SALIENCY GRAFTING: INNOCUOUS ATTRIBUTION-GUIDED MIXUP WITH CALIBRATED LABEL MIXING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Mixup scheme of mixing a pair of samples to create an augmented training sample has gained much attention recently for better training of neural networks. A straightforward and widely used extension is to combine Mixup and regional dropout methods: removing random patches from a sample and replacing it with the features from another sample. Albeit their simplicity and effectiveness, these methods are prone to create harmful samples due to their randomness. In recent studies, attempts to prevent such a phenomenon by selecting only the most informative features are gradually emerging. However, this maximum saliency strategy acts against their fundamental duty of sample diversification as they always deterministically select regions with maximum saliency, injecting bias into the augmented data. To address this problem, we present *Saliency Grafting*, a novel Mixup-like data augmentation method that captures the best of both ways. By stochastically sampling the features and 'grafting' them onto another sample, our method effectively generates diverse yet meaningful samples. The second ingredient of *Saliency Grafting* is to produce the label of the grafted sample by mixing the labels in a saliency-calibrated fashion, which rectifies supervision misguidance introduced by the random sampling procedure. Our experiments under CIFAR and ImageNet datasets show that our scheme outperforms the current state-of-the-art augmentation strategies not only in terms of classification accuracy, but is also superior in coping under stress conditions such as data corruption and data scarcity. The code will be released.

1 INTRODUCTION

Modern deep neural networks (DNNs) have achieved unprecedented success in various fields including computer vision (Krizhevsky et al., 2012), natural language processing (Devlin et al., 2018) and speech processing (Chan et al., 2016). However, due to their over-parameterized nature, DNNs require an immense amount of training data to generalize well for unseen data. Otherwise, DNNs are predisposed to memorize the training samples and exhibit lackluster performance on the unseen data - in other words, overfit.

Acquiring a sufficient amount of data for a given task is not always possible as it consumes manpower and budget. One common approach to combat data scarcity is data augmentation, which aims to enlarge the effective size of a dataset by producing virtual samples from the training data by means such as injecting noise (Amodei et al., 2016) or cropping out regions (DeVries & Taylor, 2017). Datasets fortified with these augmented samples are shown to effectively improve the generalization performance of the trained model. Furthermore, data augmentation is proven to be effective not only for promoting generalization but also in boosting the robustness of a model (Hendrycks et al., 2019) and acquiring visual representations without human supervision (Chen et al., 2020).

To this end, conventional augmentation methods focused on creating new images by transforming a given image using means such as flipping, resizing and more. However, a recently proposed augmentation method called Mixup (Zhang et al., 2017) proposed the idea of crafting a new sample out of a pair of samples by taking a convex combination of them. Inspired by this pioneering work, Yun et al. (2019) proposed CutMix, a progeny of Mixup and Cutout (DeVries & Taylor, 2017), which crops a random region of an image and pasting it on another. These methods are able to generate a wider variety of samples while effectively compensating the information loss by actions such as

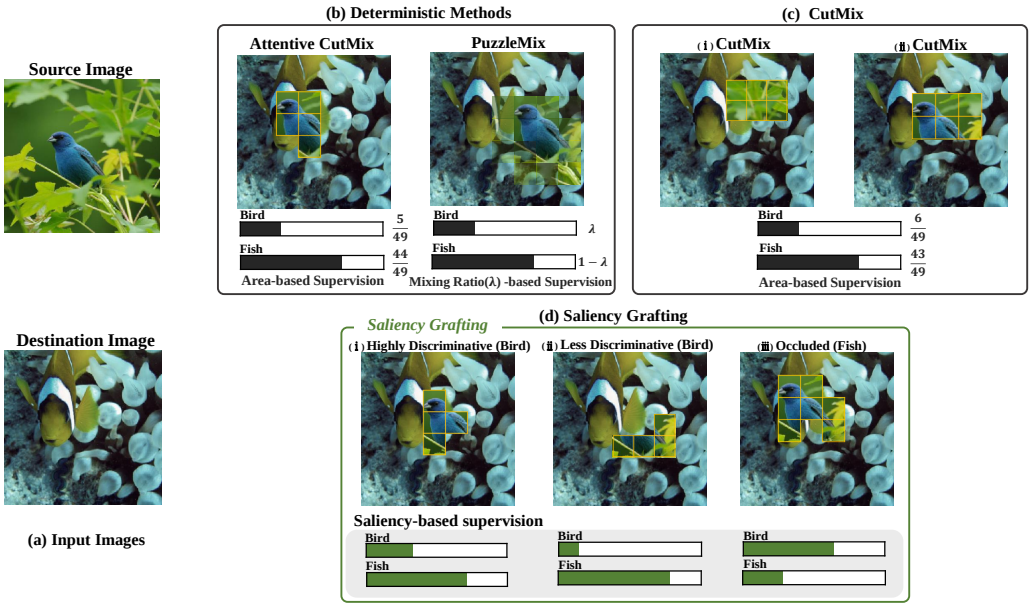


Figure 1: Comparison of augmented samples generated by mixup-based augmentations.

cropping. However, the context-agnostic nature of these methods gives way to creating samples that are potentially harmful. Since the images are combined randomly without considering their contexts and labels, incorrect augmentation is destined to occur (Figure 1(c)). For instance, an object can be cropped out and replaced by a different kind of object from another image, or the background part of the image can be pasted on top of an existing object. Even worse, their labels are naively mixed according only to their mixing proportions, disregarding any information transfer or loss caused by the data mixing. This mismatch in data and its supervision signal yields harmful samples.

To address this problem, saliency-guided augmentation methods have been recently proposed (Walawalkar et al., 2020; Kim et al., 2020). These approaches allegedly refrain from generating harmful samples by preserving the region of maximum saliency based on the saliency maps of the image. Attentive Cutmix (Walawalkar et al., 2020) preserves the maximum saliency regions of the donor image by locating the k-most salient patches of it and merging them on top of the acceptor image. PuzzleMix (Kim et al., 2020) tries to salvage the most salient regions of each images by mixing one to another and solving an optimal transport problem and region-wise mixup ratio to maximize the saliency of the created sample. However, these precautionary measures sacrifice sample diversity - the advantage of the previous works. Unlike CutMix that teaches to attend the whole object by probabilistically choosing diverse regions of the image, maximum saliency methods lose this feature as the most discriminative region is always included in the resulting image. Moreover, they still overlook making appropriate supervision to describe the augmented image properly, and use semantically inaccurate labels determined by the mixing ratio or the size of the pasted region, which can easily mislead the network (Figure 1(b)).

To solve the drawbacks present in contemporary augmentation methods, we propose *Saliency Grafting*, a novel data augmentation method that can generate diverse but innocuous augmented data (Figure 1(d)). Instead of blindingly selecting the maximum saliency region, our method scales and thresholds the saliency map to grant all salient regions equal chance. The selected regions are then imposed with Bernoulli distribution, and are sampled to generate stochastic patches. These patches are then 'grafted' on top of another image. To compensate for the side effects of grafting such as label mismatch, we propose a novel label mixing strategy: saliency guided label mixing. By mixing the labels of the two images according to their *saliency* instead of their area, potential bad apples are effectively neutralized.

Our contribution is threefold:

- We discuss the potential weaknesses of current Mixup-based augmentation strategies and present a novel data augmentation strategy that is able to generate diverse yet meaningful data through saliency based sampling.

- We present a novel label mixing method to calibrate the generated label to match the information contained in the newly generated data.
- Through extensive experiments, we show that models trained with our method outperforms others - even under data corruption or data scarcity.

2 RELATED WORK

Data augmentation Image data augmentation played a formidable role in breakthroughs of deep learning based computer vision (LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan & Zisserman, 2015). Recently, regional dropout methods such as Cutout (DeVries & Taylor, 2017), Dropblock (Ghiasi et al., 2018) and Random Erasing (Zhong et al., 2020) were proposed to promote generalization by removing selected regions of an image or a feature map to diversify the model's focus. However, the removed regions are bound to suffer from information loss. The recently proposed Mixup (Zhang et al., 2017) and its variants (Verma et al., 2019; Guo et al., 2019), shifted the augmentation paradigm by not only transforming a sample but using a pair of samples to create a new augmented sample via convex combination. Although successful on multiple domains, Mixup is met with lost opportunities when applied to images as it cannot exploit the spatial locality. To remedy this issue, Cutmix (Yun et al., 2019), a method combining Cutout and Mixup, was proposed. By cropping out a region then filling it with a patch of another image, Cutmix executes regional dropout with less information loss. However, in Cutmix, a new problem arises as the cut-and-paste strategy incurs semantic information loss and label mismatch. To fix this issue, methods exploiting maximum saliency regions were proposed. Attentive Cutmix (Walawalkar et al., 2020) selects the top-k regions to cut and paste to another image, and Puzzlemix (Kim et al., 2020) selects maximum saliency regions of the two images and solves a transportation problem to maximize the saliency of the mixed image. However, since the maximum saliency region is always pertained, the model is robbed of the opportunities to learn from challenging but beneficial samples present in CutMix.

Saliency methods In neuroscience literature, Koch & Ullman (1987) first proposed saliency maps as a mean to understand the attention patterns of the human visual cortex. As contemporary CNNs bear close resemblance to the visual cortex, it is plausible to adapt this tool to observe the inner workings of CNNs. These saliency techniques inspired by human attention are divided into two groups: bottom-up(backward) and top-down(forward) (Katsuki & Constantinidis, 2014). For backward methods, saliency is determined in a class-discriminative fashion. Starting from the output of the network, the saliency signal is back-propagated starting from the label logit and attributed to the regions of the input image. Simonyan et al. (2013), Zhou et al. (2016), Selvaraju et al. (2017) utilizes the backpropagated gradients to construct saliency maps. Methods such as Montavon et al. (2018), Nam et al. (2020) proposed to backpropagate saliency scores with carefully designed backpropagation rules that preserves the total saliency score across a selected layer. On the other hand, forward saliency techniques start from the input layer and accumulate the detected signals up the network. The accumulated signals are then extracted at a higher convolutional layer(often the last convolutional layer) to obtain a saliency map. Unlike backward approaches, forward methods are class agnostic as the convolutional layers extract features from all possible objects inside an image to support the last classifier. These maps are used in a variety of fields such as classification (Oquab et al., 2015) and transfer learning (Zagoruyko & Komodakis, 2017).

3 PRELIMINARIES

We first clarify the notations used throughout the section by describing a general form of Mixup-based augmentation procedures. Let $f_\theta(\cdot)$ be a Convolutional Neural Network (CNN) parametrized by θ. For a given batch B of input data $\{x_1, \dots, x_m\} \in \mathcal{X}^{|B|}$ and the corresponding labels $\{y_1, \dots, y_m\} \in \mathcal{Y}^{|B|}$, a mixed image $\tilde{x}$ is generated by the augmentation function $\phi(\cdot)$ and the corresponding label $\tilde{y}$ is created through the label mixing function $\psi(\cdot)$: $\tilde{x} = \phi(x_i, x_j)$ and $\tilde{y} = \psi(y_i, y_j)$ for data index i and its random permutation j.

Then, Mixup-based augmentation methods define their own $\phi(\cdot)$ as a pixel-wise convex combination of two randomly selected pair, as follows:

$$\phi(x_i, x_j) = M_\lambda \odot h(x_i) + (\mathbf{1} - M_\lambda) \odot h(x_j)$$

Table 1: Overview of various mixed sample augmentations.

Method	**Augmentation function** ϕ	**Label mixing function** ψ
Mixup	$\lambda x_i + (1-\lambda)x_j$	$\lambda y_i + (1-\lambda)y_j$
Manifold mixup	$\lambda h(x_i) + (1-\lambda)h(x_j)$	
CutMix	$\mathbf{1}_i^{\text{Rect}} \odot x_i + (1-\mathbf{1}_i^{\text{Rect}}) \odot x_j$	
Puzzle Mix	$Z \odot \Pi_i^T x_i + (1-Z) \odot \Pi_j^T x_j$	
Saliency Grafting	$M_i \odot x_i + (1-M_i) \odot x_j$	$\lambda(S_i, S_j, M_i)y_i + \big(1-\lambda(S_i, S_j, M_i)\big)y_j$

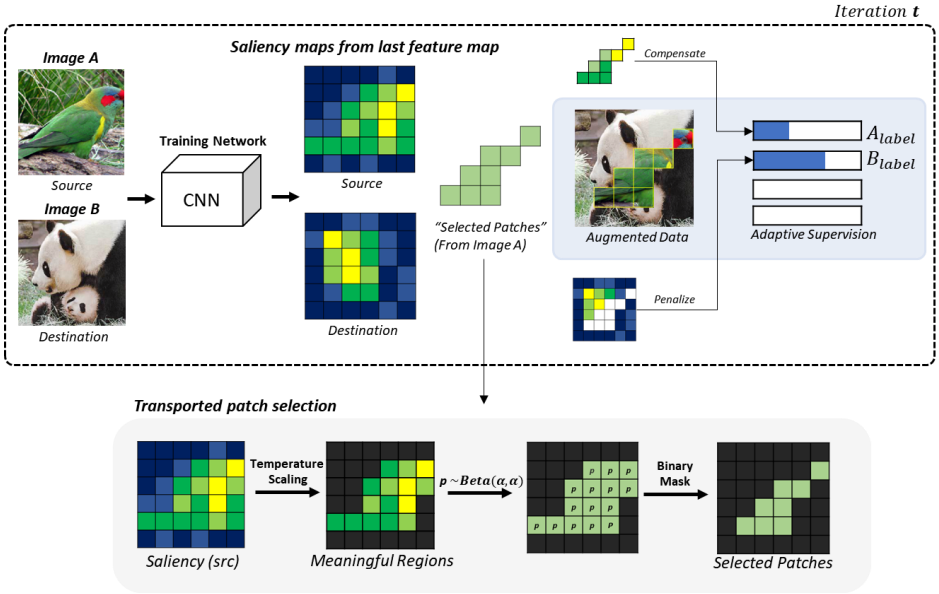


Figure 2: Overview of *Saliency Grafting*.

where M_λ is a mixing matrix controlled by a mixing ratio λ, $\odot$ is the element-wise Hadamard product, and $h(\cdot)$ is some pre-processing function.

The vanilla (input) Mixup defines the augmentation function ϕ as $\phi(x_i, x_j) = \lambda x_i + (1-\lambda)x_j$. Manifold Mixup uses similar function $\phi(x_i, x_j) = \lambda h(x_i) + (1-\lambda)h(x_j)$ but with the latent features. In CutMix, the augmentation function ϕ is defined as $\mathbf{1}_i^{\text{Rect}} \odot x_i + (1-\mathbf{1}_i^{\text{Rect}}) \odot x_j$. This method randomly cuts a rectangular region $\mathbf{1}_i^{\text{Rect}}$ from the source image x_i with area proportional to λ and pastes it onto the destination image x_j. PuzzleMix, recent saliency-based Mixup variant, employs the augmentation function $\phi(x_i, x_j) = Z^* \odot \Pi_i^{*T} x_i + (\mathbf{1} - Z^*) \odot \Pi_j^{*T} x_j$. This method exploits the image transportation plan Π and region-wise mask matrix Z to maximize the saliency of the mixed image. Note that unlike the vanilla Mixup, Z is discretized region-wise mixing matrix that satisfies $\lambda = \frac{1}{n}\sum_s \sum_t Z_{st}$ for given mixing rate λ. To find the optimal transportation plan Π^* and region-wise mask Z^* for the maximum saliency, PuzzleMix solves additional optimization problems in an alternating fashion, per *each* iteration.

Although it is a simpler scalar function, the label function $\psi(\cdot)$ is also defined in a similar form to the augmentation function $\phi(\cdot)$:

$$\psi(y_i, y_j) = \rho y_i + (1-\rho)y_j$$

where ρ is a label mixing coefficient determined by the sample pair $(x_i, y_i), (x_j, y_j)$ and the mixing ratio λ from ϕ. However, in all methods mentioned above, this ρ simply depends on λ, disregarding the contents of sample pair x_i and x_j: $\tilde{y} = \lambda y_i + (1-\lambda)y_j$. Table 1 summarizes ϕ and ψ for the augmentation methods described above.

4 SALIENCY GRAFTING

We now describe our simple approach, *Saliency Grafting*, that creates diverse and innocuous Mixup augmentation based on the content of instances being merged. Two key innovations in *Saliency Grafting* are stochastic patch selection (Section 4.1) and label mixing (Section 4.2), both of which utilize the saliency information at the core. Last but not least, another important element of *Saliency Grafting* is choosing a saliency map generation method (Section 4.3) for above two main components while keeping the learning cost to a minimum. The overall procedure is described in Figure 2. Now we discuss the details of each component in the subsequent subsections.

4.1 STOCHASTIC PATCH SELECTION WITH SALIENCY INFORMATION

The stochastic patch selection of *Saliency Grafting* aims to choose regions that can create diverse and meaningful instances. The key question here is how to select regions to be grafted, given a saliency matrix S_i for the source image x_i (whose element S_{st} indicates the saliency for a region (s, t) of x_i). As in recent studies (Kim et al., 2020; Walawalkar et al., 2020), if only regions with high intensity of S_{st} are always selected, then these regions - which are already easy to judge by the model - are continuously augmented in the iterative training procedure. As a result, the model is repeatedly exposed to the same grafting patch, which would iteratively amplify the model's attention on the selected regions and deprive the opportunity to learn how to attend to other parts and structures of the object.

In order to eliminate this *selection bias*, the patch selection of *Saliency Grafting* consists of two steps: i) softmax thresholding and ii) stochastic sampling.

Softmax thresholding To neutralize the selection bias due to the intensity of saliency, we normalize the saliency map by applying the softmax function and then binarize the map with some threshold σ:

$$S'_{st}(\boldsymbol{x};T) = \frac{\exp\left(S_{st}(\boldsymbol{x})/T\right)}{\sum_h^H \sum_w^W \exp\left(S_{hw}(\boldsymbol{x})/T\right)}, \quad S''_{st}(\boldsymbol{x};T) = \begin{cases} 1, & \text{if } S'_{st}(\boldsymbol{x};T) > \sigma \\ 0, & \text{otherwise} \end{cases}$$

given the temperature hyperparameter T to control the sharpness of the normalized saliency map. Here threshold σ also has a variety of options, but we use $\sigma = \sum_h^H \sum_w^W S'_{hw}$, using the mean value of the normalized saliency map.

Stochastic sampling Although the selection bias is significantly mitigated by thresholding, the high intensity regions are never *removed*, as the softmax function preserves the order of the regions. To address this issue, we stochastically sample the grafting regions based on the binarized saliency map produced above. The final mixing matrix is constructed by taking the Hadamard product of S''_i and a region-wise i.i.d. random Bernoulli matrix of same dimensions $P \sim \mathbf{Bern}(p_B)$: $M_i = P \odot S''_i$. Here, the batch-wise sampling probability p_B is drawn from a Beta distribution $p \sim Beta(\alpha, \alpha)$. The final augmentation function ϕ for *Saliency Grafting* is $M_i \odot x_i + (1 - M_i) \odot x_j$.

4.2 CALIBRATED LABEL MIXING BASED ON SALIENCY MAPS

In addition to the method of grafting diverse and innocuous augmentations described above, attaching an appropriate label for supervision to the generated data is also the core of *Saliency Grafting*. Although extreme, to highlight the drawbacks of the existing label mixing strategy used in all baselines, suppose that source image x_i is combined with destination image x_j, both of which have saliency concentrated in some small regions. Suppose further that this region of x_i is selected and grafted to the region where the original class of destination x_j is concentrated. Then, most of the information of class y_i is retained while most of the information on class y_j is lost. However, the label is determined in proportion to the mixing rate or the size of the area used, as all the baselines do, the generated label will be close to class y_j since most areas of it originally came from destination image x_j.

To tackle this issue, we propose a novel label mixing procedure that can adaptively mix the labels again based on saliency maps. Regarding on the destination image x_j receiving the graft, the ground truth label y_j is penalized according to the degree of occlusion. Specifically, the importance of

the destination image $I(S_j, 1 - M_i)$[1] given the mixing matrix M_i is calibrated using the saliency values of the remaining part not occluded by the source image, $I(S_j, 1 - M_i) = \frac{\|S_j \odot (\mathbf{1} - M_i)\|_1}{\|S_j\|_1}$. On the other hand, with regard to the source image x_i, the corresponding label y_i is compensated in proportion to the importance of the selected region: $I(S_i, M_i) = \frac{\|S_i \odot M_i\|_1}{\|S_i\|_1}$.

The final label mixing rate is computed based on the relative importance of x_i and x_j, so that their coefficients sum to 1 to define the calibrated label mixing function $\psi(y_i, y_j) = \lambda(S_i, S_j, M_i) y_i + \big(1 - \lambda(S_i, S_j, M_i)\big) y_j$ where $\lambda(S_i, S_j, M_i) = \frac{I(S_i, M_i)}{I(S_i, M_i) + I(S_j, 1 - M_i)}$.

4.3 Saliency map generation

Technically, *Saliency Grafting* can be combined with various saliency generation methods without the dependence on a specific method. However, the caveat here is that the performance of *Saliency Grafting* is, by design, highly affected by the quality of saliency map, or how accurately the saliency map corresponds to the ground truth label. From this point of view, the forward saliency methods, which incur less false negatives, may support Salient Grafting more stably than the backward methods (see Section 2 for forward and backward saliency methods). This is because the backward methods are likely to break down and exclude true salient regions when the model fails to predict the true label, whereas the forward methods preserve all the feature maps inside the saliency map, i.e., they act like a class-agnostic saliency detector (Mahendran & Vedaldi, 2016).

In an environment where there is no separate pre-trained model, another advantage of using forward saliency is gained: Saliency maps can be naturally constructed based on the terms already calculated in the learning process. In this environment, since the generated maps can be noisy in the early phases of training, we employ warmup epochs without no data augmentation.

We now describe the specific choice of generating the saliency maps to guide our augmentation process. We adopt the channel-collapsed absolute feature map of the network as our saliency map, mainly due to its simplicity: $S^{(l)} = \sum_{c=1}^{C} \left| A_c^{(l)} \right|$ where $A \in \mathbb{R}^{C \times H \times W}$ is the feature map at the l-th layer. Although it is possible to extract saliency maps from any designated layer in the network, we extract the maps from the last convolutional layer of the model as it generally conveys the high-level spatial information (Bengio et al., 2013).

5 Experiments

In this section, we conduct a collection of experiments to test *Saliency Grafting* against other popular augmentation methods. First, we test standard prediction performance on the standard image classification datasets. We also conduct multiple stress tests to measure the enhancement in generalization capability. Finally, we conduct an ablation study to investigate the contributions of each sub-components of *Saliency Grafting*.

5.1 Classification tasks

CIFAR-100 We evaluate our method *Saliency Grafting* on CIFAR-100 dataset (Krizhevsky et al., 2009) using two neural networks: PyramidNet-200 with widening factor $\tilde{\alpha} = 240$ (Han et al., 2017) and WRN28-10 (Zagoruyko & Komodakis, 2016). For the case of PyramidNet-200, we follow the experimental setting of Yun et al. (2019) , which trains PyramidNet-200 for 300 epochs. The results of baselines on PyramidNet-200 are as reported in Yun et al. (2019). For WRN28-10, the network is trained for 400 epochs as (Kim et al., 2020; Verma et al., 2019) reported. In this experiment, we reproduce other augmentation baselines (Zhang et al., 2017; Hendrycks et al., 2019; Yun et al., 2019; Kim et al., 2020) following the original setting of each paper. Detailed hyperparameter settings are provided in Appendix B.1.

As shown in Table 2, *Saliency Grafting* exhibits a significant improvement with PyramidNet-200 architecture compared to other augmentation baselines where *Saliency Grafting* achieves **13.59%**

[1]We use ℓ_1 norm to define the importance I in the sense that the overall saliency is simply the same as the sum of saliency in each region, but similar importance can be obtained with other norms

Table 2: Top-1/Top-5 errors on CIFAR-100 for PyramidNet-200($\tilde{\alpha} = 240$) in comparison to state-of-the-art regularization methods. The experiment was performed three times and the averaged best error rates are reported.

PyramidNet-200 ($\tilde{\alpha} = 240$) (# params: 26.8 M)	Top-1 Error (%)	Top-5 Error (%)
Baseline	16.45	3.69
Cutout	16.53	3.65
DropBlock	15.73	3.26
Mixup ($\alpha = 1.0$)	15.63	3.99
Manifold Mixup ($\alpha = 1.0$)	16.14	4.07
ShakeDrop	15.08	2.72
Cutout + Mixup ($\alpha = 1.0$)	15.46	3.42
Cutout + Manifold Mixup ($\alpha = 1.0$)	15.09	3.35
Attentive CutMix (N = 6)	15.24	3.46
CutMix	14.47	2.97
CutMix + ShakeDrop	13.81	2.29
PuzzleMix	16.52	3.70
Saliency Grafting	**13.59**	3.01
Saliency Grafting + ShakeDrop	**13.00**	2.34

Top-1 error and even outperforms the best performance obtained by combining the two regularization methods. Furthermore, when used together with Shakedrop, *Saliency Grafting* achieves additional enhancement - **13.00%** Top-1 error. In Appendix B.1, our method also shows best generalization performance - **15.24%** Top-1 error with WRN28-10. (Table 10)

ImageNet We evaluate on large-scale dataset ImageNet (Russakovsky et al., 2015) using ResNet-50 (He et al., 2016). The network is trained for 100 epochs. For a fair comparison, we follow the training protocol in Kim et al. (2020); Wong et al. (2020). Detailed experiment settings are in Appendix B.2. As shown in Table 3, *Saliency Grafting* achieves again state-of-the-art performance in both Top-1/Top-5 error rates, +0.37% higher than the best Top-1 error rate of baselines.

Additional experiments Due to the space constraint, two additional experiments are deferred to Appendix A. The first experiment shows that *Saliency Grafting* is useful for speech dataset beyond the image classification task, and the second experiment implies that the final model learned through *Saliency Grafting* contains more useful saliency information.

5.2 Stress Testing

Data scarcity The situation where data augmentation is most required is when data is scarce. In this condition, it is important to improve the generalization performance by increasing the data volume while preventing overfitting. To this end, we reduce the number of data per class to 50%, 20%, and 10%, respectively, with WRN28-10 model on the CIFAR-100 dataset. In Table 4, *Saliency Grafting* exhibits best performance in every condition from 50% to 10%. Note that the performance of CutMix deteriorates as the number of data per class decreases due to their randomness occurring label mismatching. This is in line with the fact that, as investigated by Rolnick et al. (2017), a number of data are required for performance in conditions where label corruption exists. Our method maintains the diversity of CutMix to prevent overfitting, while exploiting adaptive label mixing to reduce the mismatch between data and labels, improving the generalization performance.

Robustness against data corruption Another important stress condition is data corruption. Commercially deployed DNNs are often met with data corrupted with noise, which may be indecipherable for models trained under immaculate data. To this end, Hendrycks et al. (2019) showed that carefully crafted data augmentation can be utilized as a mean to temper the model to gain noise robustness. Although *Saliency Grafting* is not specifically built to counter data corruption, we found that

Table 3: Comparison of state-of-the-art data augmentation methods on ImageNet dataset.

ResNet-50 (# params: 25.6M)	Top-1 Error (%)	Top-5 Error (%)
Baseline	24.31	7.34
Mixup	22.99	6.48
Manifold Mixup	23.15	6.50
CutMix	22.92	6.55
AugMix	23.25	6.70
PuzzleMix	22.49	6.24
Saliency Grafting	**22.12**	**6.15**

Table 4: Top-1 error on the CIFAR-100 dataset with reduced number of data per class. The experiment was performed three times and the averaged best error rates are reported.

# of data per class	50 (10 %)	100 (20 %)	250 (50 %)
Baseline	59.36	46.28	33.28
Mixup	51.86	40.50	28.97
CutMix	55.71	42.21	28.73
PuzzleMix	52.63	41.45	28.04
Saliency Grafting	**51.31**	**39.15**	**26.97**

Table 5: Top-1 / mean Corruption error rates on CIFAR-100 and CIFAR-100-C for WRN28-10.

Method	Top-1 Error (%)	Corruption Error (%)
Baseline	22.02	50.13
AugMix	20.48	33.05
Mixup *(w/ AugMix)*	17.35	30.81
CutMix *(w/ AugMix)*	16.08	33.29
PuzzleMix *(w/ AugMix)*†	16.50	29.91
Saliency Grafting *(w/ AugMix)*	**15.74**	**29.49**

our method is able to promote noise robustness by grafting the one AugMix image into another AugMix image. Experiments on CIFAR100-C (Hendrycks & Dietterich, 2019) show that *Saliency Grafting* was able to outperform other contenders (Table 5) while being computationally efficient, requiring a *single* additional forward pass, compared to Hendrycks et al. (2019) requiring 2 additional forward passes and Kim et al. (2020) requiring one additional backward pass and solving a separate optimization problem.

5.3 Ablation Study

Stochastic selection VS deterministic selection In Section 4.1, we argued that the deterministic patch selection process of existing methods (Walawalkar et al., 2020; Kim et al., 2020) leads to performance degradation. Here, we measure the classification accuracy on CIFAR-100 with PyramidNet-200 where the deterministic top-k selection of Walawalkar et al. (2020) is replaced by our stochastic selection. For fair comparison, the softmax temperature T is adjusted to satisfy $\mathbb{E}_i[\sum_s \sum_t M_{i,st}] = k$. Results show that stochastic selection indeed outperforms deterministic selection (Table 6). The training and validation loss curves (Appendix C) show that stochastic selection resists overfitting, showing training curves similar to that of Dropout (Srivastava et al., 2014) (higher training loss, lower validation loss).

Label mixing strategies In Section 4.2, we discussed the pitfalls of naive area-based label mixing and proposed saliency-based label mixing as a solution. Here, we compare the two strategies. We experiment on CIFAR-100 with PyramidNet-200 and replace the mixing strategy of *Saliency Grafting* with area-based mixing. Results in Table 6 confirms that saliency-based mixing outperforms area-based mixing.

Forward saliency VS Backward saliency To support our choice of forward saliency maps (Section 4.3), we conduct an additional experiment on CIFAR-100 With WRN28-10 where the forward saliency map of *Saliency Grafting* is replaced by CAM(Zhou et al., 2016), a backward saliency map. The detailed settings are kept identical to Section 5.1. Results show that the classification error increases when a backward saliency map is used (Table 7).

Table 6: Top-1/Top-5 errors on CIFAR-100 for PyramidNet-200($\tilde{\alpha} = 240$).

Method	Top-1 Error (%)	Top-5 Error (%)
Deterministic + area labels	14.30	2.87
Stochastic + area labels	14.03	2.74
Stochastic + saliency labels	**13.59**	**2.34**

Table 7: Top-1/Top-5 errors on CIFAR-100 for WRN28-10.

Method	Top-1 Error (%)	Top-5 Error (%)
Backward (CAM)	15.70	3.8
Forward (ours)	**15.24**	**3.73**

6 Conclusion

We have presented *Saliency Grafting*, a data agumentation method that generates diverse saliency-guided samples via stochastic sampling and neutralizing any induced data-label mismatch with saliency-based label mixing. Through extensive experiments, we have shown that models equipped with *Saliency Grafting* outperforms existing mixup-based data augmentation techniques under both normal and extreme conditions while using less computational resources.

REFERENCES

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pp. 173–182, 2016.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964. IEEE, 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 10727–10737, 2018.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019.

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

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 and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019.

Fumi Katsuki and Christos Constantinidis. Bottom-up and top-down attention: different processes and overlapping neural systems. *The Neuroscientist*, 20(5):509–521, 2014.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning (ICML)*, 2020.

Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pp. 115–141. Springer, 1987.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. *AAAI*, 2020.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 685–694, 2015.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3642–3646. IEEE, 2020.

Pete Warden. Launching the speech commands dataset. https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html, 2017. Accessed: 2010-09-30.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

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 International Conference on Computer Vision*, pp. 6023–6032, 2019.

S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A ADDITIONAL EXPERIMENTS

Speech data To test our method on data outside the distribution of natural images, we use the Google Speech Commands dataset (Warden, 2017). The training samples were first augmented in the time domain by applying random changes in amplitude, speed and pitch and in the frequency domain by stretching and time-shifting the spectrogram. Then, random background noises clip drawn from the noise compilation in the dataset were added to the samples. Finally, the samples are transformed into 32x32 mel-spectrograms by using 32 MFCC filters. To evaluate our method on this data, we used the WRN28-10 architecture. As in Table 8, our method was able to outperform other methods in a non-natural image domain.

Table 8: Top-1 error on Google Speech Commands in comparison to other augmentation methods.

WRN28-10 (# params: 36.5 M)	Top-1 Error (%)
Baseline	2.81
Mixup	2.72
CutMix	2.62
Saliency Grafting	**2.51**

Weakly supervised object localization To examine how our method affects the *backward* saliency of a model (how a model 'thinks'), we measure the weakly supervised object localization performance on the CUB200-2011 dataset (Wah et al., 2011). For ResNet-50, we slightly modify the last convolution layer to make featuremap size from 7x7 to 14x14. We first obtain the backward saliency map with CAM (Zhou et al., 2016). The map is then thresholded using a 15% of the maximum value of CAM and enclosed by the smallest possible bounding box. We measure the Intersection-over-Union(IoU) between this estimated bounding box and the ground truth bounding box. For localization accuracy, IoU between the estimated bounding box and ground truth box is greater than 0.5, and, simultaneously, the predicted class label should be correct. We used Adam optimizer, and the initial learning rate, weight decay, batch size were 0.001,0.0001, and 32. The learning rate is decaying by the factor of 0.1 per 150 epochs. All the experiments was performed three times and the averaged best error rates are reported.

Table 9: Performance of weakly supervised obeject localization on the CUB200-2011 dataset.

Method	Loc Acc(%)
ResNet-50 + CAM	26.80
ResNet-50 + Mixup	35.86
ResNet-50 + CutMix	26.81
ResNet-50 + PuzzleMix	34.98
ResNet-50 + Saliency Grafting	**37.28**

B DETAILED EXPERIMENTAL SETTINGS

B.1 CIFAR-100 CLASSIFICATION

We use stochastic gradient descent (SGD) with momentum 0.9 for both network models. Mixing ratio λ is sampled from *Beta*(1,1) with regard to Mixup and CutMix. In the Manifold Mixup case, we adopt *Beta*(2,2) for sampling distribution to follow the original paper. PuzzleMix has four hyper-parameters: label smoothness term β, data smoothness term γ, prior term η, and transport cost ξ. We use $(\beta, \gamma, \eta, \xi) = (1.2, 0.5, 0.2, 0.8)$. For the classification task, our method use a temperature T 0.1 and *Beta*(2,2) for stochastic sampling. For early convergence, we warmup the model for 5 epochs. The weight decay of each augmentation method is different with CIFAR dataset, so the results of our paper are reported as having better results among 0.0005 and 0.0001. For PyramidNet-200 network, the initial learning rate was set to 0.25 and decayed by the factor of 0.1 at 100 and 200 epoch. For WRN28-10 network, the initial learning rate was set to 0.1 and decayed by the factor of 0.1

per 75 epochs. All the experiments was performed three times and the averaged best error rates are reported.

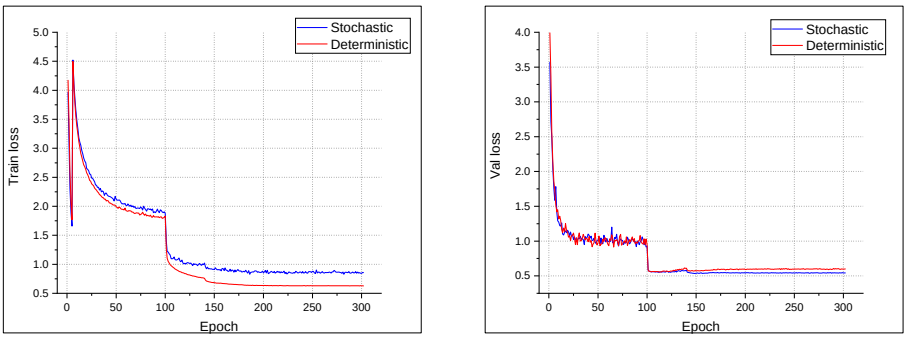
B.2 IMAGENET CLASSIFICATION

For ImageNet, we follow the training process in Kim et al. (2020). We trained the ResNet-50 model with an image resized to 224 by 224. Our proposed method does not require the tricky tuning of the learning rate decay, but we followed the cyclic learning rate decay setting for a fair comparison. Our method maintains temperature T for 0.1 and sampling α for 2 since it does not require complicated hyperparameter tuning for each dataset. We use SGD optimizer, and the initial learning rate, momentum, weight deacy, and batch size were 0.1, 0.9, 0.0001, 256. For ImageNet dataset, we warmup the model only for 1 epoch.

C ADDITIONAL TABLES AND FIGURES

Table 10: Top-1/Top-5 errors on CIFAR-100 for WRN28-10 in comparison to state-of-the-art regularization methods. The experiment was performed three times and the averaged best error rates are reported. † indicates the reported result in the original paper.

WRN28-10 (# params: 36.5 M)	Top-1 Error (%)	Top-5 Error (%)
Baseline	22.02	6.18
Mixup ($\alpha = 1.0$)	18.04	5.17
Manifold Mixup† ($\alpha = 1.0$)	18.04	-
CutMix	17.51	5.16
AugMix	20.48	5.74
PuzzleMix†	15.95	3.92
PuzzleMix(half)†	16.23	3.90
Saliency Grafting	**15.24**	**3.73**



(a) Train loss (b) Validation loss

Figure 3: Training and validation loss curves of Attentive Cutmix equipped with deterministic and stochastic sampling.

D EXAMPLES

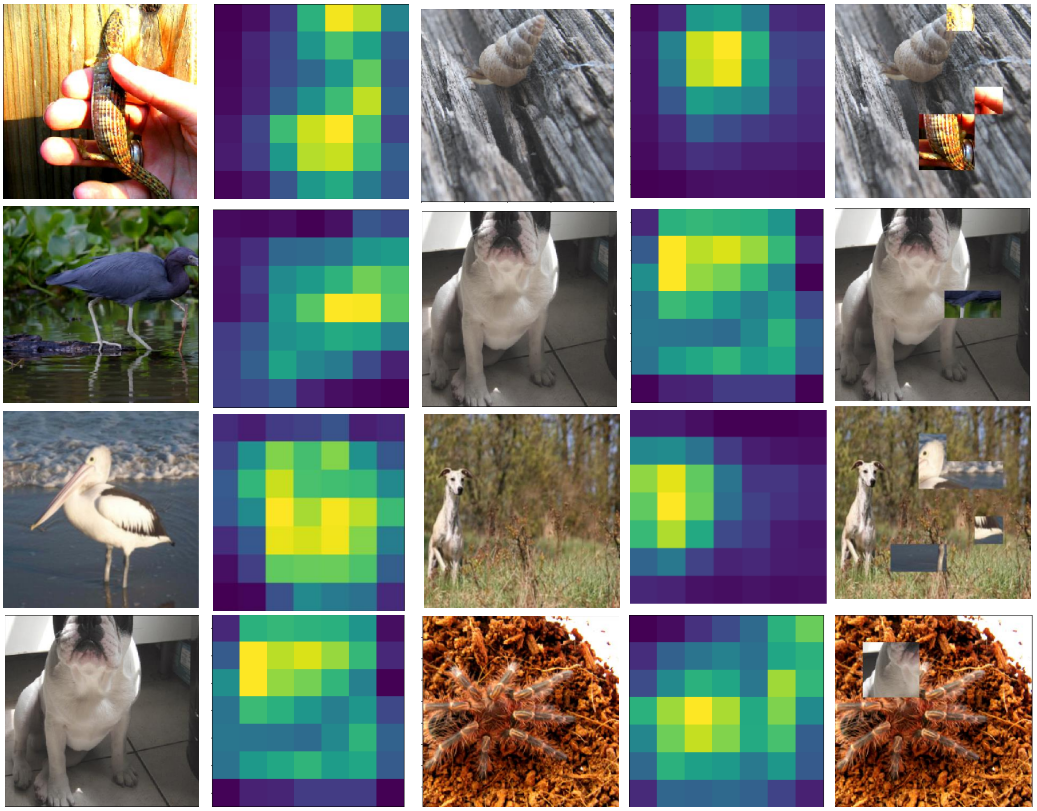


Figure 4: *Saliency Grafting* images.