Implicit Semantic Data Augmentation for Deep Networks

Yulin Wang*[1], Xuran Pan*[1], Shiji Song[1], Hong Zhang[2], Cheng Wu[1], Gao Huang[1]

[1]Department of Automation, Tsinghua University; [2]Baidu Inc.

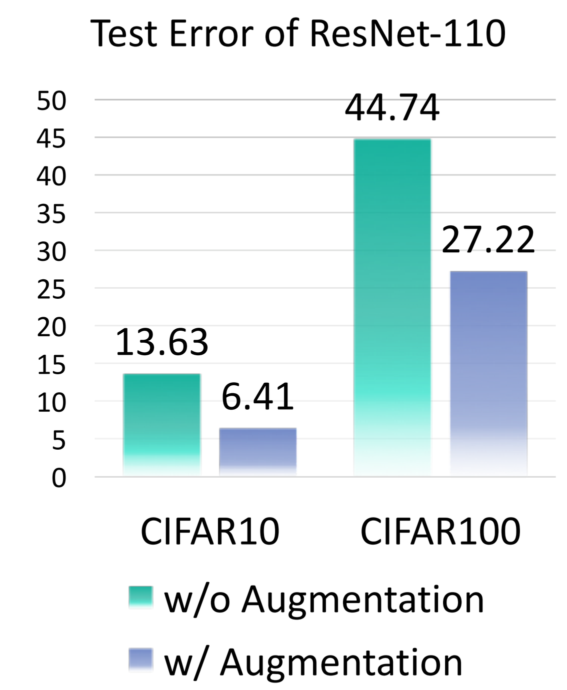
*Equal Contribution; Email: yulin.bh@gmail.com

Code Paper

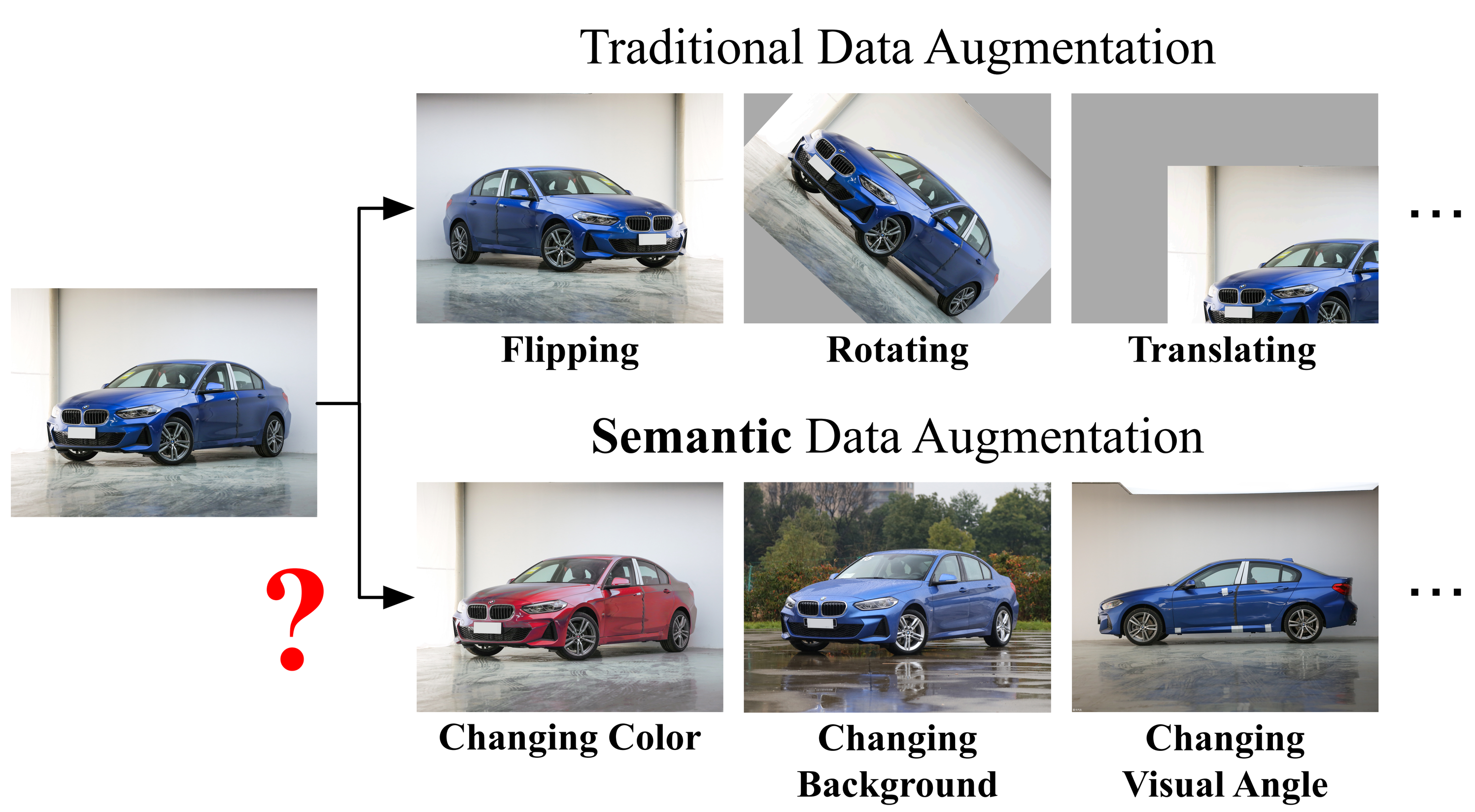
Motivations

Data Augmentation is a famous technique to improve the generalization performance.

- Rotating, flipping, translating, ...

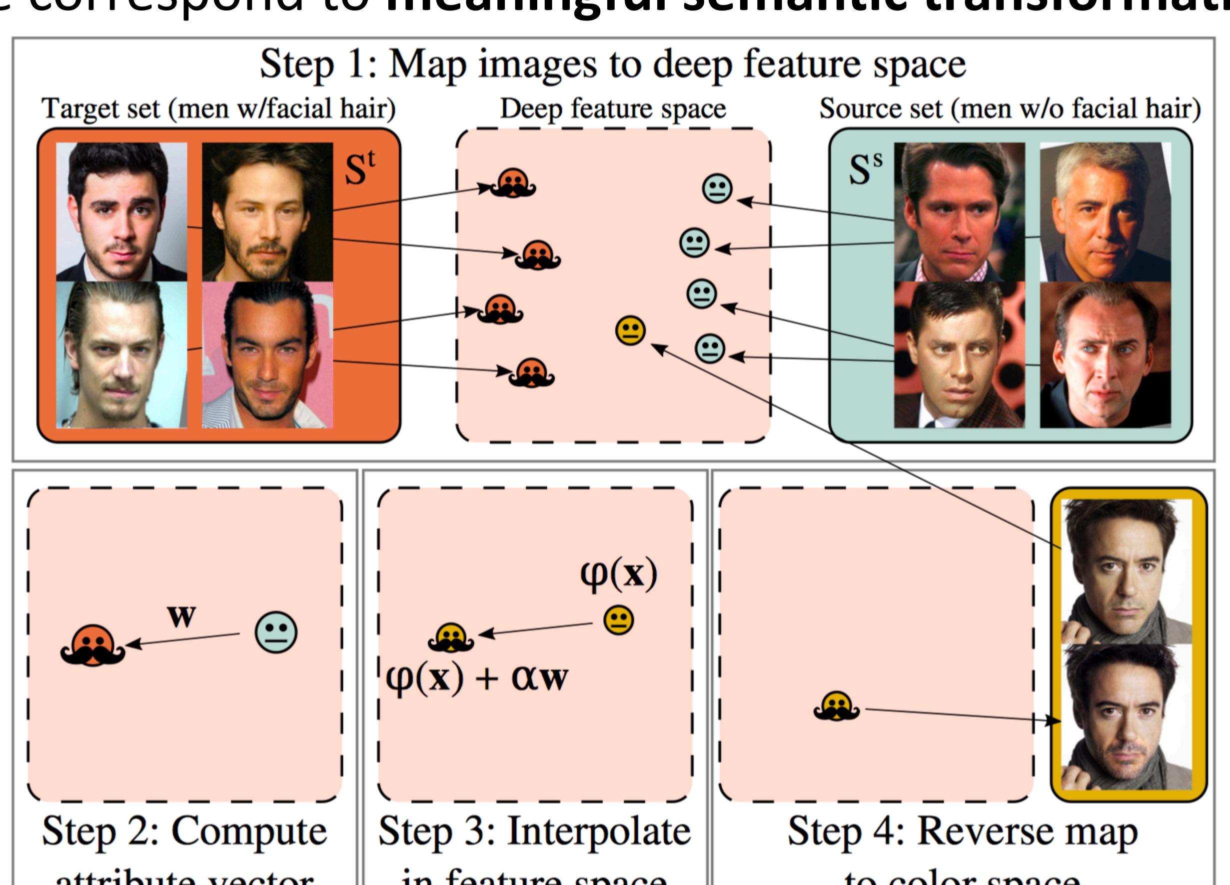


Question: Can we perform **semantic transformations** to complement traditional augmentation techniques?



Preliminaries

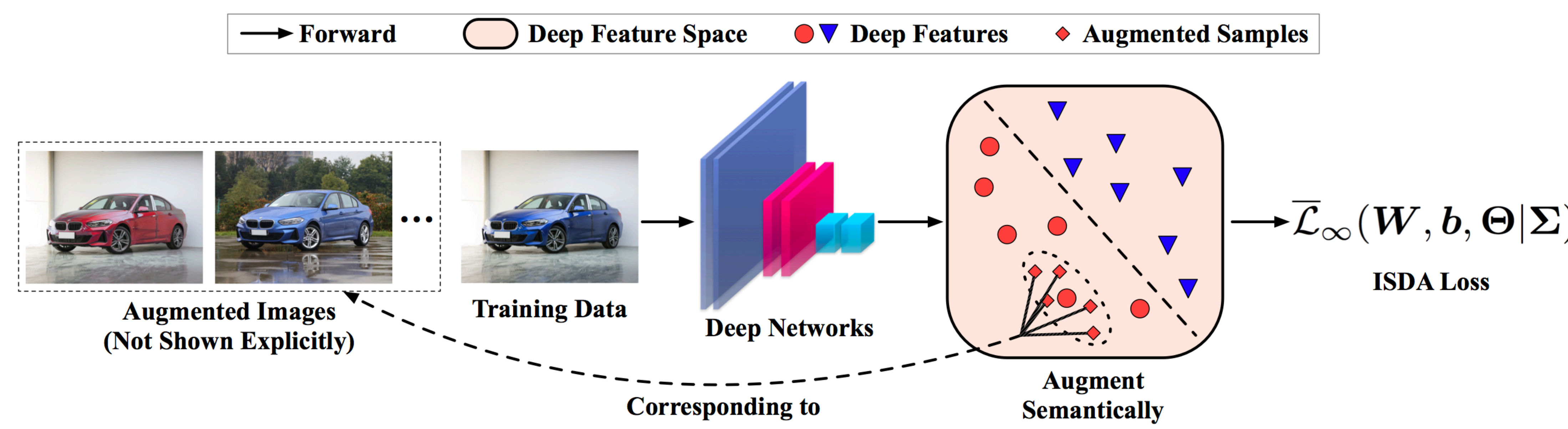
It is shown in [1] that certain directions in the deep feature space correspond to **meaningful semantic transformations**.



Implicit Semantic Data Augmentation (ISDA)

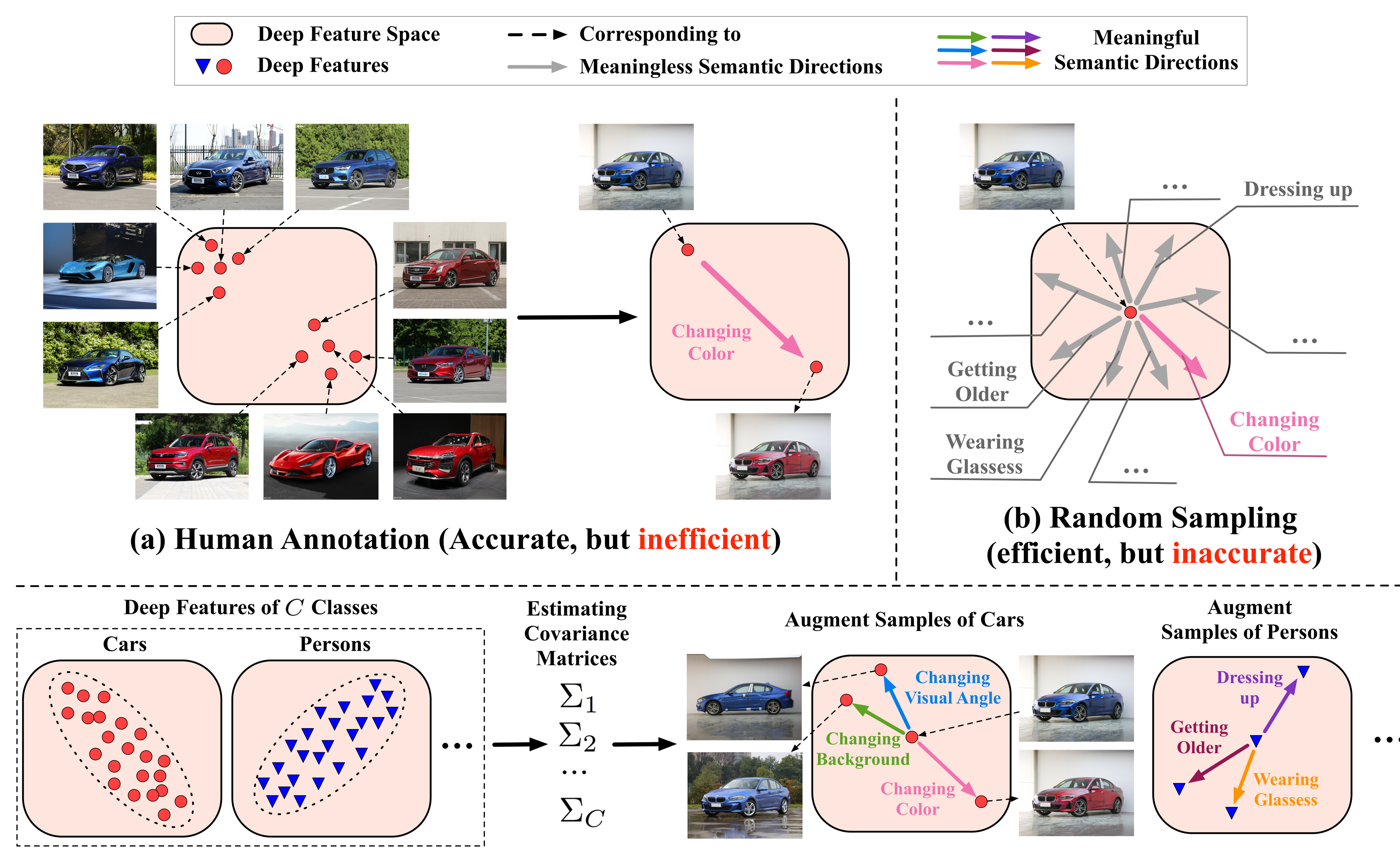
Overview of ISDA

- We propose to perform semantic data augmentation by **translating deep features** along meaningful semantic directions.
- ISDA is highly efficient as it amounts to minimizing **a novel robust loss**.



Find Semantic Directions by Sampling

◆ How to obtain semantic directions for augmentation?



(a) Human Annotation (Accurate, but inefficient)

(b) Random Sampling (efficient, but inaccurate)

(c) Class-Conditional Gaussian Sampling (Our method, a nice trade-off between efficiency and accuracy)

Upper bound of the expected loss

◆ ISDA boils down to a robust loss function.

Given a deep feature vector $\boldsymbol{a}_i$, the augmented feature is sampled from:

$$\tilde{\boldsymbol{a}}_i \sim \mathcal{N}(\boldsymbol{a}_i, \lambda\boldsymbol{\Sigma}_{y_i})$$

Naively, we can sample M times from $\tilde{\boldsymbol{a}}_i$, and minimize the average loss:

$$\mathcal{L}_M = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{k=1}^{M}[-\log(\frac{e^{w_{y_i}^T a_i^k + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T a_i^k + b_j}})].$$

To be **more efficient**, we consider the case that $M \to \infty$:

$$\mathcal{L}_\infty = \lim_{M\to\infty}\mathcal{L}_M = \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}_{\tilde{a}_i}[-\log(\frac{e^{w_{y_i}^T \tilde{a}_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T \tilde{a}_i + b_j}})].$$

An easy-to-compute upper bound of $\mathcal{L}_\infty$ bound is derived as the surrogate loss:

Proposition 1: *Suppose that $\tilde{\boldsymbol{a}}_i \sim \mathcal{N}(\boldsymbol{a}_i, \lambda\boldsymbol{\Sigma}_{y_i})$, an upper bound of $\mathcal{L}_\infty$ is*

$$\mathcal{L}_\infty \leq \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{e^{w_{y_i}^T a_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T a_i + b_j + \frac{\lambda}{2}(w_j^T - w_{y_i}^T)\Sigma_{y_i}(w_j - w_{y_i})}}\right) \triangleq \bar{\mathcal{L}}_\infty.$$

Results

Consistent improvements with various popular deep models:

Method	Params	CIFAR-10	CIFAR-100
ResNet-32 [4]	0.5M	$7.39 \pm 0.10\%$	$31.20 \pm 0.41\%$
ResNet-32 + ISDA	0.5M	$\mathbf{7.09 \pm 0.12\%}$	$\mathbf{30.27 \pm 0.34\%}$
ResNet-110 [4]	1.7M	$6.76 \pm 0.34\%$	$28.67 \pm 0.44\%$
ResNet-110 + ISDA	1.7M	$\mathbf{6.33 \pm 0.19\%}$	$\mathbf{27.57 \pm 0.46\%}$
SE-ResNet-110 [33]	1.7M	$6.14 \pm 0.17\%$	$27.30 \pm 0.03\%$
SE-ResNet-110 + ISDA	1.7M	$\mathbf{5.96 \pm 0.21\%}$	$\mathbf{26.63 \pm 0.21\%}$
Wide-ResNet-16-8 [34]	11.0M	$4.25 \pm 0.18\%$	$20.24 \pm 0.27\%$
Wide-ResNet-16-8 + ISDA	11.0M	$\mathbf{4.04 \pm 0.29\%}$	$\mathbf{19.91 \pm 0.21\%}$
Wide-ResNet-28-10 [34]	36.5M	$3.82 \pm 0.15\%$	$18.53 \pm 0.07\%$
Wide-ResNet-28-10 + ISDA	36.5M	$\mathbf{3.58 \pm 0.15\%}$	$\mathbf{17.98 \pm 0.15\%}$
ResNeXt-29, 8x64d [35]	34.4M	$3.86 \pm 0.14\%$	$18.16 \pm 0.13\%$
ResNeXt-29, 8x64d + ISDA	34.4M	$\mathbf{3.67 \pm 0.12\%}$	$\mathbf{17.43 \pm 0.25\%}$
DenseNet-BC-100-12 [5]	0.8M	$4.90 \pm 0.08\%$	$22.61 \pm 0.10\%$
DenseNet-BC-100-12 + ISDA	0.8M	$\mathbf{4.54 \pm 0.07\%}$	$\mathbf{22.10 \pm 0.34\%}$
DenseNet-BC-190-40 [5]	25.6M	3.52%	17.74%
DenseNet-BC-190-40 + ISDA	25.6M	**3.24%**	**17.42%**

Significantly complement SOTA non-semantic augmentation techniques:

Dataset	Networks	Cutout [31]	Cutout + ISDA	AA [32]	AA + ISDA
CIFAR-10	Wide-ResNet-28-10 [34]	$2.99 \pm 0.06\%$	$\mathbf{2.83 \pm 0.04\%}$	$2.65 \pm 0.07\%$	$\mathbf{2.56 \pm 0.01\%}$
	Shake-Shake (26, 2x32d) [36]	$3.16 \pm 0.09\%$	$\mathbf{2.93 \pm 0.03\%}$	$2.89 \pm 0.09\%$	$\mathbf{2.68 \pm 0.12\%}$
	Shake-Shake (26, 2x112d) [36]	2.36%	**2.25%**	2.01%	**1.82%**
CIFAR-100	Wide-ResNet-28-10 [34]	$18.05 \pm 0.25\%$	$\mathbf{17.02 \pm 0.11\%}$	$16.60 \pm 0.40\%$	$\mathbf{15.62 \pm 0.32\%}$
	Shake-Shake (26, 2x32d) [36]	$18.92 \pm 0.21\%$	$\mathbf{18.17 \pm 0.08\ \%}$	$17.50 \pm 0.19\%$	$\mathbf{17.21 \pm 0.33\%}$
	Shake-Shake (26, 2x112d) [36]	$17.34 \pm 0.28\%$	$\mathbf{16.24 \pm 0.20\ \%}$	$15.21 \pm 0.20\%$	$\mathbf{13.87 \pm 0.26\%}$

[1] P. Upchurch, J. R. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Q. Weinberger, Deep feature interpolation for image content changes.