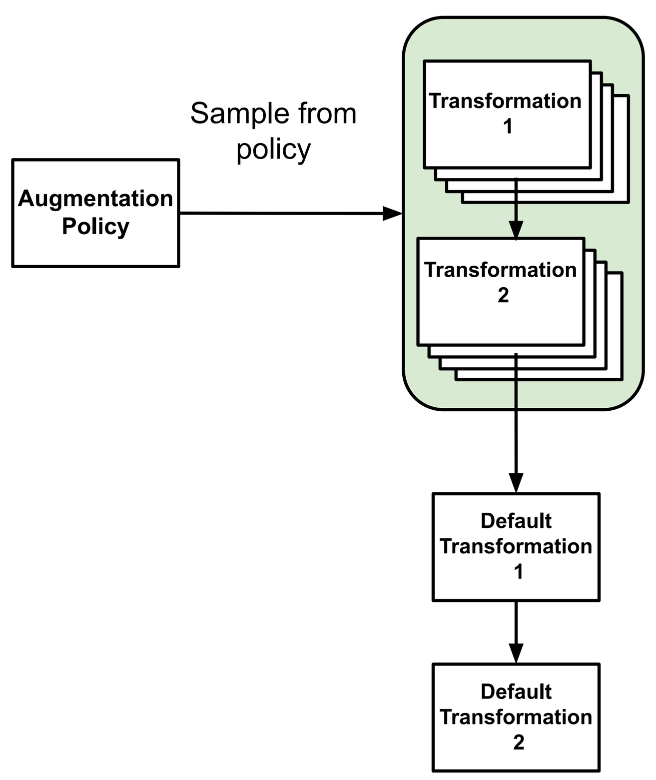
Deep AutoAugment

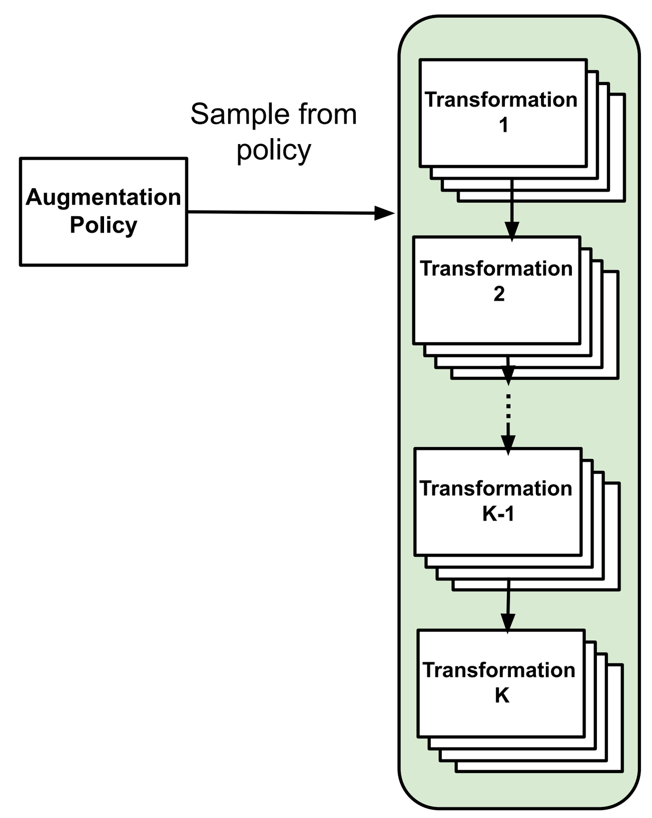
Yu Zheng[1], Zhi Zhang[2], Shen Yan[1], Mi Zhang[1]
Michigan State University[1], Amazon Web Services[2]

Overview

- Existing automated data augmentation approaches requires hand-picked default transformations (e.g. flip -> cutout -> crop), and need to manually determine the depth of augmentation.
- We propose ***Deep AutoAugment (DeepAA)***, a fully automated data augmentation search method that finds a multi-layer data augmentation policy from scratch.

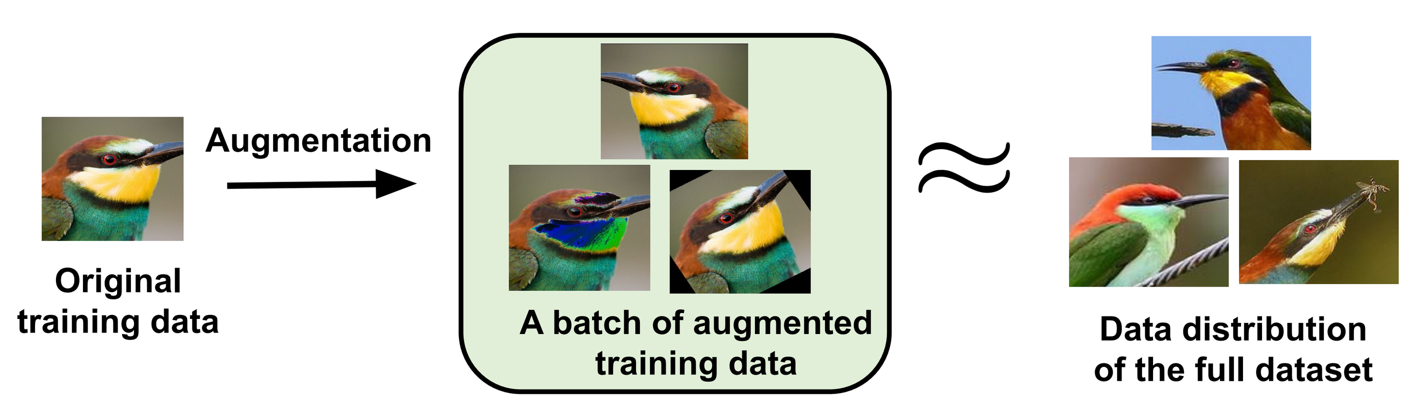


Existing Approach



Our Approach (DeepAA)

Challenge#1: What training signal should we use?

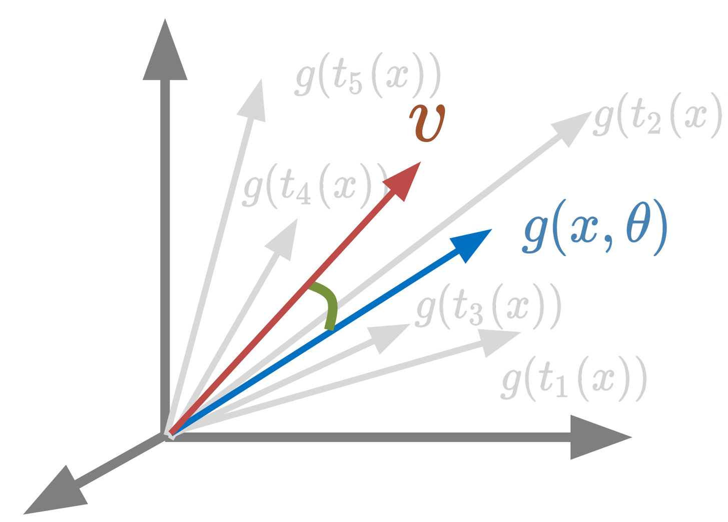


As the distribution of augmented data gets closer to the ture data distribution, the direction of gradient of the augmented data should match the gradient of the validation batch sampled form the true data distribution. We hence optimize the cosine similarity between them.

arXiv

GitHub

Code available:
https://github.com/MSU-MLSys-Lab/DeepAA



- x denotes a training data point sampled from the dataset
- t_n denotes an augmentation transformation from the candidate set $\{t_1, t_2, \cdots, t_N\}$
- $g(t_n(x))$ denotes the gradient of sample x augmented with transformation t_n .
- $p_\theta(n)$ denotes the probability of transformation t_n , which serves as the **augmentation policy**.
- v denotes the gradient of the validation batch sampled from the true data distribution.

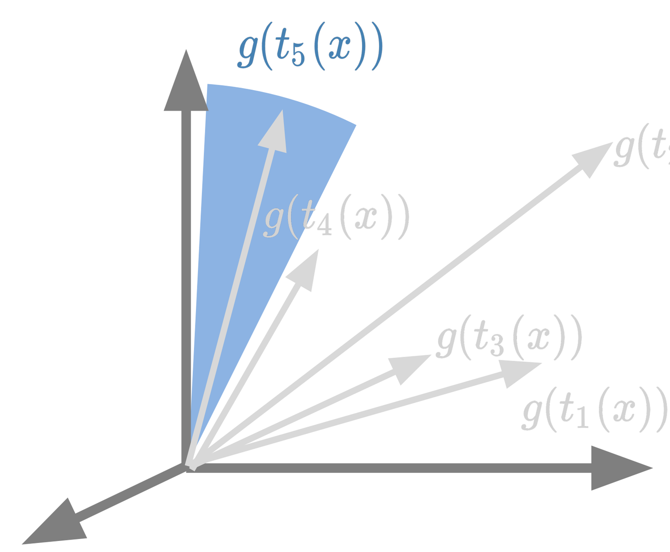
The average gradient of augmented training data with transformations $\{t_1, t_2, \cdots, t_N\}$, and policy $\{p_\theta(1), p_\theta(2), \cdots, p_\theta(N)\}$.

$$g(x;\theta) = \sum_{n=1}^{N} p_\theta(n) g(t_n(x))$$

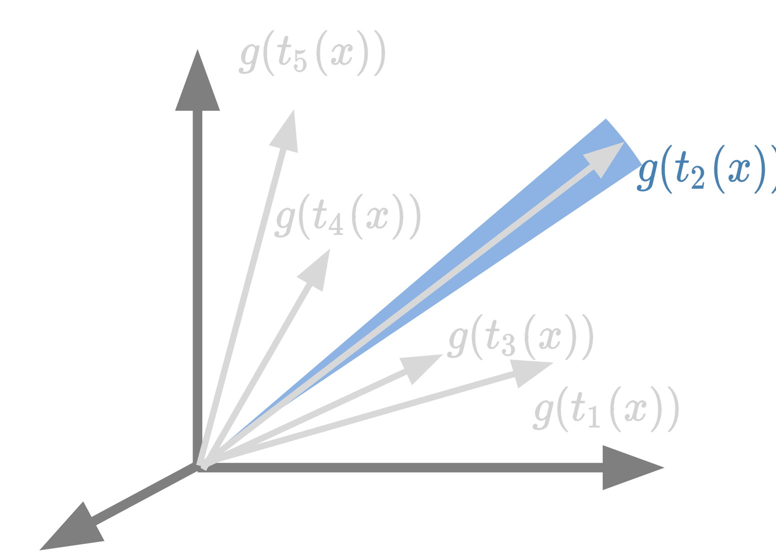
The gradient matching objective:

$$\theta = \arg\max_\theta \ \text{cosineSimilarity}(v, g(x;\theta))$$
$$= \arg\max_\theta \frac{v^T \cdot g(x;\theta)}{\|v\| \cdot \|g(x;\theta)\|}$$

We regularize the gradient matching by penalizing the transformation with high variance:

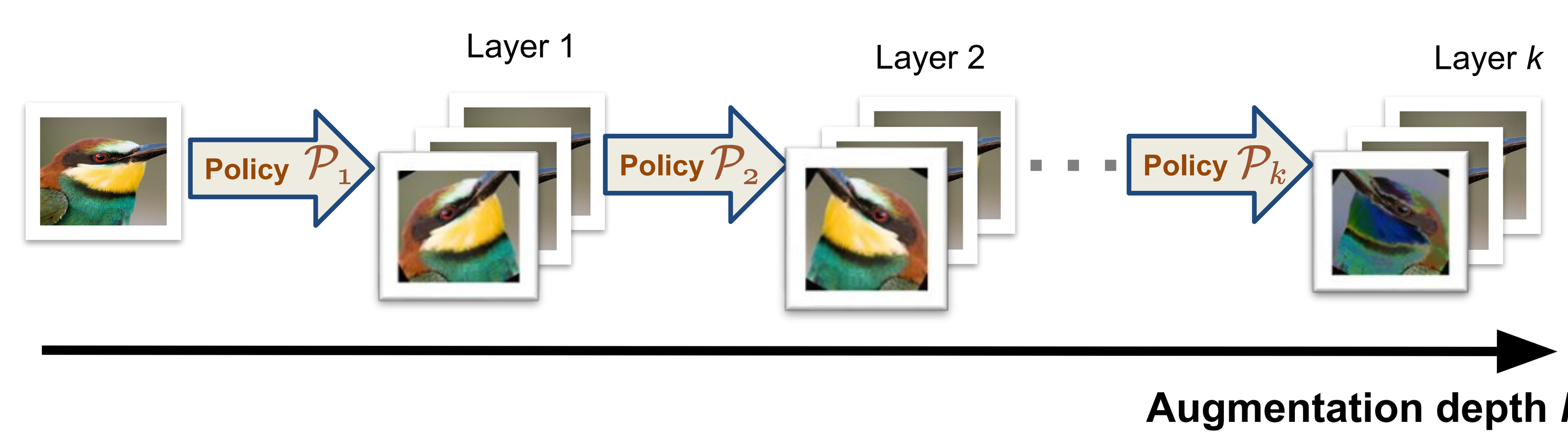


If transformation t_5 exhibits high variance for different x, we decrease the corresponding probability $p_\theta(5)$.



If transformation t_2 exhibits low variance for different x, we increase the corresponding probability $p_\theta(2)$.

Challenge#2: How to address the exponential growth of the search space?



The policy $\mathcal{P}_k$ implicitly depends on the policy of the previous *k-1* layer, i.e., $\mathcal{P}_k = p_{\theta_k}(n|\mathcal{P}_1, \cdots, \mathcal{P}_{k-1})$ while the dimension of policy at layer *k* still remains constant *N*.

Experiment results

	Baseline	AA	PBA	FastAA	FasterAA	DADA	RA	UA	TA(RA)	TA(Wide)	DeepAA
CIFAR-10											
WRN-28-10	96.1	97.4	97.4	97.3	97.4	97.3	97.3	97.33	97.46	97.46	**97.56** $\pm$ 0.14
Shake-Shake (26 2x96d)	97.1	98.0	98.0	98.0	98.0	98.0	98.0	98.1	98.05	98.21	**98.11** $\pm$ 0.12
CIFAR-100											
WRN-28-10	81.2	82.9	83.3	82.7	82.7	82.5	83.3	82.82	83.54	84.33	**84.02** $\pm$ 0.18
Shake-Shake (26 2x96d)	82.9	85.7	84.7	85.1	85.0	84.7	-	-	-	86.19	**85.19** $\pm$ 0.28

Table 1: Top-1 test accuracy on CIFAR-10/100 for Wide-ResNet-28-10 and Shake-Shake-2x96d. The results of DeepAA are averaged over four independent runs with different initializations. The 95% confidence interval is denoted by $\pm$.

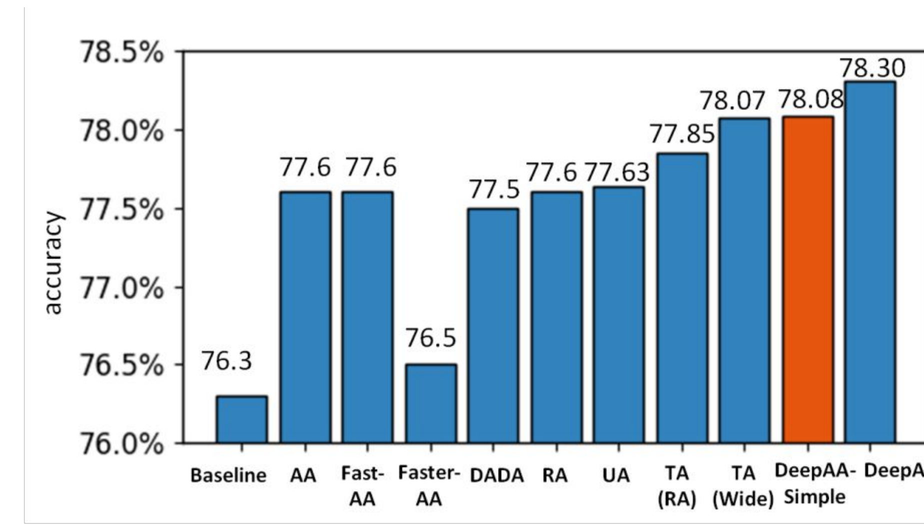
	Baseline	AA	Fast AA	Faster AA	DADA	RA	UA	TA(RA)	TA(Wide)	DeepAA
ResNet-50	76.3	77.6	77.6	76.5	77.5	77.6	77.63	77.85	78.07	**78.30** $\pm$ 0.14
ResNet-200	78.5	80.0	80.6	-	-	-	80.4	-	-	**81.32** $\pm$ 0.17

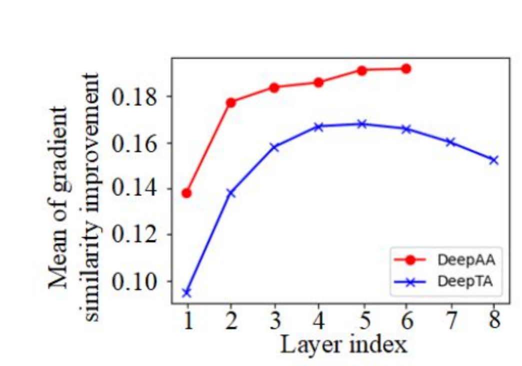
Table 2: Top-1 test accuracy (%) on ImageNet for ResNet-50 and ResNet-200. The results of DeepAA are averaged over four independent runs with different initializations. The 95% confidence interval is denoted by $\pm$.

We conduct a search with only a single layer of augmentation. When evaluating the searched policy, we apply the default augmentation in addition to the searched policy. We refer to this variant as DeepAA-Simple.

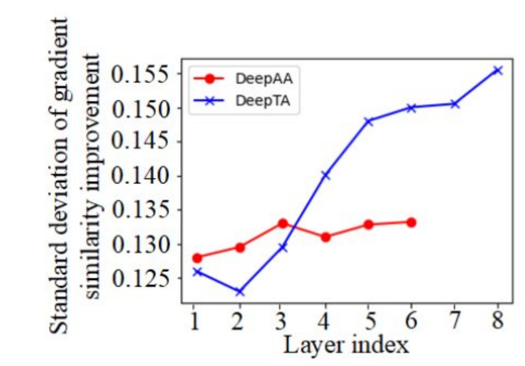
Two Key Observations:

- Even with a single searched augmentation layer, DeepAA-Simple still outperforms other methods.
- DeepAA with fully automated policy shows a 0.22% performance gain over DeepAA-Simple.

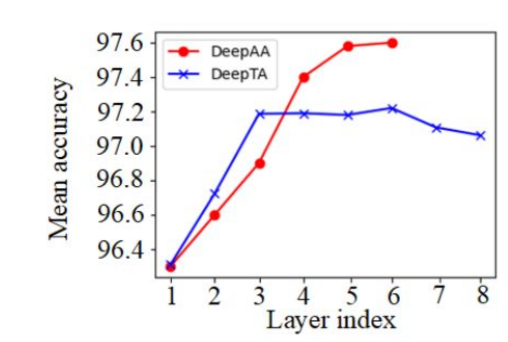




(a) Mean of the gradient similarity improvement

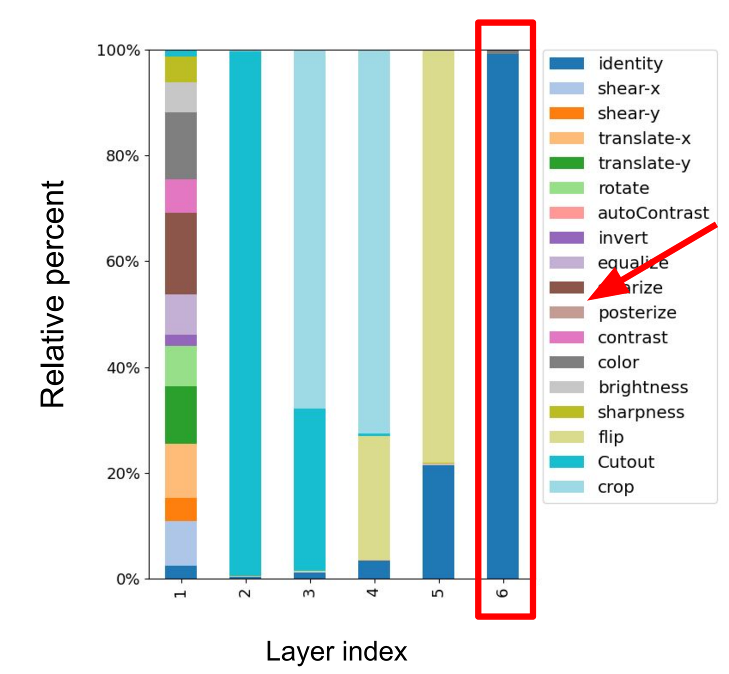


(b) Standard deviation of the gradient similarity improvement

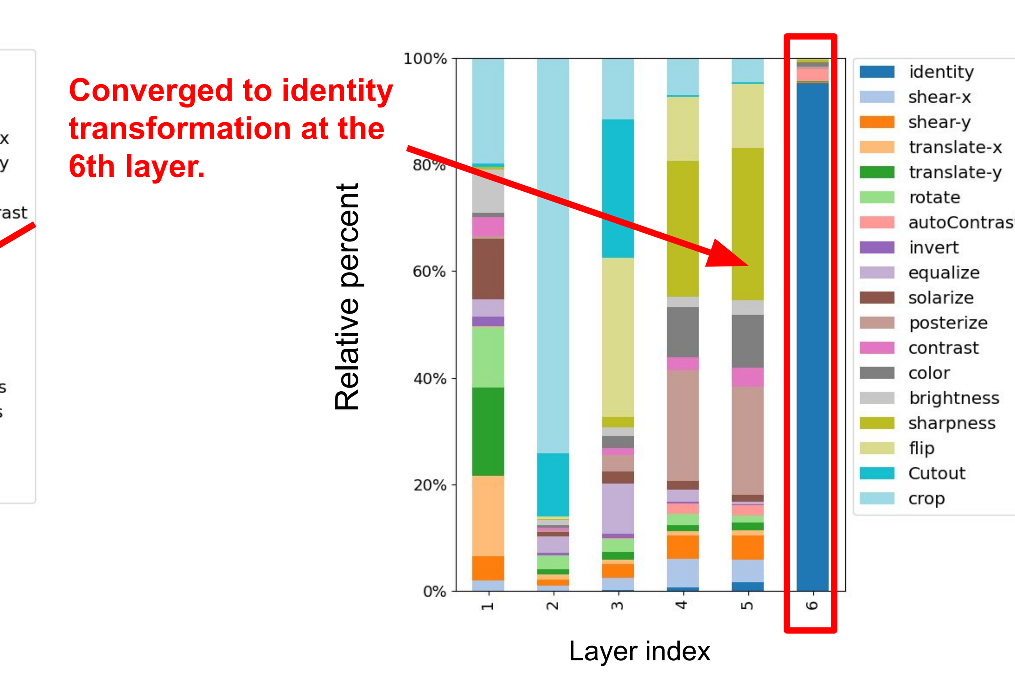


(c) Mean accuracy over different augmentation depth

- We design the baseline, DeepTA, by stacking multiple layers of TrivialAugment (TA).
- In comparison, DeepAA exhibits 1) higher cosine similarity, 2) lower variance, 3) higher accuracy.



(a) Operation distribution at each layer for **CIFAR-10/100**



(b) Operation distribution at each layer for **ImageNet**