Deep AutoAugment

Yu Zheng[1], Zhi Zhang[2], Shen Yan[1], Mi Zhang[1]

Michigan State University[1], Amazon Web Services[2]

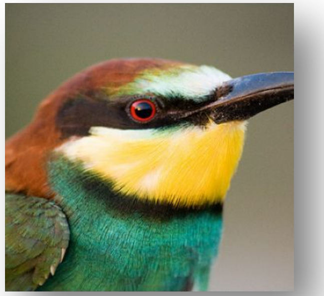
https://arxiv.org/abs/2203.06172

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

MICHIGAN STATE UNIVERSITY

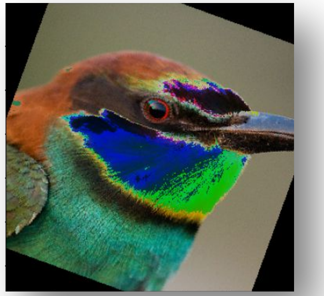
Background of Data Augmentation

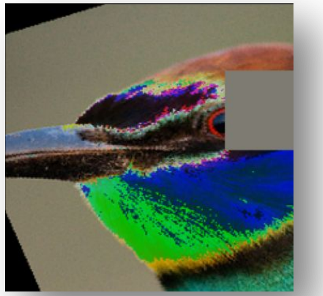
Original Image



Data Augmentation

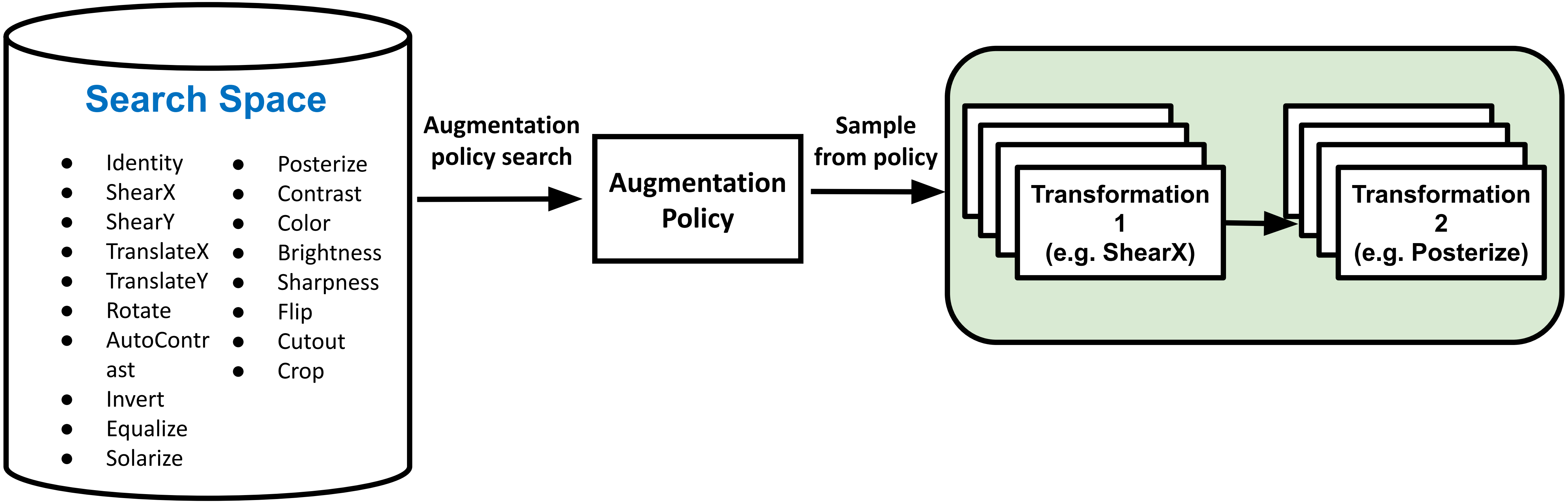
Augmented Images



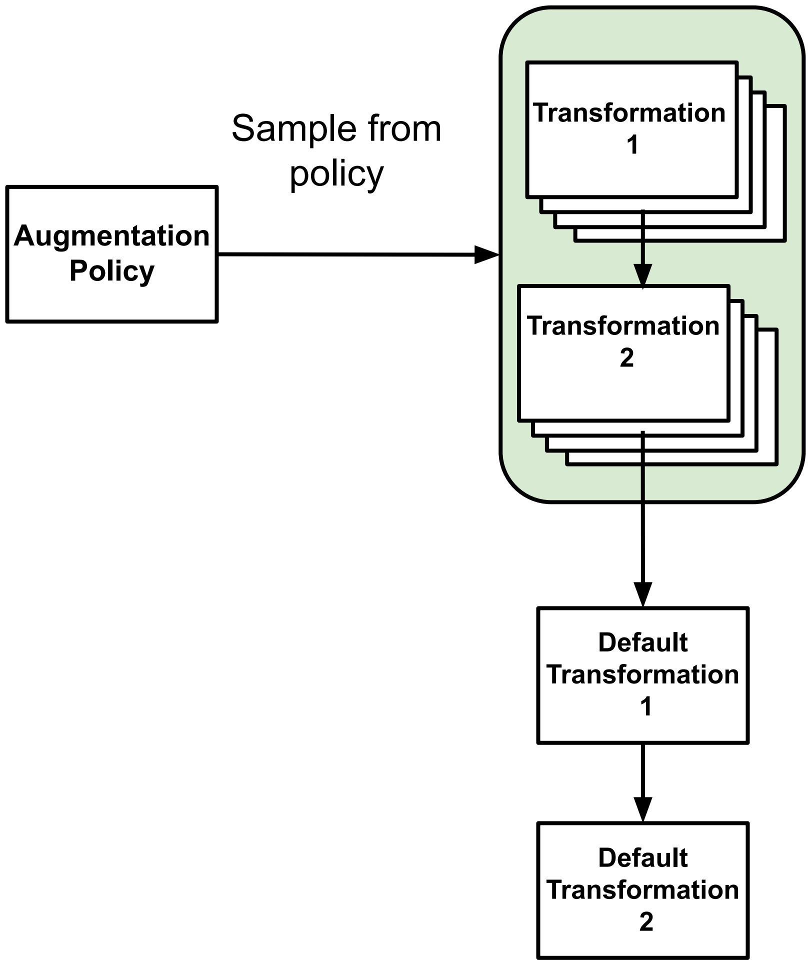


Data augmentation (DA) is a powerful technique to improve the performance of machine learning models since it effectively regularizes the model by increasing the number and the diversity of data points.

Overview of Automated Data Augmentation Pipeline



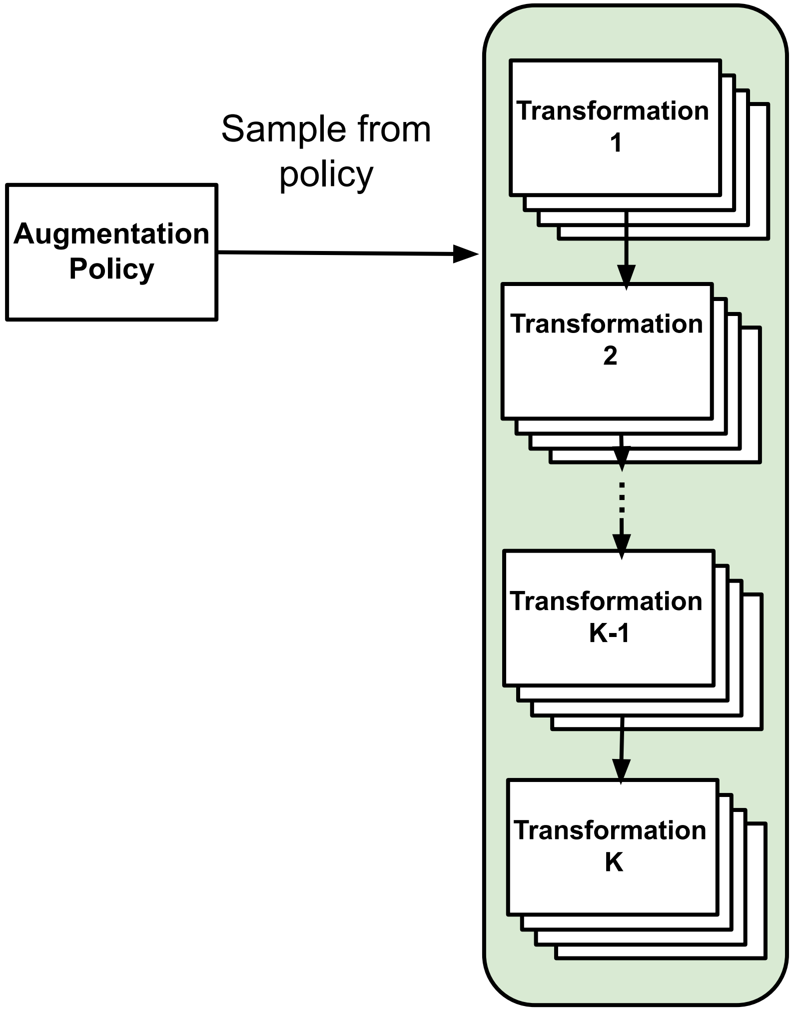
Existing Automated Data Augmentation Methods



Limitations

- Require **hand-picked default transformations**, e.g. flip -> cutout -> crop.
- Need to **manually** determine the depth of augmentation.

Deep AutoAugment (DeepAA): A Fully Automated Approach



Strengths

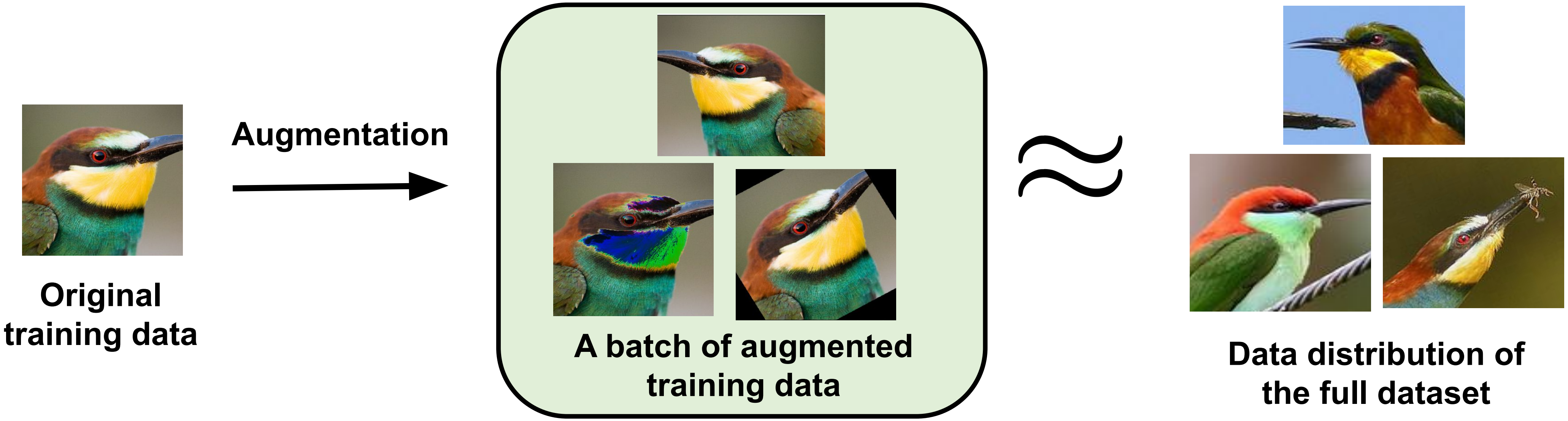
- No hand-picked default transformations.
- Automatically determine the depth of augmentation.

Two Key Challenges

Challenge#1: What training signal should we use?

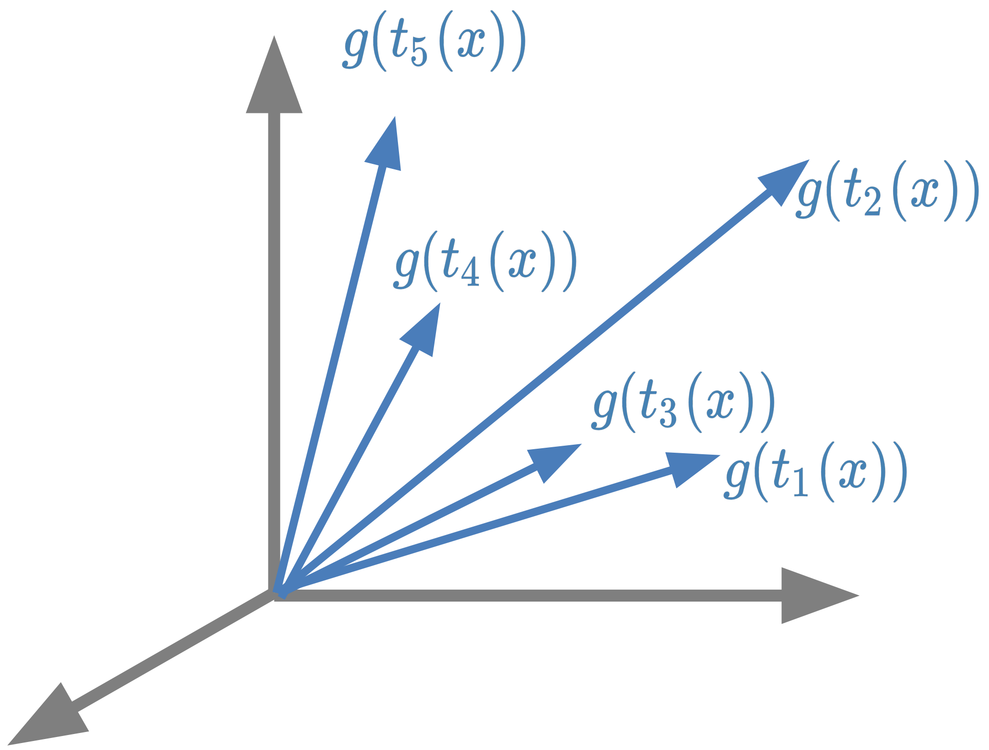
Challenge#2: How to address the exponential growth of the search space?

We use gradient matching as the training signal



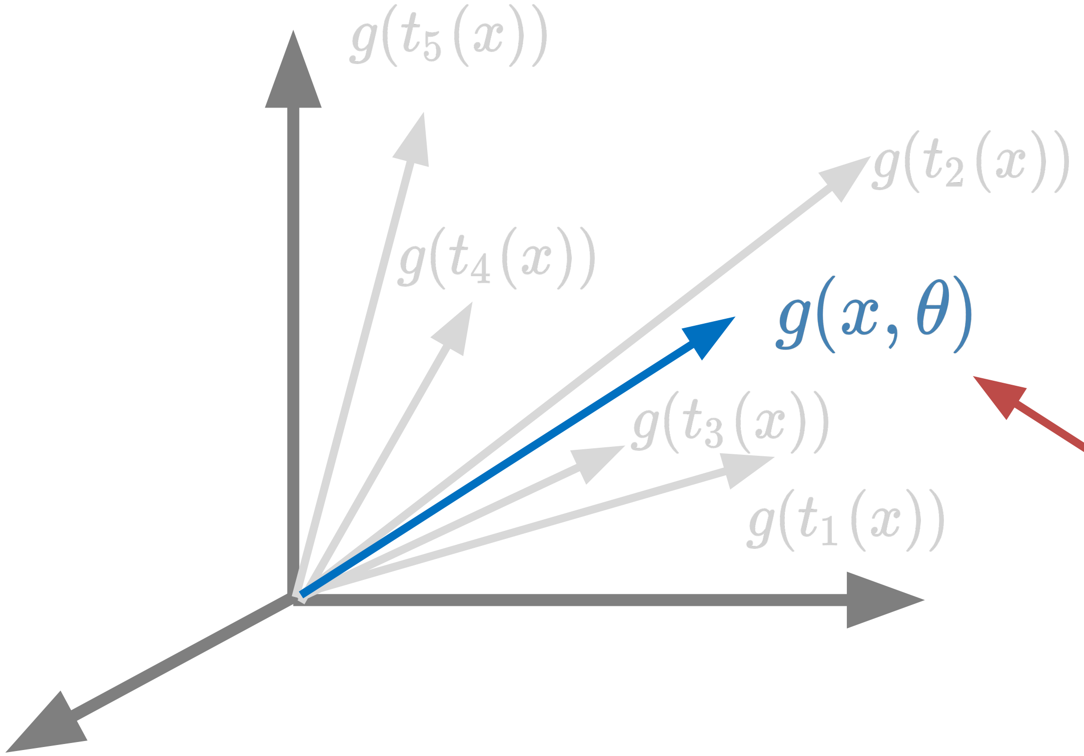
As the distribution of augmented data gets closer to the ture data distribution, the direction of gradient of the augmented data should match the gradient of the validation batch sampled form the true data distribution. We hence optimize the cosine similarity between them.

Gradient Matching



- 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 .

Gradient Matching

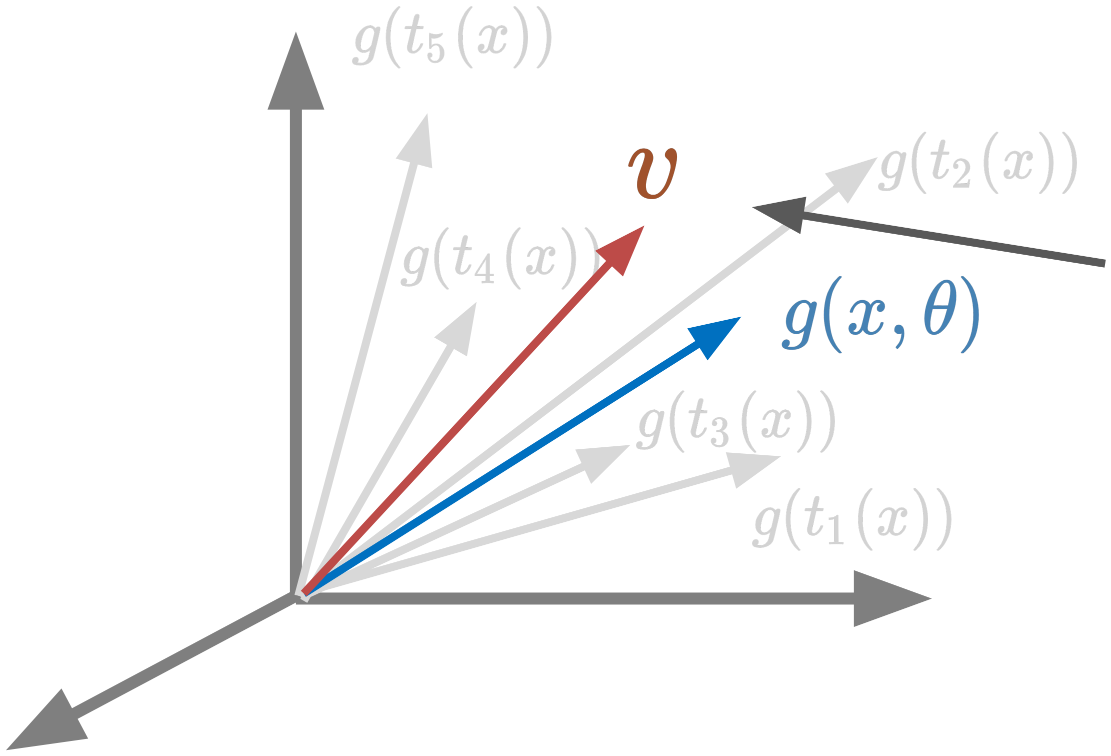


For each transformation t_n, we assign a probability $p_\theta(n)$, which servers as the augmentation policy.

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

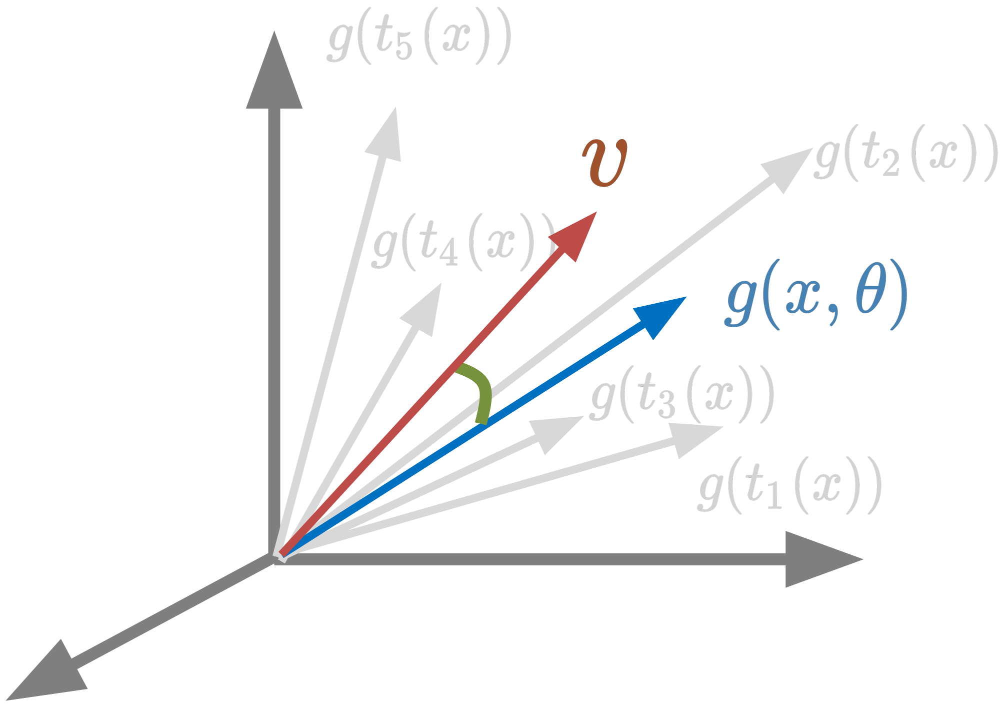
$$g(x;\theta) = \sum_{n=1}^{N} p_\theta(n) g(t_n(x))$$

Gradient Matching



Gradient of validation batch

Gradient Matching



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

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

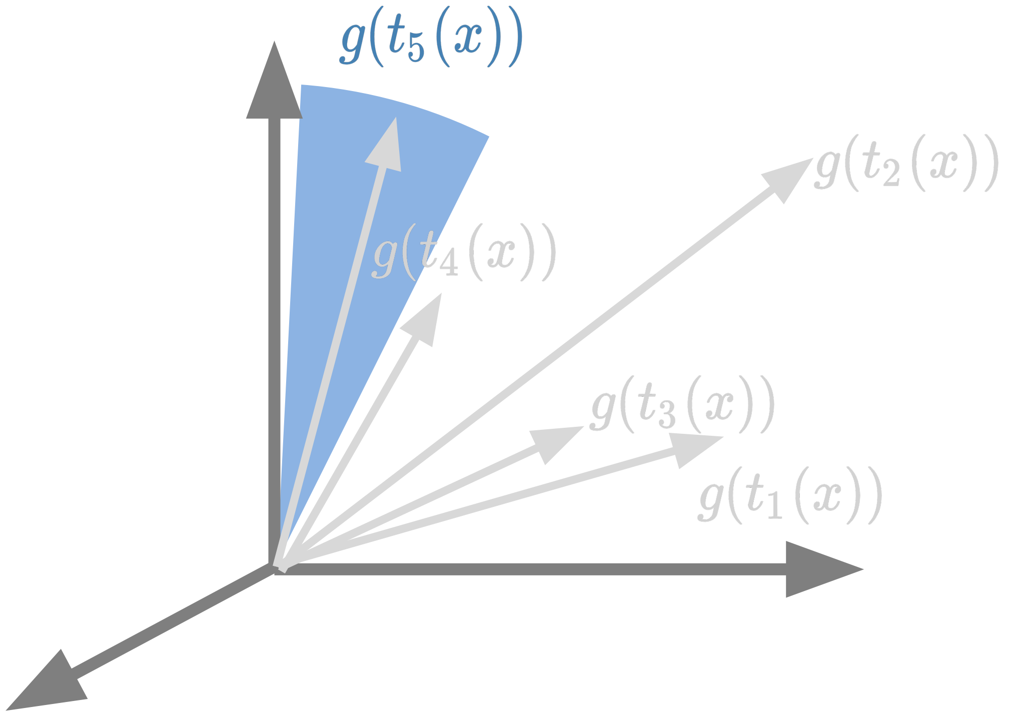
Gradient of validation batch

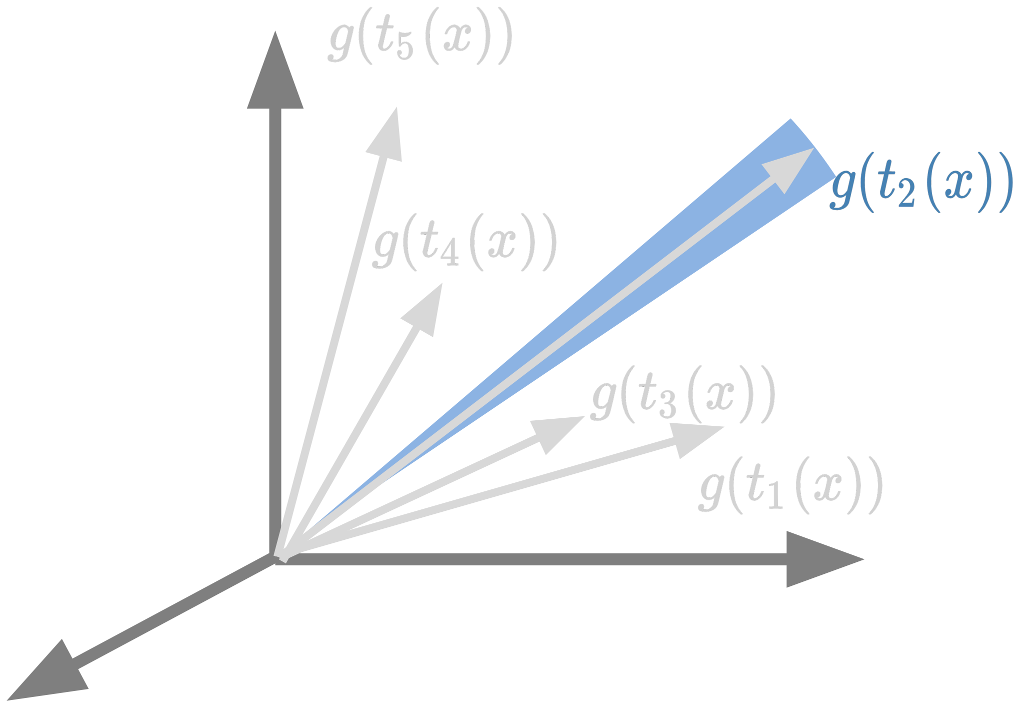
$$\theta = \arg\max_\theta \boxed{\text{cosineSimilarity}}(v, g(x;\theta))$$

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

Regularized Gradient Matching

Penalize the transformation with high variance.

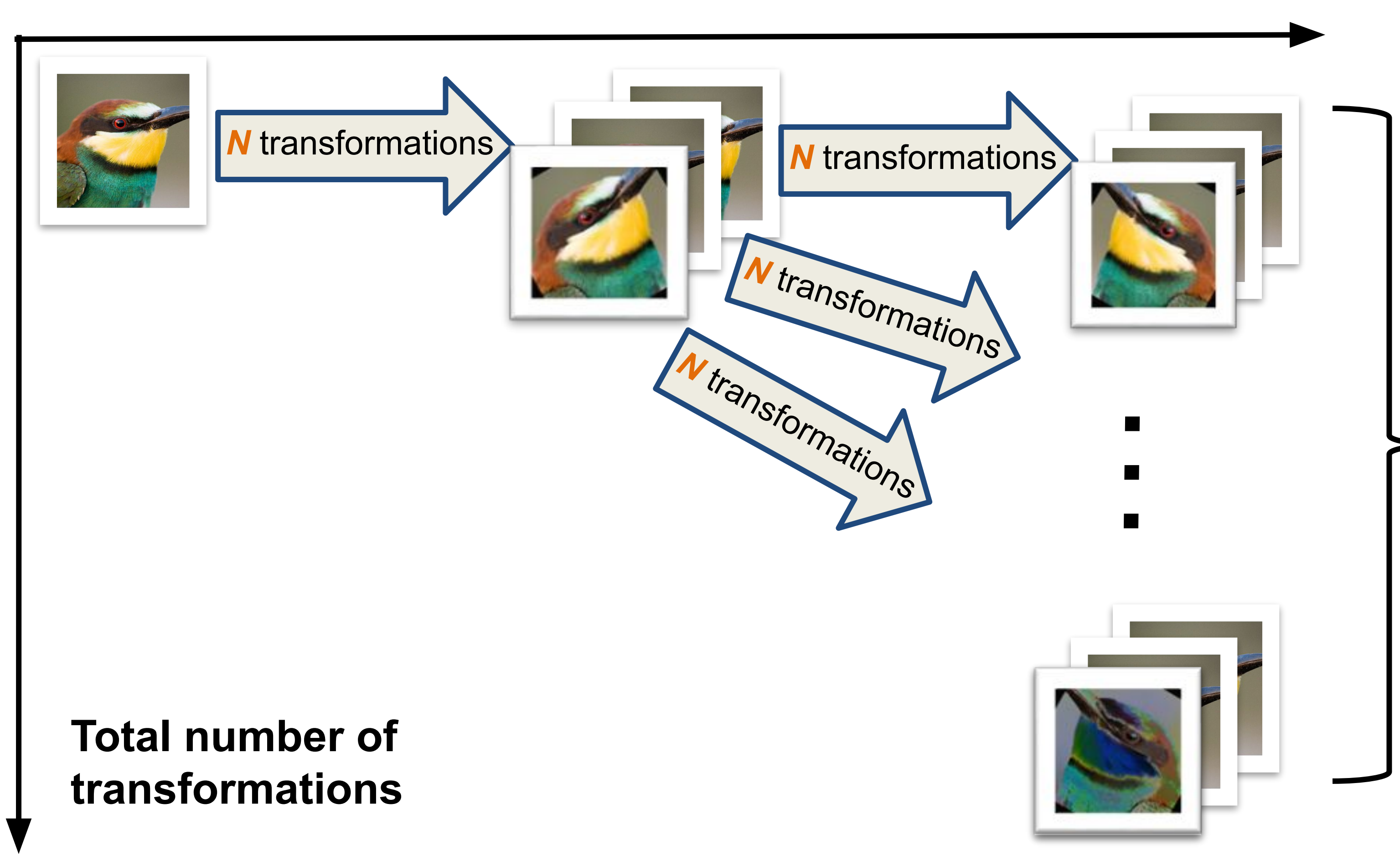




If transformation t_5 exhibits **high variance** for different x, we **decrease** the corresponding probability $p_\theta(5)$.

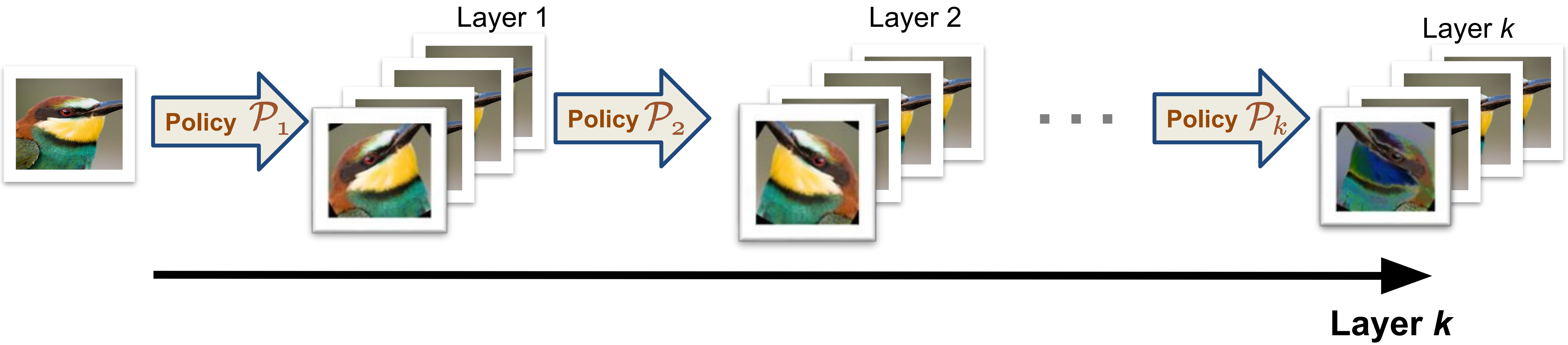
If transformation t_2 exhibits **low variance** for different x, we **increase** the corresponding probability $p_\theta(2)$.

Challenge: Search space grows exponentially when augmentation layers go deep.



The total number of transformations is N^k, which grows exponentially with respect to k.

Our Solution: We use a greedy approach, where for the *k*-th layer we search the optimal policy based on the data distribution augmented by previous *k-1* layers.



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.

Overall Performance

CIFAR-10 /100

	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 $\pm$.

ImageNet

	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 $\pm$.

Understanding DeepAA (1/3)

Effectiveness of Gradient Matching

- 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**.

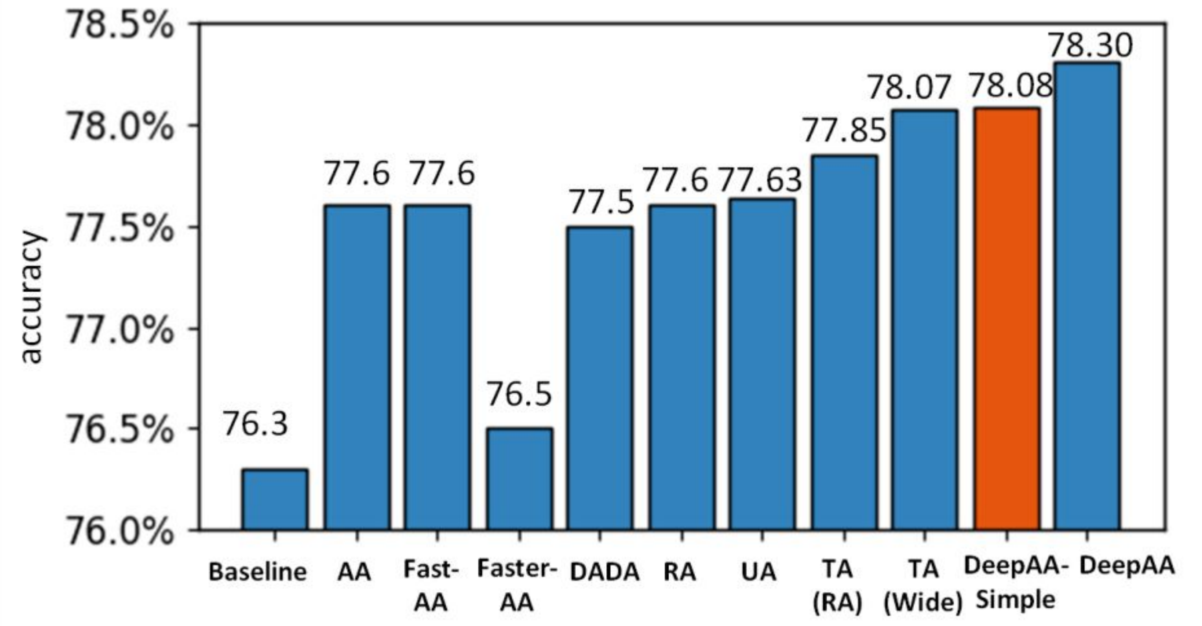
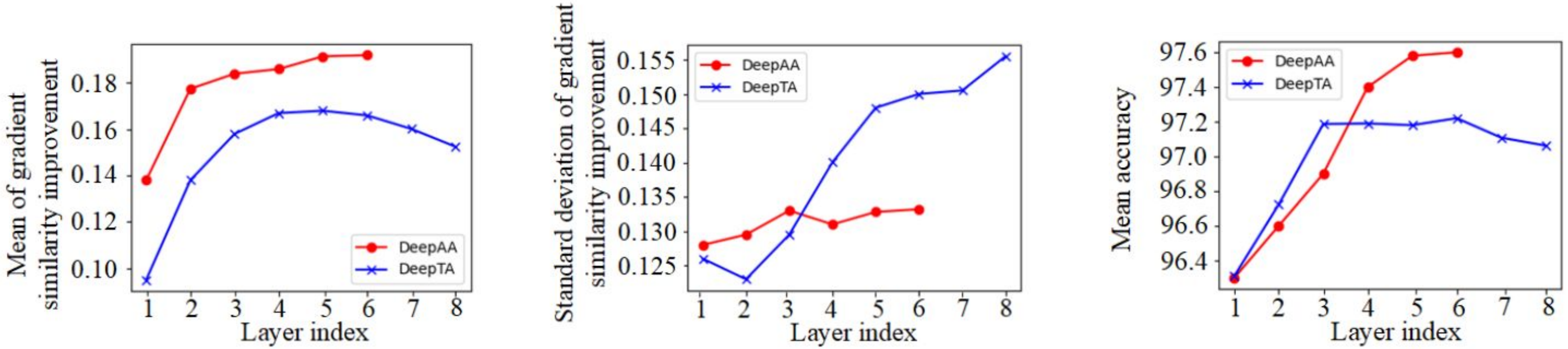


Figure: Top-1 test accuracy (%) on ImageNet of DeepAA-simple, DeepAA, and other automatic augmentation methods on ResNet-50.

Understanding DeepAA (2/3)

Validity of Optimizing with Regularized Gradient Matching



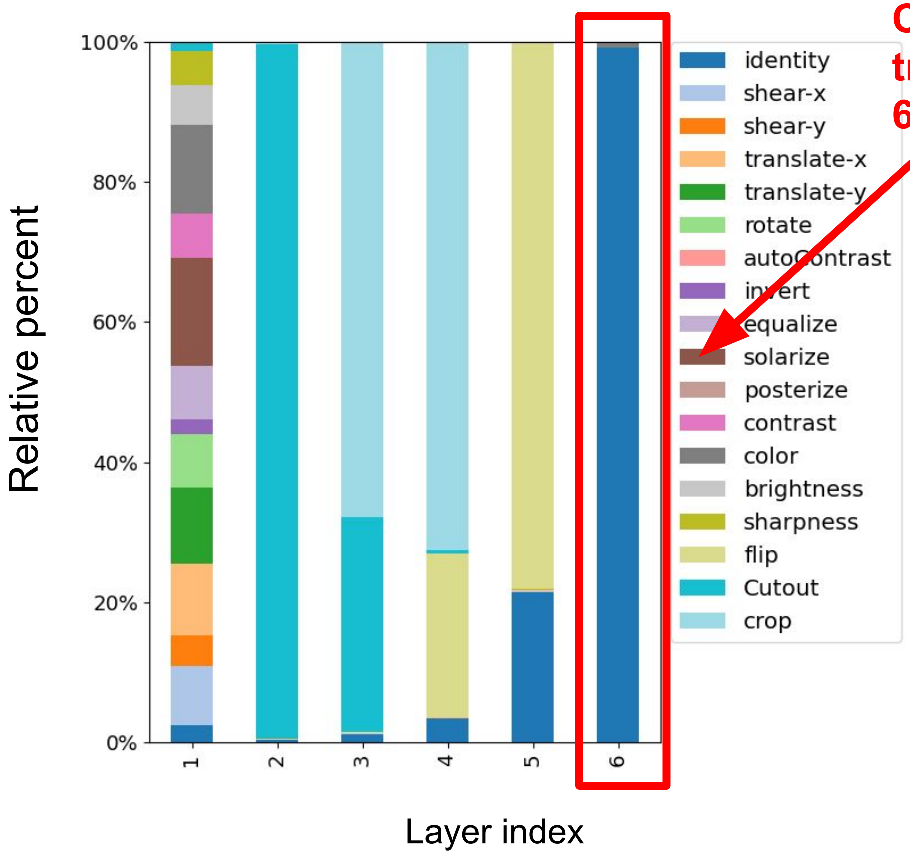
(a) Mean of the gradient similarity improvement (b) Standard deviation of the gradient similarity improvement (c) Mean accuracy over different augmentation depth

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

Understanding DeepAA (3/3)

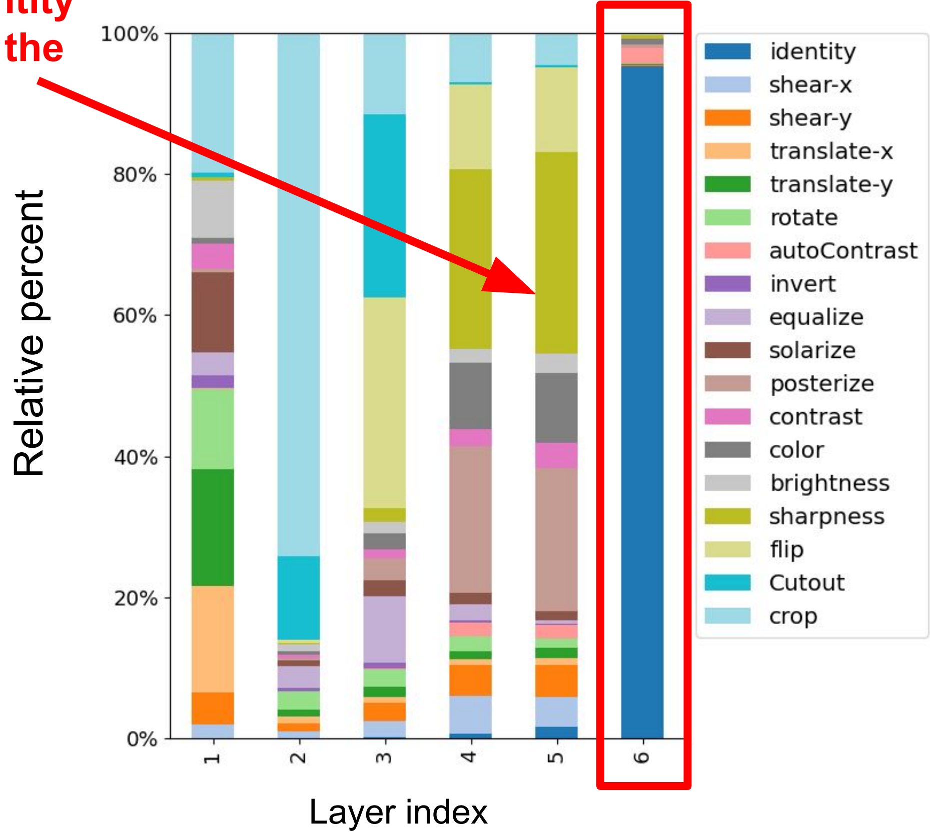
Identify the Optimal Numbers of Augmentation Layers

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



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

Converged to identity transformation at the 6th layer.



Thank You

For more detailed information and other results, please refer to our paper.

arXiv

https://arxiv.org/abs/2203.06172

Our code and augmentation policy are available at GitHub.

GitHub

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