation model can be defined by transformations other than the affine transformation, similar to [24]. However, the affine transformation can represent all geometric transformations in the search space of AutoAugment and their composite functions. Thus, we considered only the affine transformation in this work.

In addition to ϕ_c and ϕ_g, we also learned the probabilities p_c and p_g by using the gradient method. However, the decision process of applying the augmentation is non-differentiable with respect to p_c and p_g. To make it differentiable with respect to the probabilities, we used the same approach as in previous works [17, 29]. A detailed pipeline can be found in the appendix.

5. Experiments

We evaluate our method with three tasks. For an ablation study and comparison with existing automatic data augmentation search methods, we evaluate our method on image classification tasks. We train WideResNet-40-2 (WRN-40-2) [56], WideResNet-28-10 (WRN-28-10) [56], Shake-Shake (26 2×96d) [13], and PyramidNet [16] with ShakeDrop regularization [54] on CIFAR-10 and CIFAR-100 [27], and train ResNet-50 [20] on ImageNet [11]. The training and evaluation protocols are the same settings as in previous work [30].

In addition to the above experiments, we examine our method with semantic segmentation and unsupervised representation learning. For semantic segmentation, we train FCN-32s [34], PSPNet [59], and Deeplabv3 [4] on Cityscapes [8]. The training protocol is the same as [59]. For unsupervised representation learning, we pretrain ResNet-50 [20] on ImageNet [11] using SimSiam [7] with various data augmentation and then evaluate the model following the linear evaluation protocol [7].

More details can be found in the appendix.

5.1. Implementation details

We construct the geometric augmentation model using a multi-layer perceptron (MLP) and the color augmentation model using two MLPs that receive an RGB vector and a noise vector as inputs and then add up the outputs. We apply the sigmoid function to the outputs of each augmentation model and normalize the outputs in a range of $A \in (-0.25, 0.25)^{2\times3}$, $\alpha \in (0.6, 1.4)$, and $\beta \in (-0.5, 0.5)$. The dimension of the noise vector z is set to 128. For stochasticity, Dropout [46] is applied after the linear layers, except to the output layer. We initialize the weights of the output layer as zero to make the augmentation the identity mapping in the initial state. We use AdamW optimizer [35] to train the augmentation model. All the hyperparameters of AdamW (e.g., learning rate and weight decay) are set to the PyTorch default parameters [40]. More details can be found in the appendix.

5.2. Ablation study

Effect of teacher model choices. We first investigate the effect of teacher model choices. We trained WRN-40-2 with two types of teacher models: a teacher that was pretrained with the baseline augmentation, and an EMA teacher that is exponential moving average of the target model (i.e., $\hat{\theta} \leftarrow \xi\hat{\theta} + (1 - \xi)\theta$). We set the decay rate ξ to 0.999, following [50]. For the pretrained teacher, we pretrained WRN-40-2, which is the same model as the target model, and WRN-28-10, which is a stronger model than the target model.

The results are shown in Tab. 1. For both datasets, the EMA teacher achieves lower error rates than the others. The

Teacher	WRN-40-2	WRN-28-10	EMA
CIFAR-10	4.4	**3.7**	**3.7**
CIFAR-100	21.3	21.6	**20.3**

Table 1. Effect of teacher models. We report the error rates (%) of WideResNet-40-2 trained with various teacher models. WRN-40-2 and WRN-28-10 are pretrained with the baseline augmentation. EMA is an exponential moving average of the target model.

Dataset	CIFAR-10	CIFAR-100
All	**2.5**	16.8
w/o Color Reg.	3.0	**16.5**
w/o Experience Replay	2.7	17.3
w/o Label Smoothing	3.0	17.3
w/o all techniques	3.0	17.4

Table 2. Error rates (%) of WideResNet-28-10 on CIFAR-10 and CIFAR-100 with various technique choices.

augmentation model may cause overfitting to the teacher model when the pretrained teacher is used because they are not updated during training. The EMA teacher would prevent overfitting by updating parameters at the same time as the target model during training, bringing more improvement than the pretrained teachers.

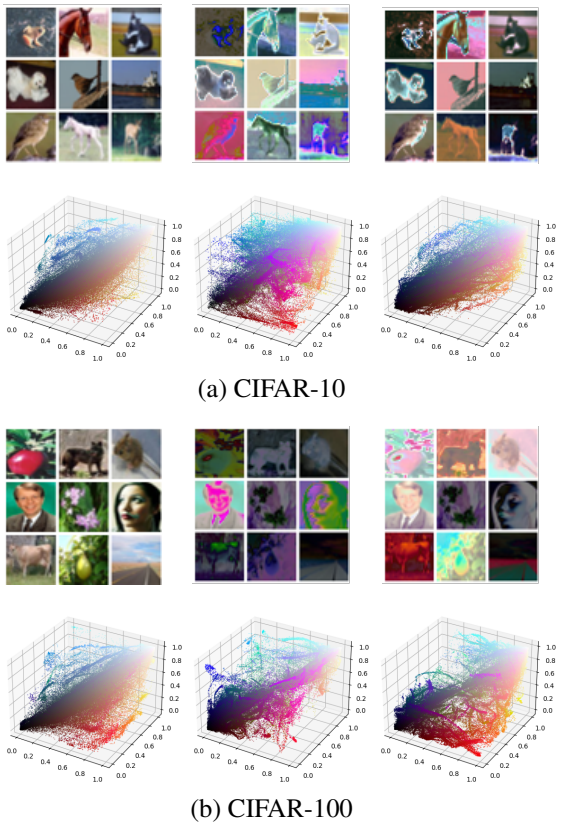
Interestingly, we found the stronger teacher was not a better teacher. In fact, WRN-28-10 demonstrates error rates comparable to the EMA teacher in CIFAR-10, but in CIFAR-100, it was slightly worse than WRN-40-2.

In the rest of the experiments, we used the EMA teacher for TeachAugment because the EMA teacher not only has lower error rates but also eliminates the architecture choices of the teacher model.

Effect of stabilizing techniques. We next investigate the effect of stabilizing techniques introduced in Sec. 3.3. We trained WRN-28-10 on CIFAR-10 and CIFAR-100, with the exception of each technique.

The results are shown in Tab. 2. Except for the color regularization on CIFAR-100, all of the techniques contributes to the improvement of error rates.

We visualize the effect of the color regularization in Fig. 4. The color distributions are aligned by the color regularization. In particular, the augmented images without regularization lose brightness in many pixels with CIFAR-100. However, alignment of the color distribution improve the error rates only for CIFAR-10. This would be due to the color diversity of CIFAR-10 being narrower than CIFAR-100, which can be seen from the color distributions in Fig. 4. In other words, in the case of training without color regularization, the color augmentation produces out-of-distribution colors, and the obtained images hurt the recognition accuracy of the target model for the in-distribution samples. The transformed image from CIFAR-10 tends to be this kind of



(a) CIFAR-10

(b) CIFAR-100

Figure 4. Augmented images and corresponding color distributions on CIFAR-10 and CIFAR-100. From left to right, original images, augmented images from the model trained without color regularization, and augmented images from the model trained with color regularization. We randomly picked 128 images and plotted their pixel colors (i.e., each point corresponds to a pixel).

an out-of-distribution sample due to the low diversity of colors.

We evaluate label smoothing with various smoothing parameter values, and the results are shown in Fig. 5. For the easy tasks of CIFAR-10, label smoothing with the large ϵ works well. As already described in Sec. 3.3, the gradient of the non-saturating loss tends to be large for easy tasks or strong models, meaning that the model's predictions tend to be very confident. Thus, a larger ϵ avoids exploding gradients and stabilizes training for CIFAR-10.

Evaluation of the proposed objective function. We evaluate the effectiveness of the proposed objective function, eq. (1). As baseline methods, we replace the objective function in our framework with the loss of adversarial AutoAugment (Adv. AA) [58] and PointAugment (PA) [28]. For a fair comparison, all of the models were trained with the same protocol and the difference between them was the objective function. The detailed setting is described in the appendix.

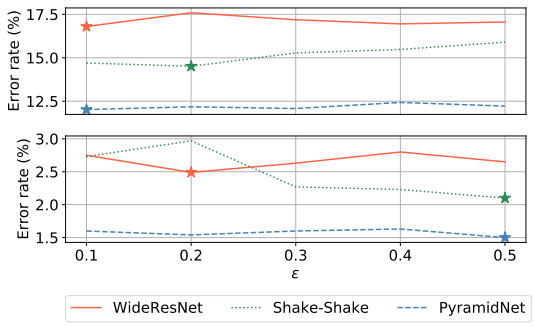


Figure 5. Effect of ϵ on label smoothing. The upper figure shows error rates on CIFAR-100, and the lower figure shows the error rates on CIFAR-10. Stars indicate ϵ achieving minimum error rates.

Objective	Baseline	Adv. AA [58]	PA [28]	Ours
CIFAR-10	3.1	3.6	2.9	**2.5**
CIFAR-100	18.4	19.4	17.5	**16.8**

Table 3. Error rates (%) of WideResNet-28-10 trained with various online optimization frameworks. All results are obtained under the same augmentation models and training protocol, except for the objective function.

The results are shown in Tab. 3. The error rates of Adv. AA degrades from the baseline because the augmentation model produces the unrecognizable images that confuse the target model. For Adv. AA, one needs to carefully tune the size of the search space to guarantee the convergence [58] but we do not adjust it. In addition, for such instability, the proposed augmentation model would be unsuitable because of its property, i.e., the color augmentation model can transform the input image into any image. In other words, the proposed augmentation would work only for the methods that guarantee transformed images to be recognizable. Our method achieves better error rates than PointAugment. PointAugment controls the upper bound of the loss for the augmented data with a dynamic parameter, but our method has no such restriction. As a result, our method provides more diverse transformations and better improvements in the error rates than PointAugment. The augmented images are visualized in the appendix.

Evaluation of the proposed augmentation model. We compare the proposed augmentation model to a differentiable data augmentation pipeline (DDA) proposed in [17] and augmentation models used in OnlineAugment (OA) [49]. The results are shown in Tab. 4. The improvement of DDA from baseline is less than that of ours. DDA uses some techniques (e.g., relaxation [25, 36] and straight-through estimator [1]) to make some data augmentation functions differentiable, but these techniques induce inaccurate gradients and the vanishing gradient problem that would make it difficult to learn effective augmentation in the TeachAugment framework. OA is much worse than baseline. OnlineAugment regularizes augmentation models based on prior knowledge, such as cycle consistency and smoothness, instead of bounding the search space. However, TeachAugment does not impose strong regularization. The lack of regularization would be unsuitable for augmentation with the unbounded search space. We note that the proposed augmentation is better than DDA and OA for TeachAugment, but as seen in Tab. 3, it does not work in all cases.

Augmentation	Baseline	DDA	OA	Ours
CIFAR-10	3.1	2.8	16.7	**2.5**
CIFAR-100	18.4	18.0	26.2	**16.8**

Table 4. Error rates (%) of WideResNet-28-10 with DDA [17], the augmentation models of OnlineAugment [49] (OA) and the proposed augmentation on CIFAR-10 and CIFAR-100. Each augmentation is optimized by the TeachAugment strategy.

5.3. Image classification

We compare our method to several previous methods: AutoAugment (AA) [9], PBA [22], Fast AA [30], Faster AA [17], DADA [29], RandAugment (RandAug) [10], OnlineAugment (OnlineAug) [49], and Adv. AA [58].

For CIFAR-10 and CIFAR-100, we used the smoothing parameter ϵ achieving the best error rate in the previous section. We set ϵ to 0.1 for training on ImageNet, following the tendency in Fig. 5 (i.e., the smaller ϵ tends to be better for difficult tasks).

We show the comparison results in Tab. 5. Our method achieves error rates comparable to other method except for Adv. AA, which uses multiple augmented samples per mini-batch. In particular, our method achieves the lowest top-1 error rate on ImageNet among the methods without a multiple augmentation strategy.

5.4. Semantic segmentation

We also evaluate our method on Cityscapes [8]. We implement widely used models with a ResNet-101 backbone, FCN-32s [34], PSPNet [59], and Deeplabv3 [4]. As baseline methods, we adopted RandAugment [10] and TrivialAugment [39]. These methods also does not need careful parameter tuning because they have few or no hyperparameters.

We show the results in Tab. 6. Our method achieves the best mIoU for each model. RandAugment hurts the mIoU for FCN-32s, and TrivialAugment does not improve the mIoU for any of the models. In fact, it has been reported that TrivialAugment does not work well for tasks other than image classification [39]. We suspect that the search spaces of RandAugment and TrivialAugment are not suitable for

	CIFAR-10			CIFAR-100			ImageNet
	WRN-28-10	Shake-Shake	PyramidNet	WRN-28-10	Shake-Shake	PyramidNet	ResNet-50
Baseline	3.9	2.9	2.7	18.8	17.1	14.0	23.7/6.9
Cutout [12]	3.1	2.6	2.3	18.4	16.0	12.2	-
AA [9]	2.6	2.0	1.5	17.1	14.3	10.7	22.4/6.4
PBA [22]	2.6	2.0	1.5	16.7	15.3	10.9	-
RandAug [10]	2.7	2.0	1.5	16.7	-	-	22.4/6.2
Fast AA [30]	2.7	2.0	1.7	17.3	14.6	11.7	22.4/6.3
Faster AA [17]	2.6	2.0	-	17.3	15.0	-	23.5/6.8
DADA [29]	2.7	2.0	1.7	17.5	15.3	11.2	22.5/6.5
OnlineAug [49]	2.4	-	-	16.6	-	-	22.4/-
Adv. AA† [58]	(1.9)	(1.9)	(1.4)	(15.5)	(14.1)	(10.4)	(20.6/5.5)
Ours	2.5	2.0	1.5	16.8	14.5	11.8	22.2/6.3

Table 5. Test set top-1 error rates (%) on CIFAR-10 and CIFAR-100 and validation set top-1/top-5 error rates on ImageNet. Please note that Adv. AA† uses multiple augmented samples per mini-batch.

Model	Baseline	RA [10]	TA [39]	Ours
FCN-32	71.7	71.0	71.3	**72.1**
PSPNet	77.7	**78.8**	77.5	**78.8**
Deeplabv3	78.4	79.3	78.9	**79.4**

Table 6. Validation set mIoU (%) on Cityscapes. RA and TA denote RandAugment [10] and TrivialAugment [39], respectively.

#Epochs	100	200	400
Baseline [7]	68.1	70.0	70.8
RandAugment [10]	68.0	70.0	70.7
TrivialAugment [39]	62.7	68.7	**71.3**
Ours	**68.2**	**70.2**	71.0

Table 7. ImageNet linear evaluation accuracy (%) for various pretraining epochs. We set batch size to 256 for pretraining.

semantic segmentation tasks and model capacities. However, our method improves mIoU for all conditions without adjusting parameters from the classification tasks.

5.5. Unsupervised representation learning

Finally, we evaluate our method with unsupervised representation learning tasks using SimSiam [7]. To generate two different views of the same image, we used two augmentation models, a_{ϕ_1} and a_{ϕ_2}. The detailed settings can be found in the appendix.

The results are shown in Tab. 7. TrivialAugment causes underfitting for 100 and 200 epoch training due to the diversity of augmentation induced by the strong randomness. RandAugment does not contribute to the improvement of accuracy. Our proposed method consistently improves the accuracy for all training schedules, although the improvements are slight. This would be because the online optimization frameworks have aspects similar to curriculum learning [2]. In other words, our method adjusts the augmentation magnitude according to the learning progress of the target model.

6. Conclusion

We proposed an online data augmentation optimization method called TeachAugment that introduces a teacher model into the adversarial data augmentation and makes it more informative without the need for careful parameter tuning. We also proposed neural network based augmentation that simplified the search space design and enabled updating of the data augmentation with the gradient method. In experiments, our method outperformed existing data augmentation search frameworks, including state-of-the-art methods, on image classification, semantic segmentation, and unsupervised representation learning tasks without adjusting the hyperparameters for each task.

Limitation. The proposed color augmentation cannot represent transformation using global information of a target image, such as Equalize and AutoContrast, because of the lack of global information in the input. Such transformations may be realized using the color histogram as the context vector, although at the expense of computational cost, especially for high-resolution images. Moreover, we only focused on geometric and color augmentation, but there are many advanced augmentation that are not categorized with them, for example, cutout [12] and mixup [57]. Considering such augmentations will be future work.

The training time for TeachAugment is approximately three times longer than the conventional training procedure because updating the augmentation model requires the forward and backward computation of the target model and the teacher model, in addition to updating the target network. However, the training time of TeachAugment is almost the same as other online methods [28, 49] and is a realistic time in comparison to AutoAugment [9]. We believe that it is a negligible problem under the recent advances of computational power.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 7

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 8

[3] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. 4

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

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

[7] 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. 2, 5, 8

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 7

[9] 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. 1, 2, 7, 8

[10] 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. 2, 7, 8

[11] 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

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2, 8

[13] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 5

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3, 4

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[16] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5927–5935, 2017. 5

[17] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020. 1, 2, 5, 7, 8

[18] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Meta approach to data augmentation optimization. *arXiv preprint arXiv:2006.07965*, 2020. 1, 2, 3

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[20] 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. 5

[21] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 2

[22] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019. 1, 7, 8

[23] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 2

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015. 5

[25] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 7

[26] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000. 4

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4, 5

[28] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6378–6387, 2020. 1, 2, 3, 4, 6, 7, 8

[29] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada:

Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020. 2, 5, 7, 8

[30] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019. 1, 2, 5, 7, 8

[31] Chen Lin, Minghao Guo, Chuming Li, Xin Yuan, Wei Wu, Junjie Yan, Dahua Lin, and Wanli Ouyang. Online hyper-parameter learning for auto-augmentation strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6579–6588, 2019. 2

[32] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992. 4

[33] Tom Ching LingChen, Ava Khonsari, Amirreza Lashkari, Mina Rafi Nazari, Jaspreet Singh Sambee, and Mario A Nascimento. Uniformaugment: A search-free probabilistic data augmentation approach. *arXiv preprint arXiv:2003.14348*, 2020. 2

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 5, 7

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[36] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 7

[37] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1, 2

[38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 4

[39] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *arXiv preprint arXiv:2103.10158*, 2021. 2, 7, 8

[40] 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 *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[41] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2018. 1, 3

[42] David Pfau and Oriol Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016. 4

[43] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2021. 2

[44] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015. 4

[45] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 2

[46] 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. 5

[47] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000. 4

[48] Teppei Suzuki and Ikuro Sato. Adversarial transformations for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5916–5923, 2020. 1, 2

[49] Zhiqiang Tang, Yunhe Gao, Leonid Karlinsky, Prasanna Sattigeri, Rogerio Feris, and Dimitris Metaxas. Onlineaugment: Online data augmentation with less domain knowledge. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 313–329. Springer, 2020. 1, 2, 3, 4, 7, 8

[50] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017. 5

[51] Keyu Tian, Chen Lin, Ming Sun, Luping Zhou, Junjie Yan, and Wanli Ouyang. Improving auto-augment via augmentation-wise weight sharing. *arXiv preprint arXiv:2009.14737*, 2020. 2

[52] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018. 2

[53] Longhui Wei, An Xiao, Lingxi Xie, Xiaopeng Zhang, Xin Chen, and Qi Tian. Circumventing outliers of autoaugment with knowledge distillation. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 2

[54] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *IEEE Access*, 7:186126–186136, 2019. 5

[55] 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

[56] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 4, 5

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 8

[58] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*, 2019. 1, 2, 3, 4, 6, 7, 8

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5, 7

[60] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 2