SAGE: Saliency-Guided Mixup with Optimal Rearrangements

Avery Ma[1,2] Nikita Dvornik[3] Ran Zhang[3] Leila Pishdad[4] Konstantinos G. Derpanis[2,3,5] Afsaneh Fazly[3]

[1]University of Toronto [2]Vector Institute [3]Samsung AI Centre Toronto [4]Borealis AI [5]York University

Motivation

Data augmentation is a key element for training accurate models by reducing overfitting and improving generalization.

- Conventional data augmentation techniques (photometric and geometric transformations) merely create slightly altered copies of the original images and thus introduce limited diversity in the augmented dataset.
- More advanced data augmentation combines multiple training examples into a new image-label pair, leading to increased diversity of the augmented set.
- Nonetheless, these approaches are agnostic to image semantics; they ignore object location cues, and as a result may produce ambiguous scenes with occluded distinctive regions.

To account for such shortcomings, can we **explicitly use visual saliency** for data augmentation?

Figure 1. **Comparison of data augmentation methods.** Thanks to the saliency-guided mixing and image rearrangements, SAGE produces more meaningful and informative scenes, as verified in our experiments.

SAGE Overview

The main idea behind SAGE is to synthesize novel images (with their labels) by blending pairs of training samples, using spatial saliency information as guidance for optimal blending.

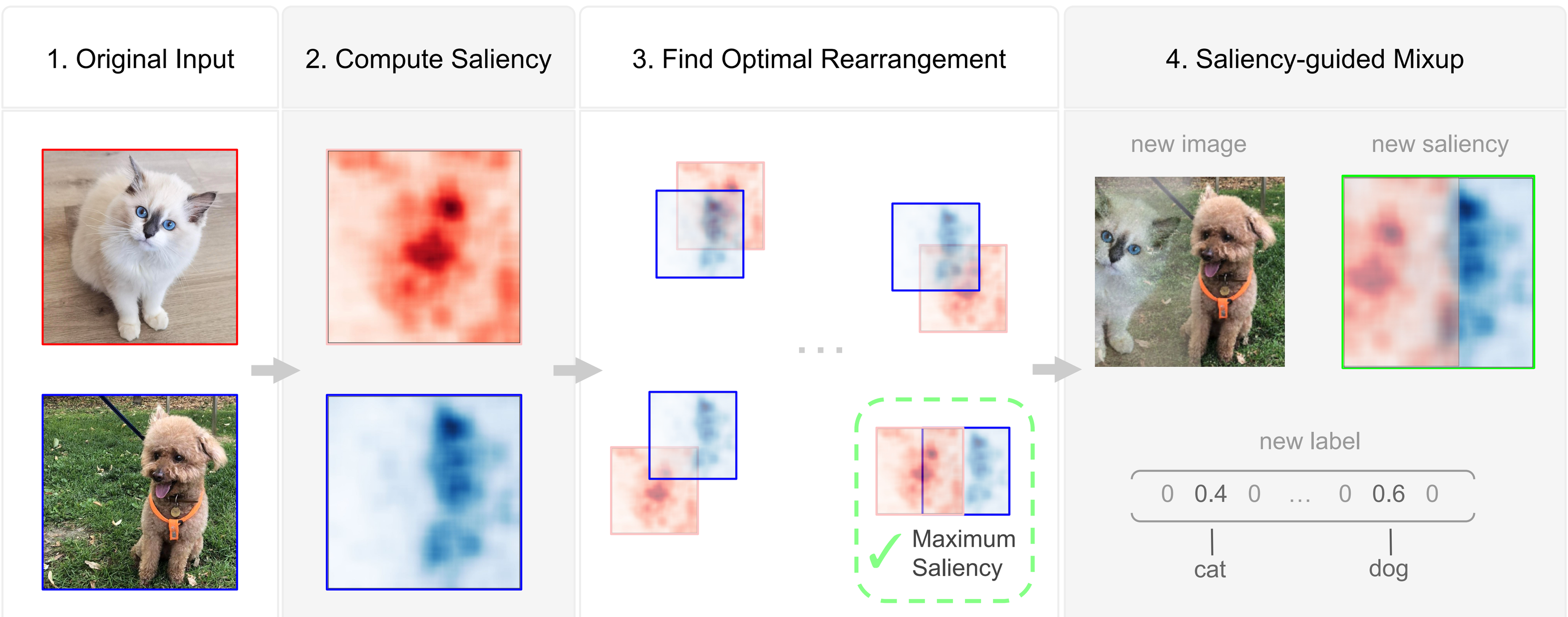


Figure 2. **SAGE overview.** Our method consists of three independent components: **i)** computing the saliency maps given the original images, **ii)** finding the best rearrangement of the images that maximizes the total saliency (in the green box), and **iii)** fusing the overlapping image parts and deriving the new label based on Saliency-guided Mixup. As a result, SAGE produces smooth, realistic and informative scenes.

Computing Saliency Maps

We define the saliency of each image pixel as its importance in making the correct prediction, using a given vision model. More formally, we are given

- a training sample, (x, y), where $x \in \mathbb{R}^{d \times d \times 3}$ is an RGB image and $y \in \mathbb{R}^C$ is the corresponding one-hot label vector, and
- a classifier, $f_\theta(\cdot)$, that is the current partially trained model, and our task loss, $\ell(f_\theta(x), y)$, measuring the discrepancy between the classifier's output and the true label.

We define the saliency, $s \in \mathbb{R}^{d \times d}$, as the magnitude of the gradient with respect to the input image,

$$s(x) = \left|\nabla_x \ell(f_\theta(x), y)\right|_{l_{2,3}}, \qquad (1)$$

where $|\cdot|_{l_{2,3}}$ denotes the $l2$-norm along the third (color) dimension. In practice, the saliency map tends to focus on the foreground objects useful for classification and ignores irrelevant background.

Saliency-guided Mixup

We propose Saliency-guided Mixup: given two images, x_0 and x_1, and their saliency maps, s_0 and s_1, we craft a 2D mixing mask, $M \in \mathbb{R}^{d \times d}$, and use it to mix the images:

$$x' = M \odot x_0 + (1 - M) \odot x_1, \;;\; M = \frac{\tilde{s}_0}{\tilde{s}_0 + \tilde{s}_1 + \zeta}, \qquad (2)$$

where $x' \in \mathbb{R}^{d \times d \times 3}$, $\tilde{s}_0$ and $\tilde{s}_1$ are spatially-normalized and Gaussian-smoothed saliency maps, ζ is a scalar hyperparameter used to avoid division-by-zero and $\odot$ denotes element-wise product.

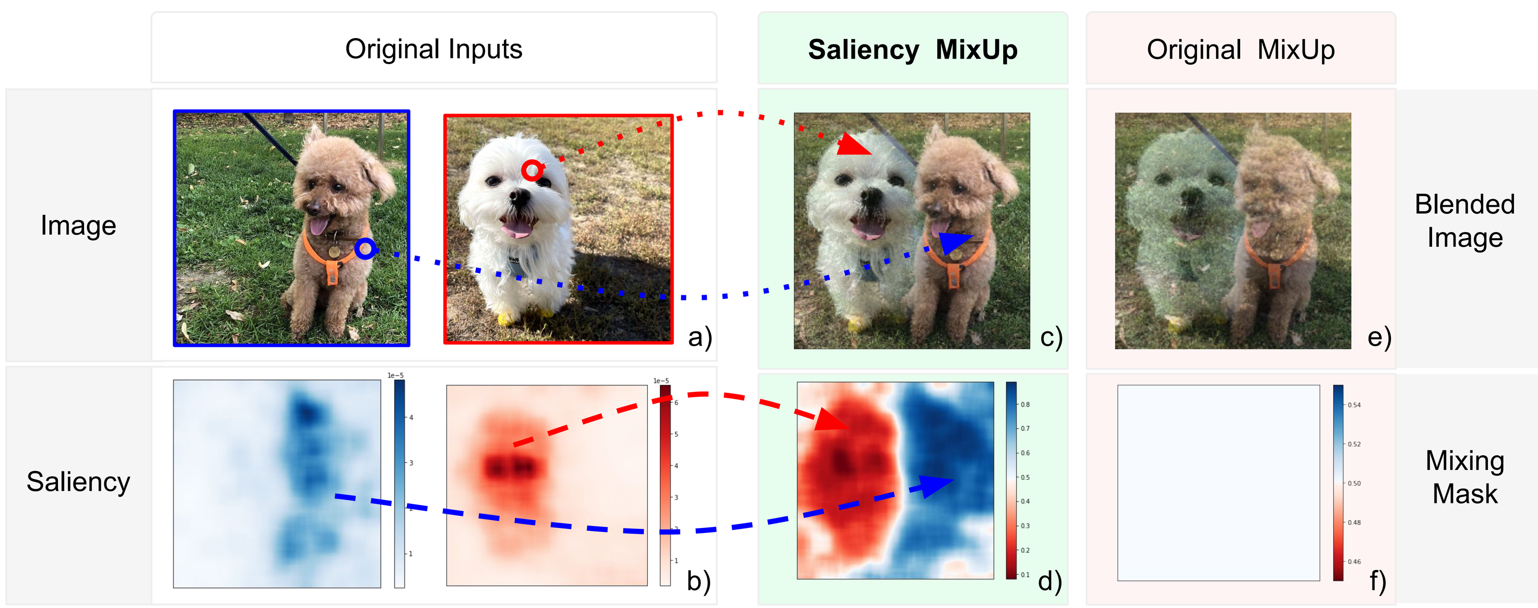


Figure 3. **Comparison between Saliency-guided Mixup and original Mixup.** Given the a) original images with b) saliency maps, our Saliency Mixup computes d) the Mixing Mask M (given by Eq. 2) based on the relative saliency of the inputs. The values of M are represented with a heatmap; blue areas indicate stronger contribution of image 1, red areas correspond to image 2 being more prominent and pale areas indicate more uniform blending. Consequently, salient regions from different images contribute to different locations and result in a realistic, informative output c). In contrast, the original Mixup produces f) a uniform mixing mask (at $\lambda = 0.5$), which results in e) an unrealistic and unclear image.

Optimal Rearrangements via Saliency Maximization

- **Issue:** When the maximally salient regions in both images spatially overlap, the mask, M, tends to suppress one or both objects, which leads to uninformative new scenes.
- **Solution:** Shift one image relative to the other prior to mixing.

Consider a translation operator that shifts a tensor z by τ pixels as $\mathcal{T}(z, \tau)$. To quantify how successful a given rearrangement is in resolving the saliency overlap, we measure the total saliency ($v(\tau) \in \mathbb{R}$) after the rearrangement:

$$v(\tau) = \sum_{i,j} \left[M^\tau \odot \tilde{s}_0 + (1 - M^\tau) \odot \mathcal{T}(\tilde{s}_1, \tau) \right], \qquad (3)$$

where $\mathcal{T}(\tilde{s}_1, \tau)$ is the saliency $\tilde{s}_1$ translated by τ and M^τ is the mixing mask (Eq. 2) computed with $\tilde{s}_0$ and $\mathcal{T}(\tilde{s}_1, \tau)$. Finally, we find the optimal rearrangement, τ^*, by solving $\tau^* = \operatorname{argmax}_{\tau \in \mathcal{O}} v(\tau)$, where $\mathcal{O}$ is the space of all possible offsets.

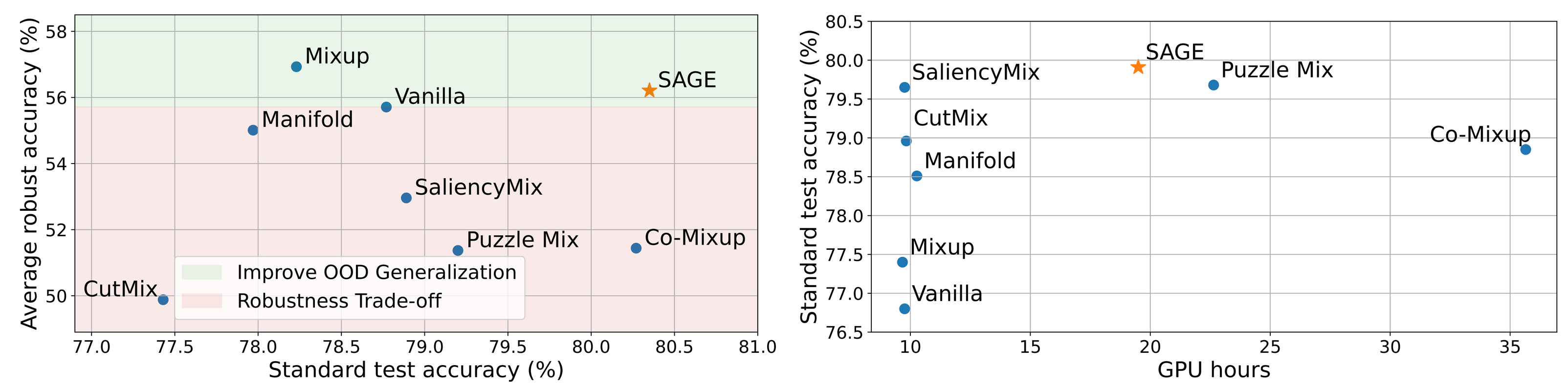
a. Total saliency $v(\tau_1) = 0.48$ b. Total saliency $v(\tau_2) = 0.57$ c. Max saliency $v(\tau^*) = 0.72$

Figure 4. **Possible rearrangements.** In each example, the saliency map corresponding to the rearrangement is shown on the left, the corresponding image is on the right. The rearrangement maximizing the total saliency is shown in c).

Results

Dataset	Model	Vanilla	Mixup	CutMix	Manifold	SaliencyMix	Puzzle Mix	Co-Mixup	SAGE
CIFAR-10	PreActResNet18	95.07	95.97	96.27	96.28	96.15	96.62	96.23	**96.95**
CIFAR-100	PreActResNet18	76.80	77.40	78.96	78.51	78.85	79.65	79.68	**79.91**
CIFAR-100	WRN16	78.55	79.83	80.03	79.77	80.16	**80.73**	80.42	80.45
CIFAR-100	ResNext29	78.77	78.23	77.43	77.97	78.89	79.20	80.27	**80.35**

Table 1. **Image classification accuracy.** CIFAR-10 and CIFAR-100 results are obtained by averaging over three independent training runs.



a. Robustness Comparison b. Runtime Comparison

Figure 5. **Robustness and efficiency analysis of SAGE.** (a) Robustness vs. standard accuracy in OOD generalization. The methods in the green area improve both accuracy and robustness relative to vanilla augmentation, while the others in red improve standard test accuracy at the cost of decreased robustness. (b) Runtime comparison of SAGE and other baselines. For SAGE, there is no noticeable overhead besides the additional forward and backward pass to compute the saliency map which approximately doubles the time of Vanilla training.