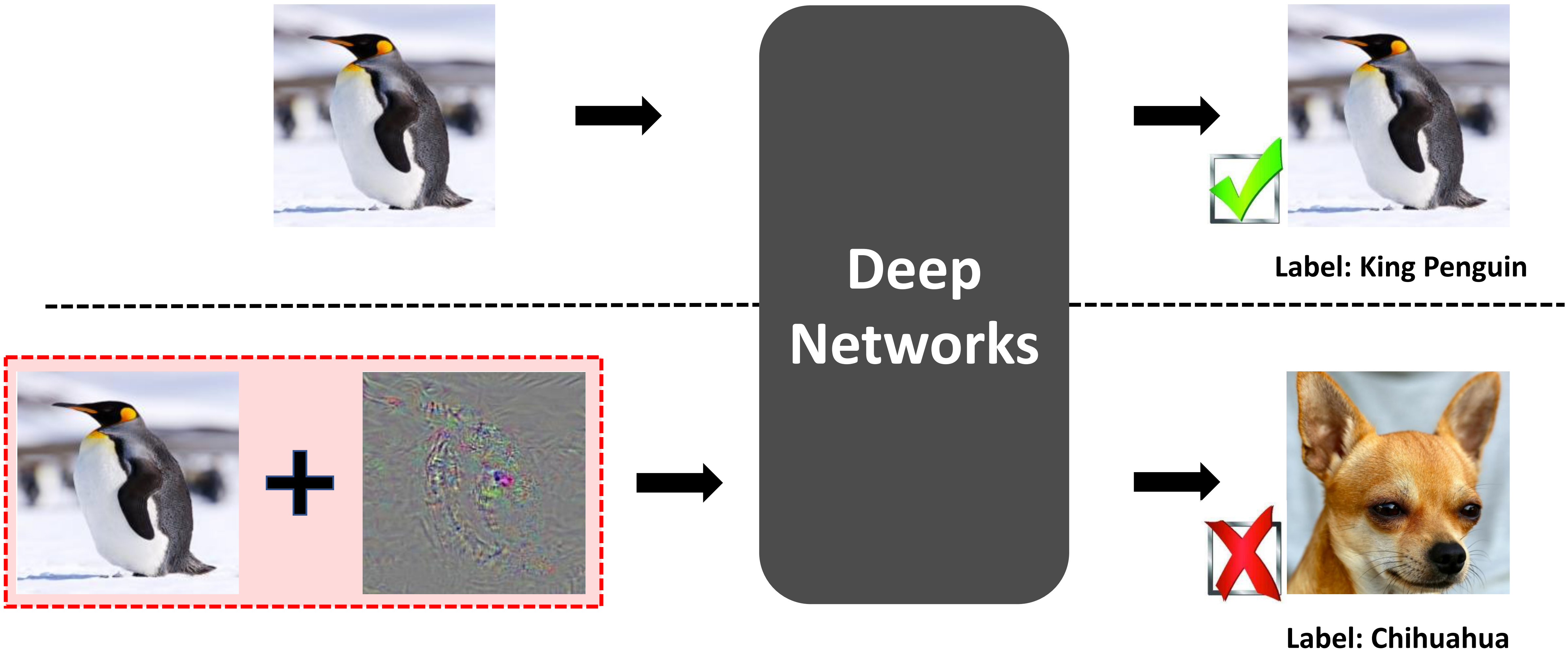
Adversarial Examples Improve Image Recognition (CVPR'20)





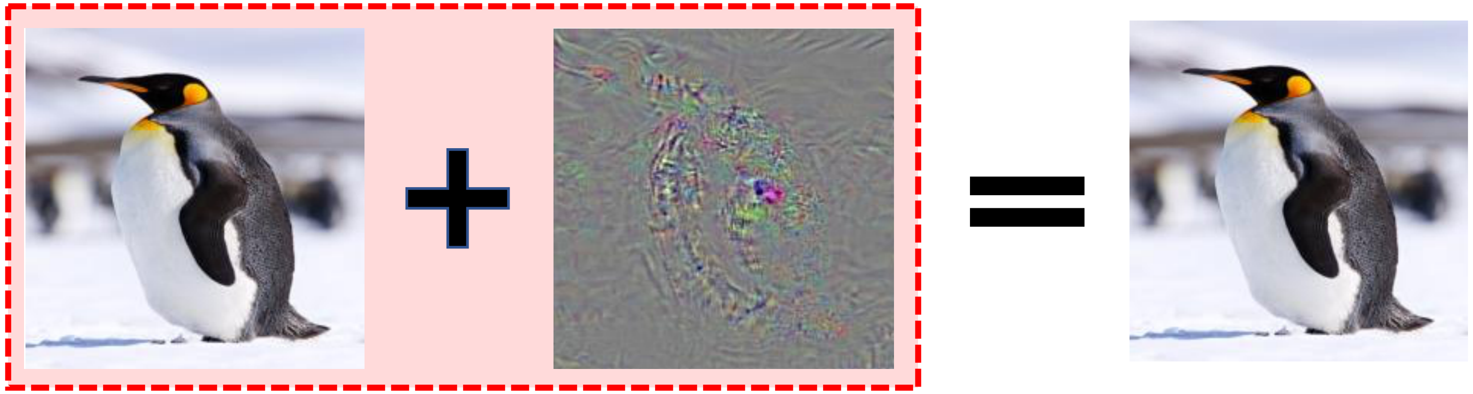
Recall: What Are Adversarial Examples



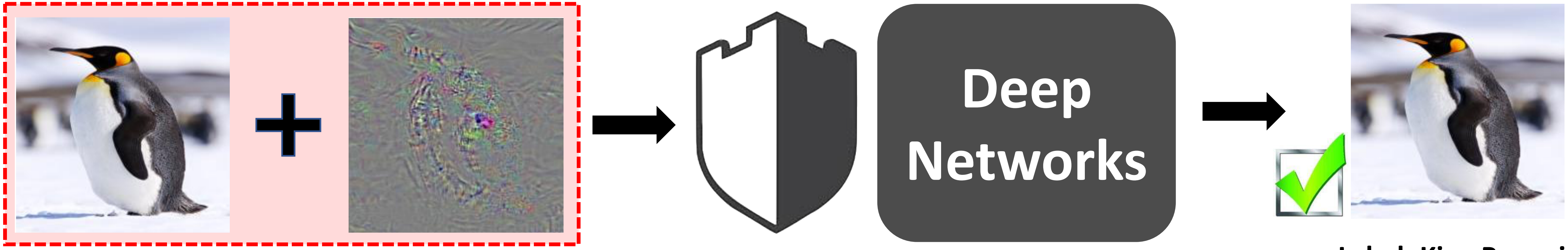
Recall: What Are Adversarial Examples

We call such images as

Adversarial Examples



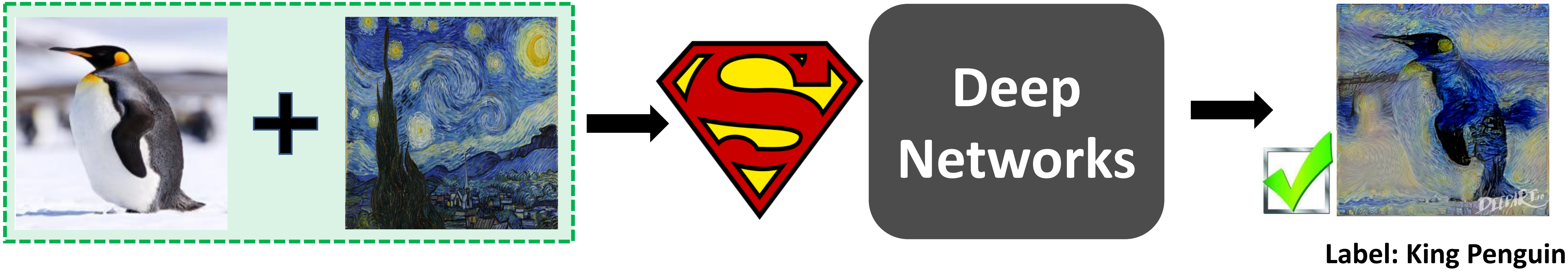
Adversarial Examples Are **THREATS** to Deep Networks



Label: King Penguin

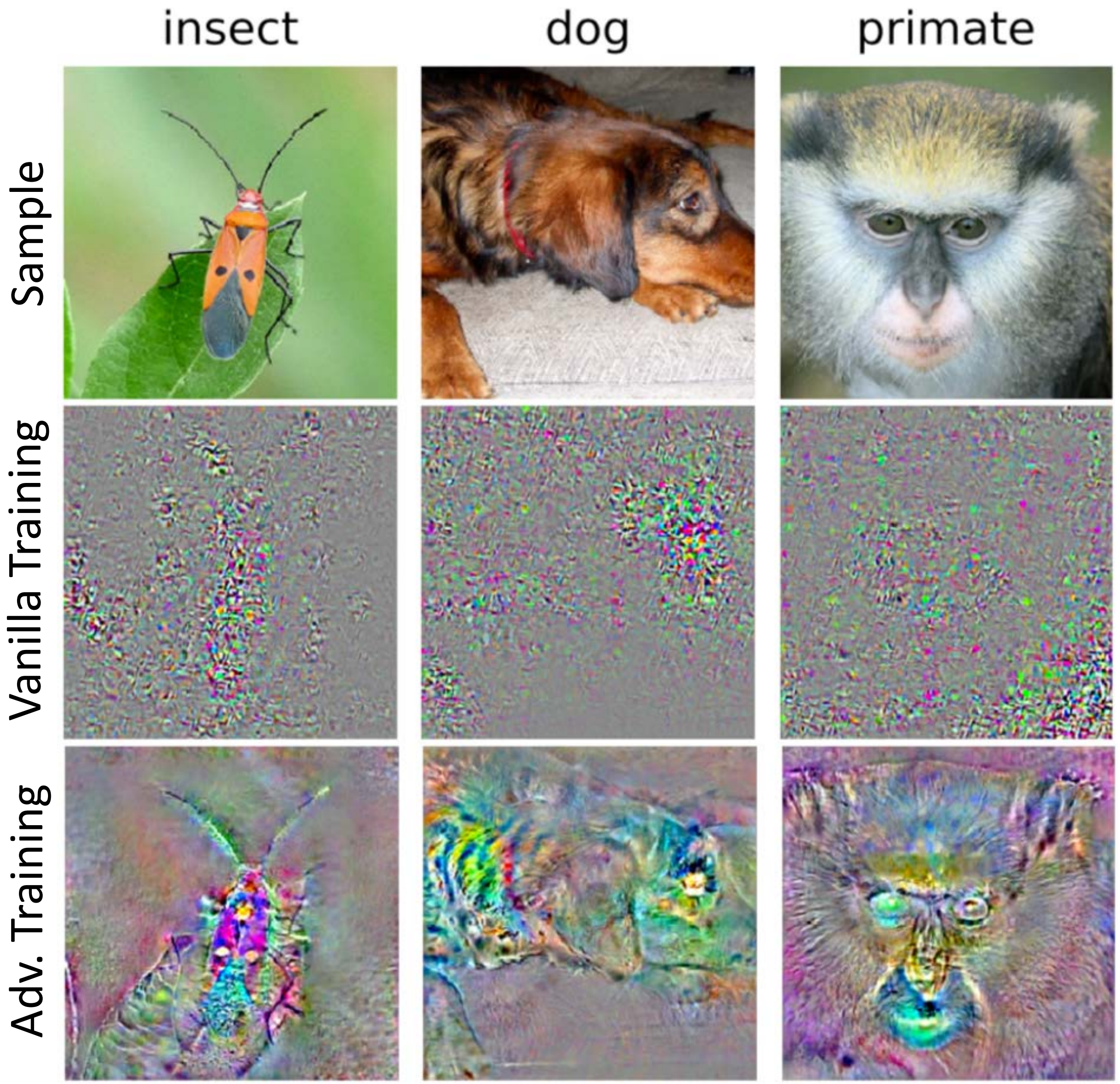
Icons made by Freepik from www.flaticon.com

Can we use Adversarial Examples to **HELP** Deep Networks?
e.g., to improve the representation learning?



Icons made by Freepik from www.flaticon.com

Motivation: Adversarial examples provide VALUABLE & NEW features



Tsipras et al. [9] shows that the loss gradient w.r.t. the input pixel of adversarially trained models is **HUMAN-ALIGNED**

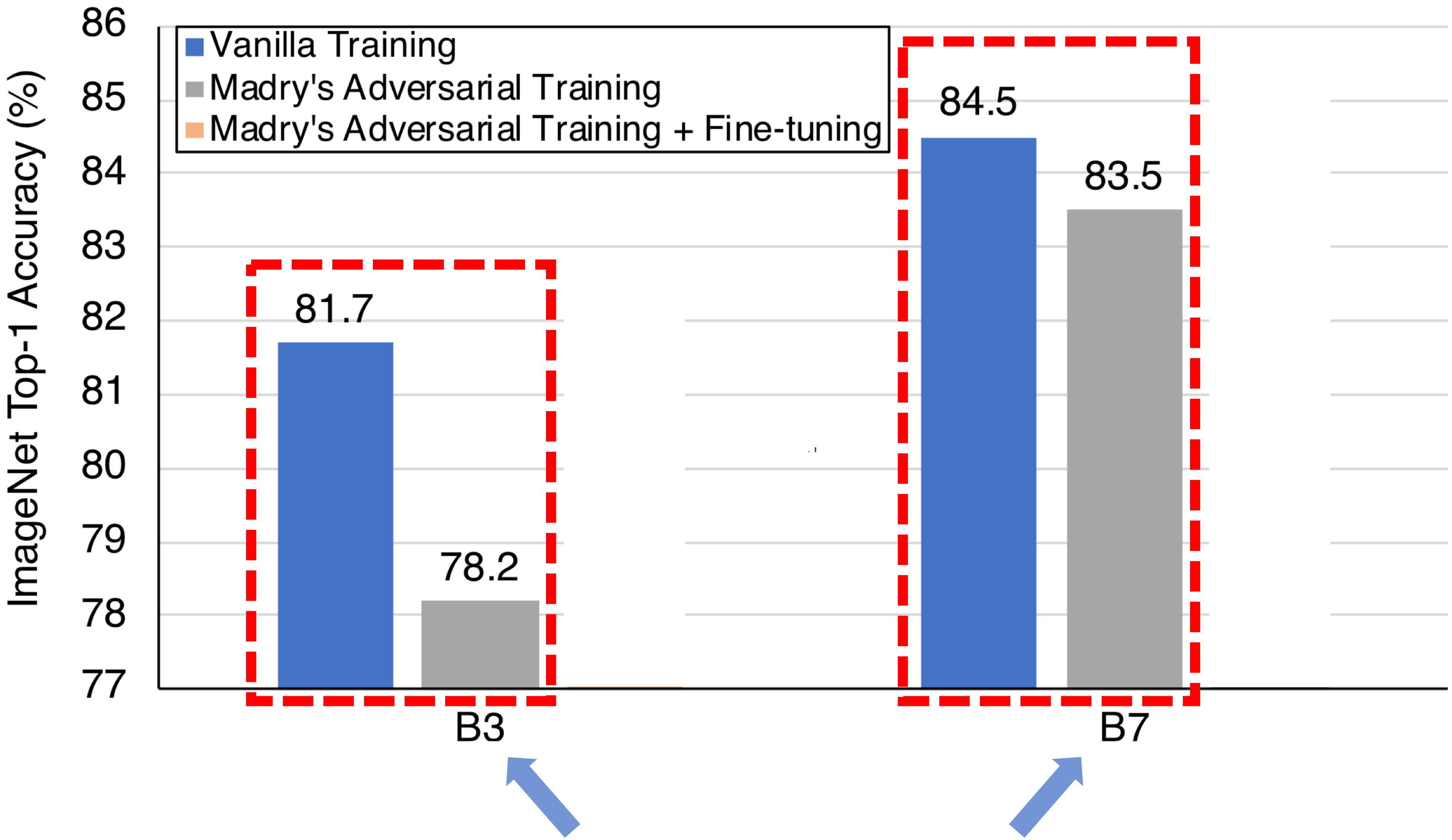
[9] Dimitris Tsipras, et al. ”Robustness May Be at Odds with Accuracy." In *ICLR,* 2019.

Motivation: Adversarial examples provide VALUABLE & NEW features

Santurkar et al. [10] shows that an adversarially trained model are pretty good at tackle several **IMAGE SYNTHESIS TASKS**

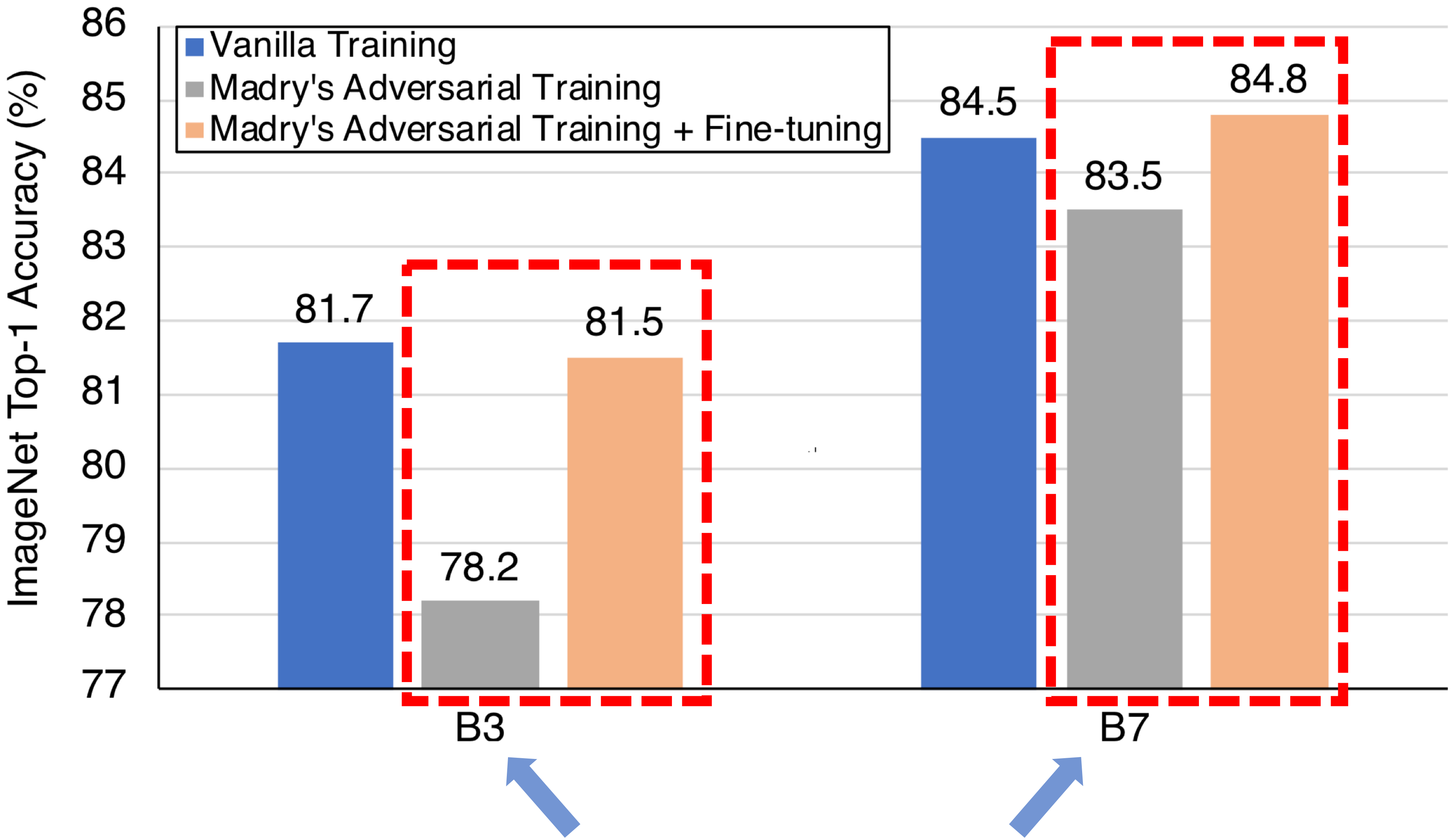
[9] Santurkar, Shibani, et al. "Computer vision with a single (robust) classifier." In *NeurIPS,* 2019.

BUT using features from adversarial examples ALONE are NOT ENOUGH



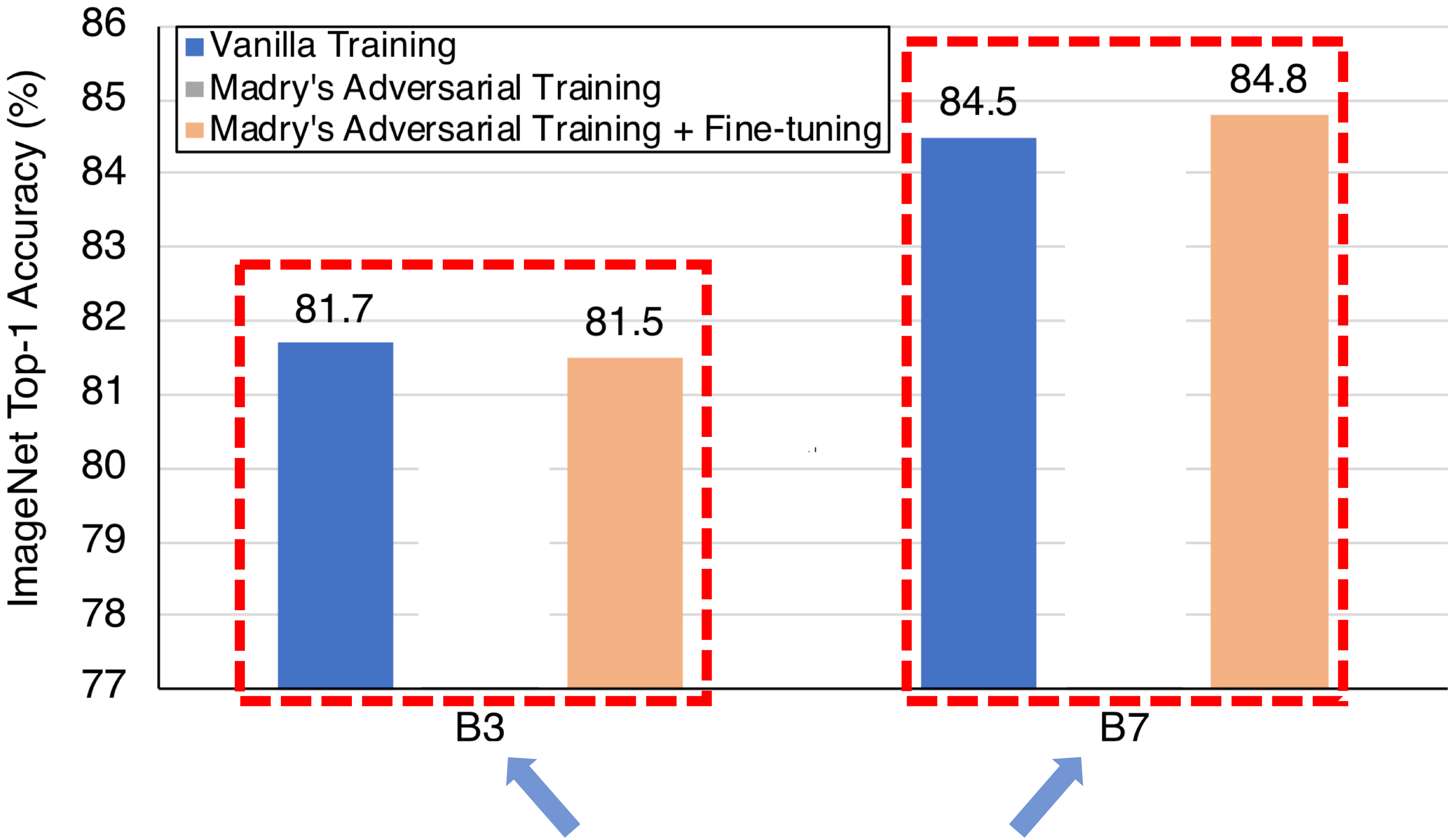
Training **EXCLUSIVELY** on adversarial examples **DEGRADES** performance on clean images

Bridging this distribution mismatch can IMPROVE performance



Simply **FINETUNING** with clean images **IMPROVES** performance on clean images

Bridging this distribution mismatch can IMPROVE performance



CAN WE DO BETTER?

Our Solution: joint training but with distinction

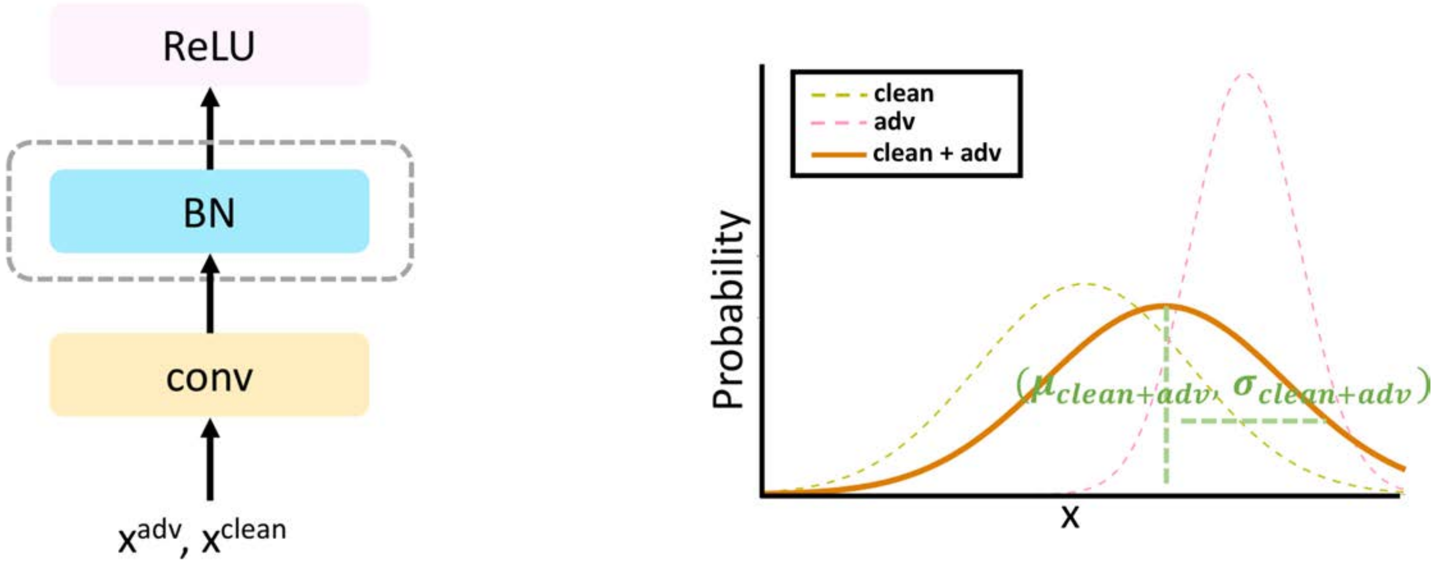
Our Solution: JOINT TRAINING but with distinction

Finetuning may **OVERRIDE** features learned from adversarial examples, therefore it is better to jointly train with adversarial examples and clean images as in [12]

$$\min_{\theta}\Big[\underbrace{\max_{\mathbf{r}}\ \mathrm{loss}(f(\mathrm{xadv}),\ \mathrm{ytrue};\ \theta)}_{\textbf{adversarial examples}} + \underbrace{\mathrm{loss}(f(x),\ \mathrm{ytrue};\ \theta)}_{\textbf{clean images}}\Big]$$

[12] Ian Goodfellow, et al. "Explaining and harnessing adversarial examples." In *ICLR,* 2015.

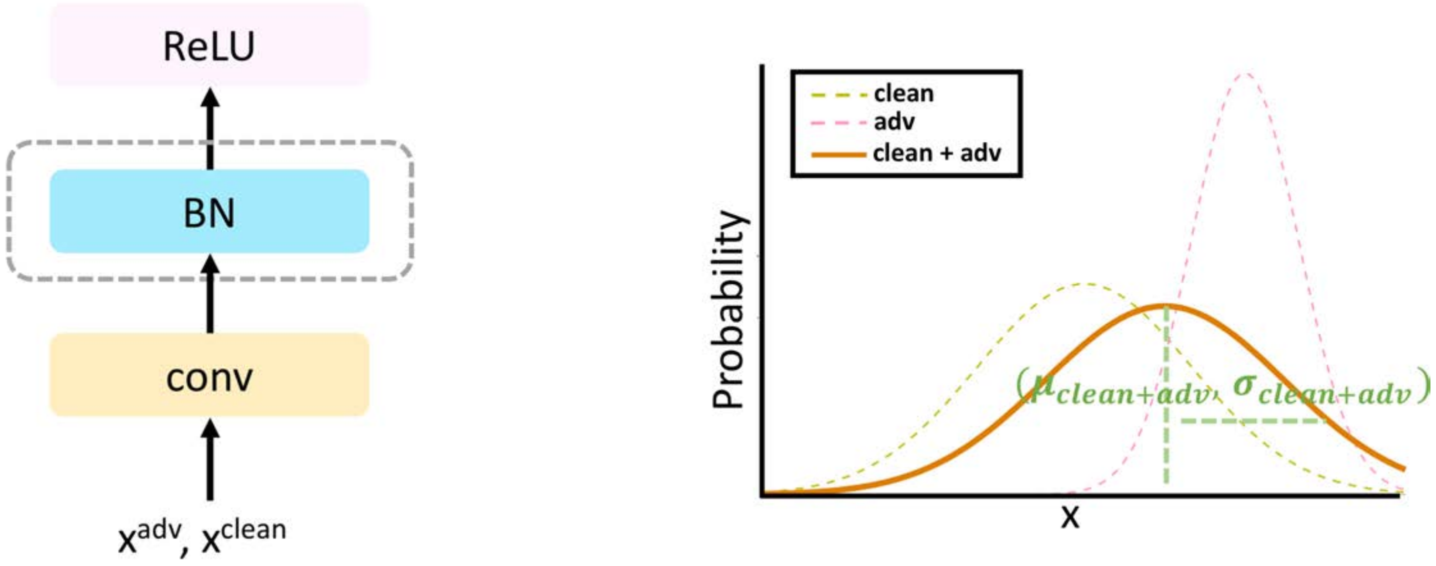
Our Solution: joint training BUT WITH DISTINCTION



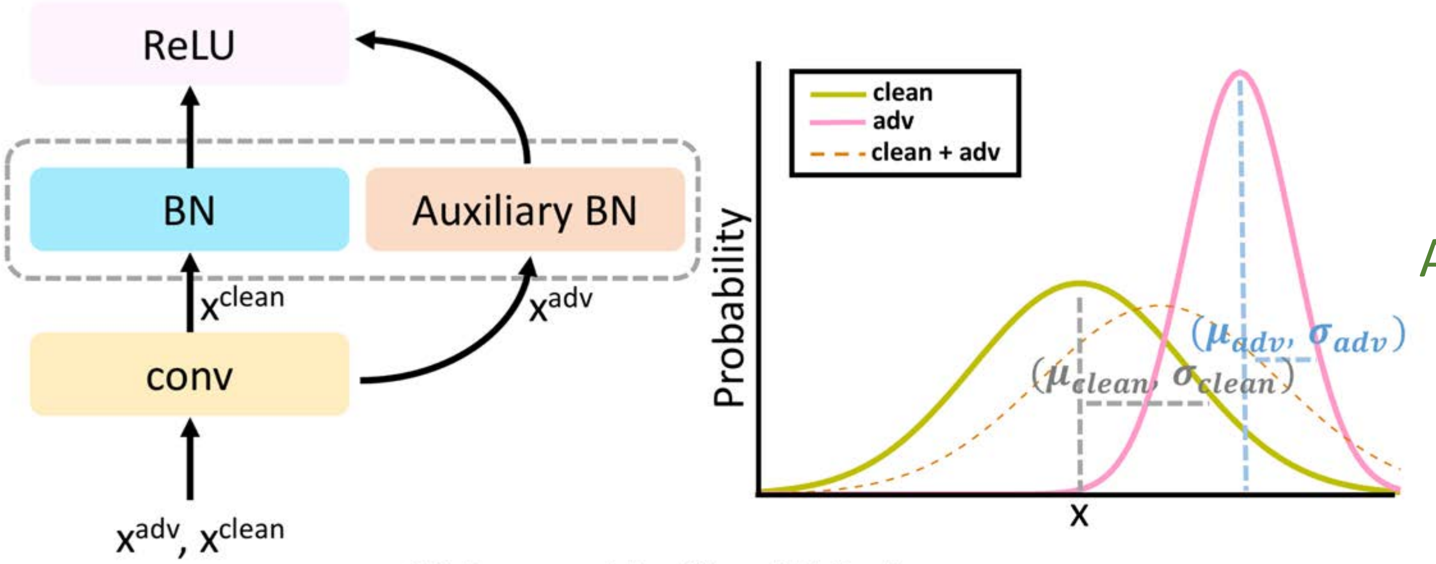
(a) Traditional BN

The statistics estimation at BN may be confused when facing a mixture distribution

Our Solution: joint training BUT WITH DISTINCTION



(a) Traditional BN



(b) Proposed Auxiliary BN Design

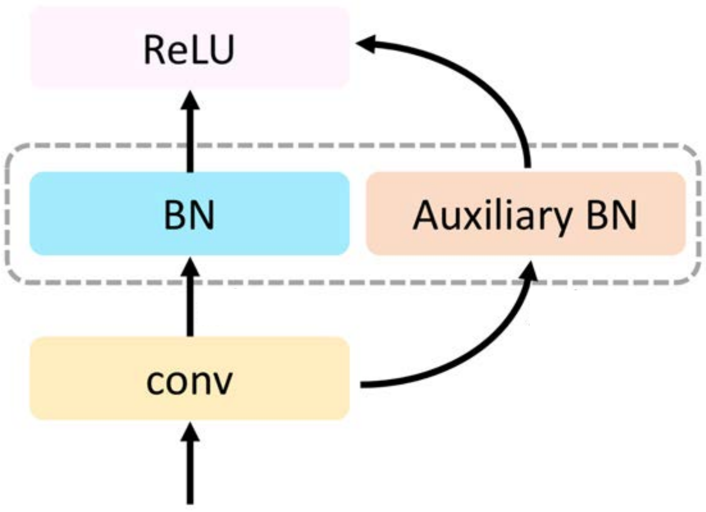
Auxiliary BN guarantees that data from different distributions are normalized separately

Adversarial Propagation (AdvProp)

Algorithm 1: Pseudo code of AdvProp

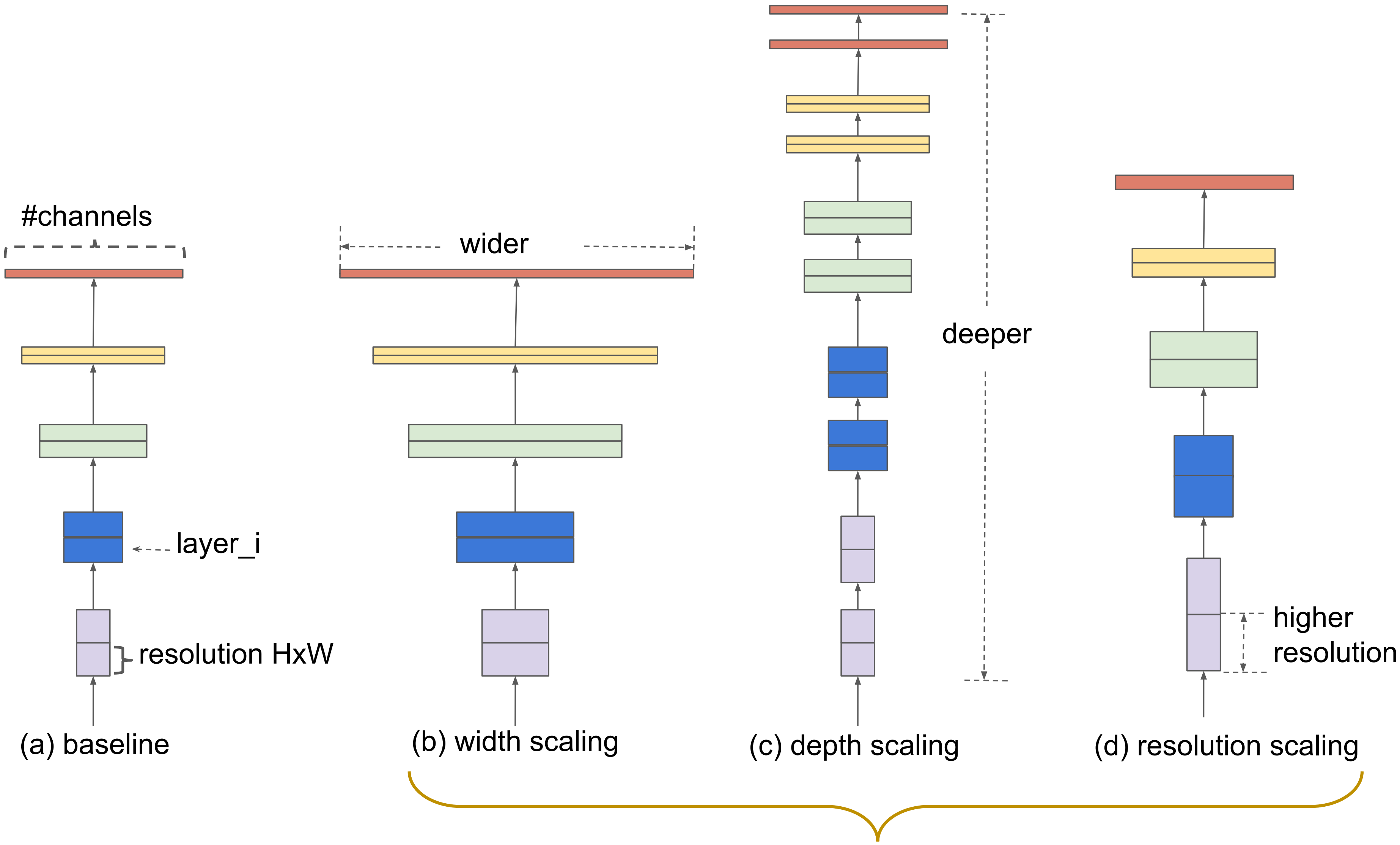
Data: A set of clean images with labels;
Result: Network parameter θ;
for *each training step* **do**
  Sample a clean image mini-batch x^c with label y;
  Generate the corresponding adversarial mini-batch x^a using the auxiliary BNs;
  Compute loss $L^c(\theta, x^c, y)$ on clean mini-batch x^c using the main BNs;
  Compute loss $L^a(\theta, x^a, y)$ on adversarial mini-batch x^a using the auxiliary BNs;
  Minimize the total loss w.r.t. network parameter $\arg\min_{\theta} L^a(\theta, x^a, y) + L^c(\theta, x^c, y)$.
end
return θ

Adversarial Propagation (AdvProp)



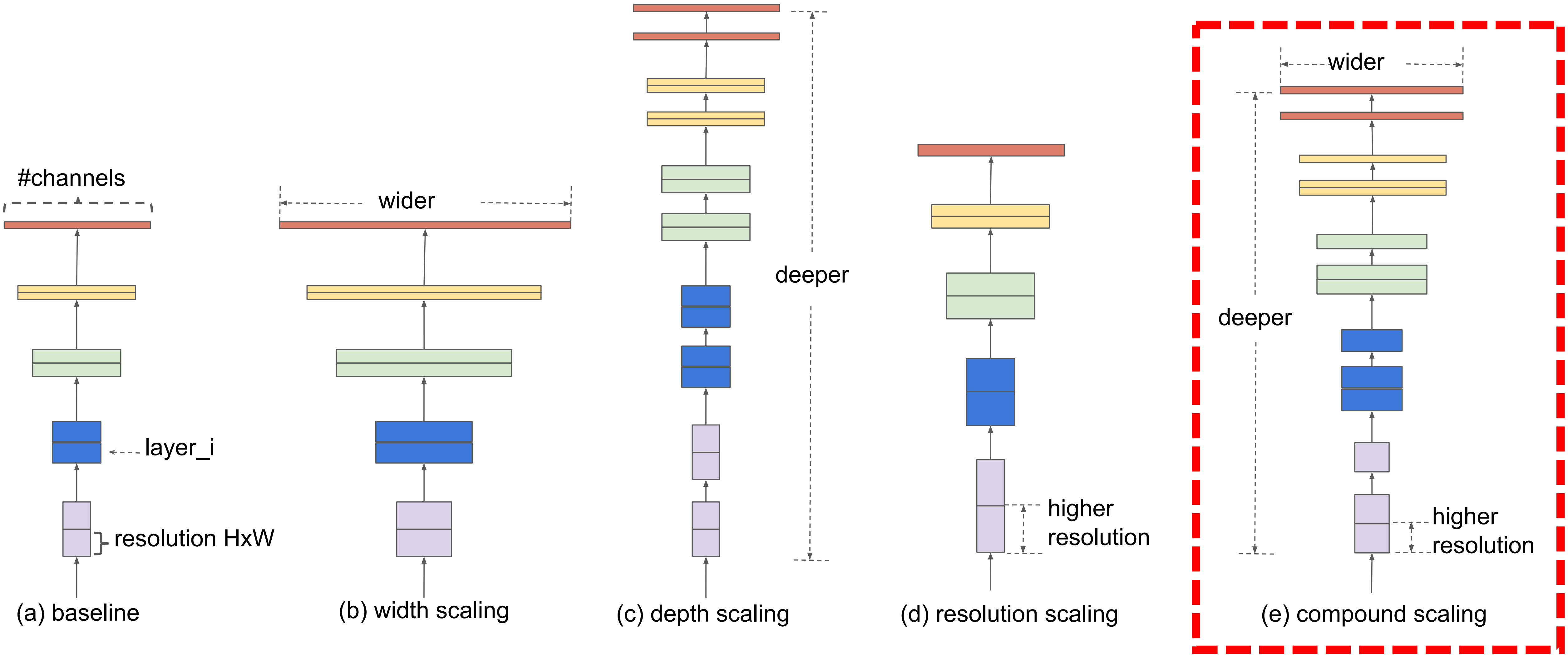
Only Main BN is used at the inference stage

Backbone --- EfficientNet



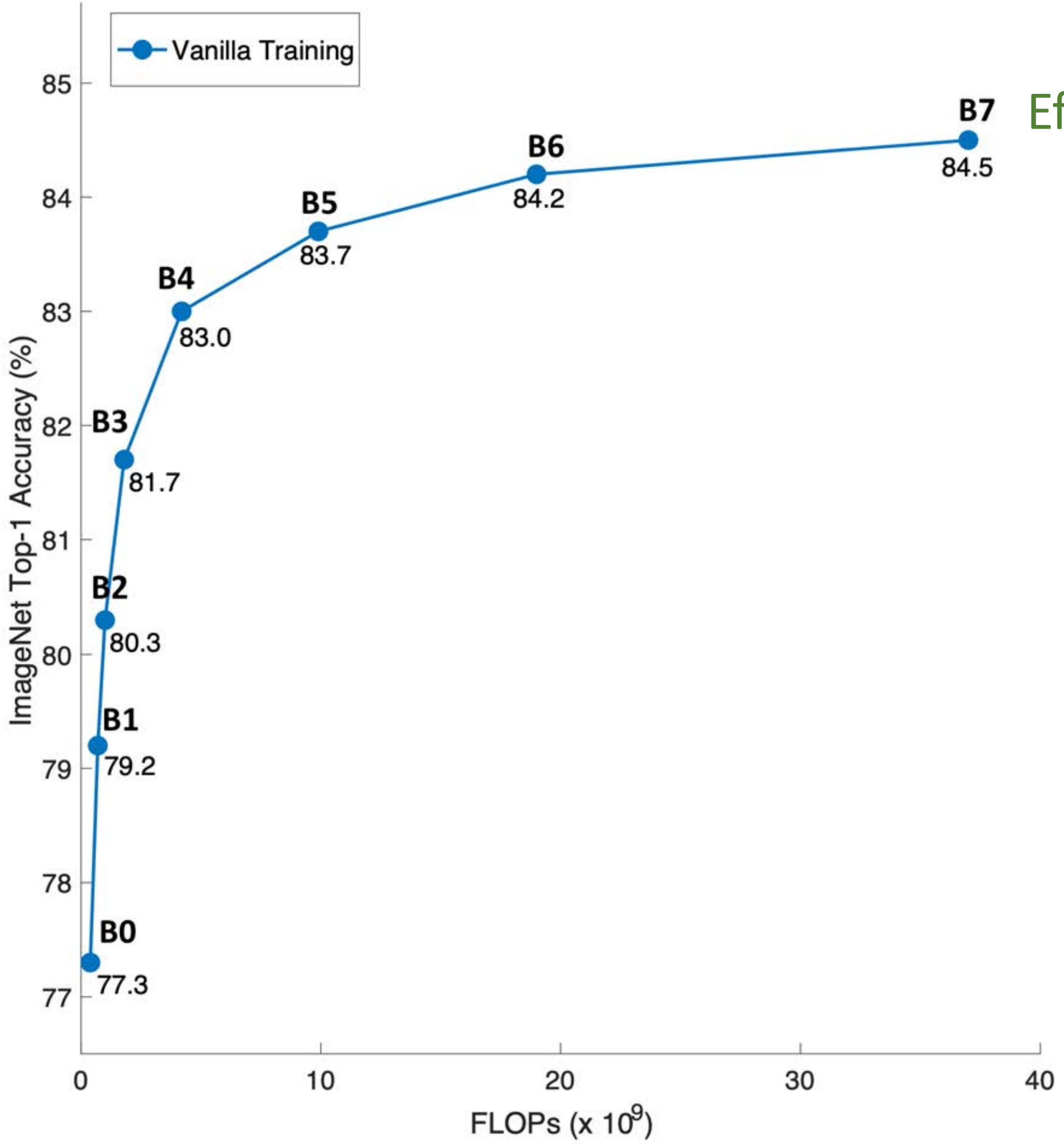
We already know three important scaling factors

Backbone --- EfficientNet



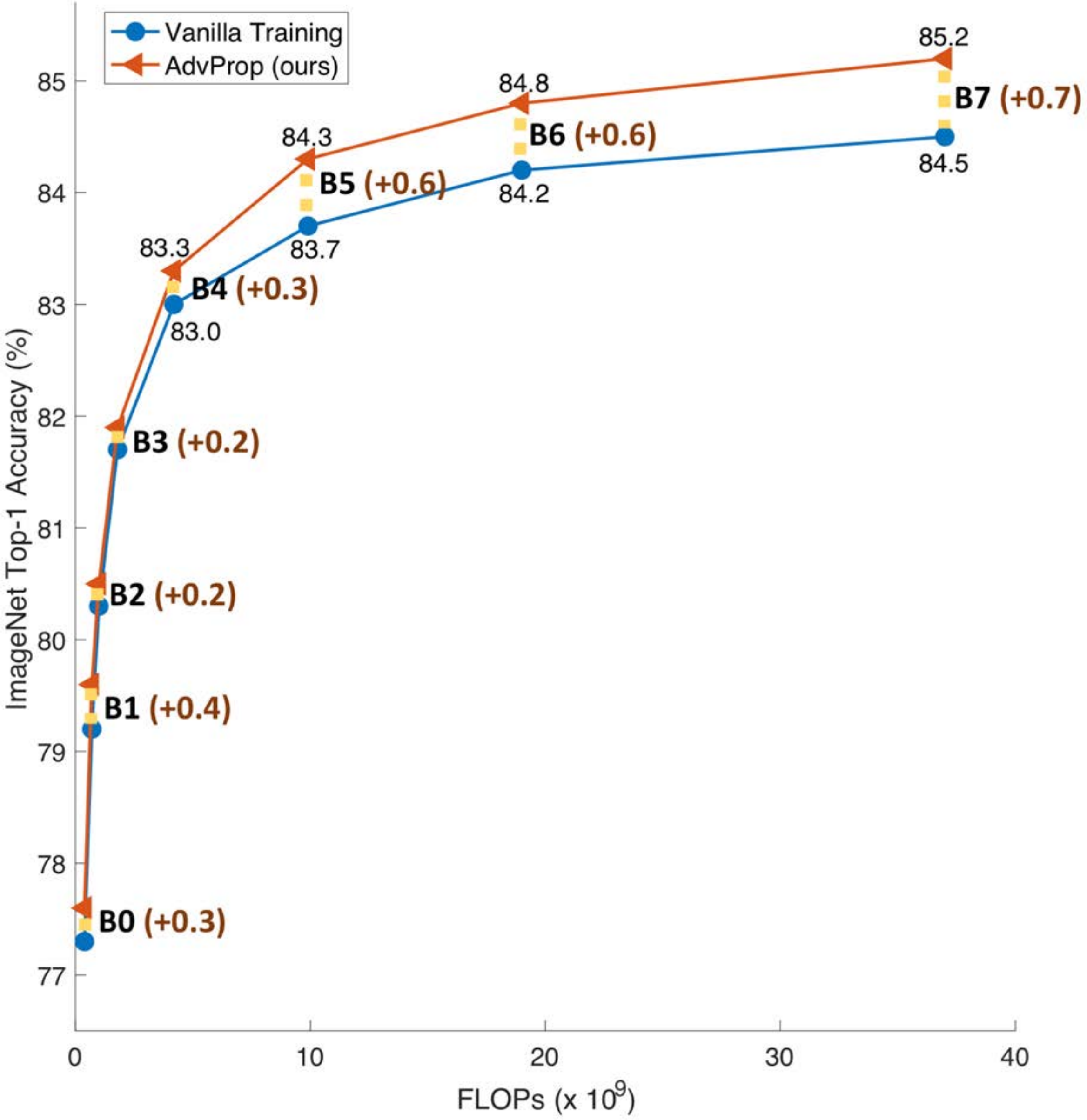
A Better Scaling-Up Policy

Results on ImageNet



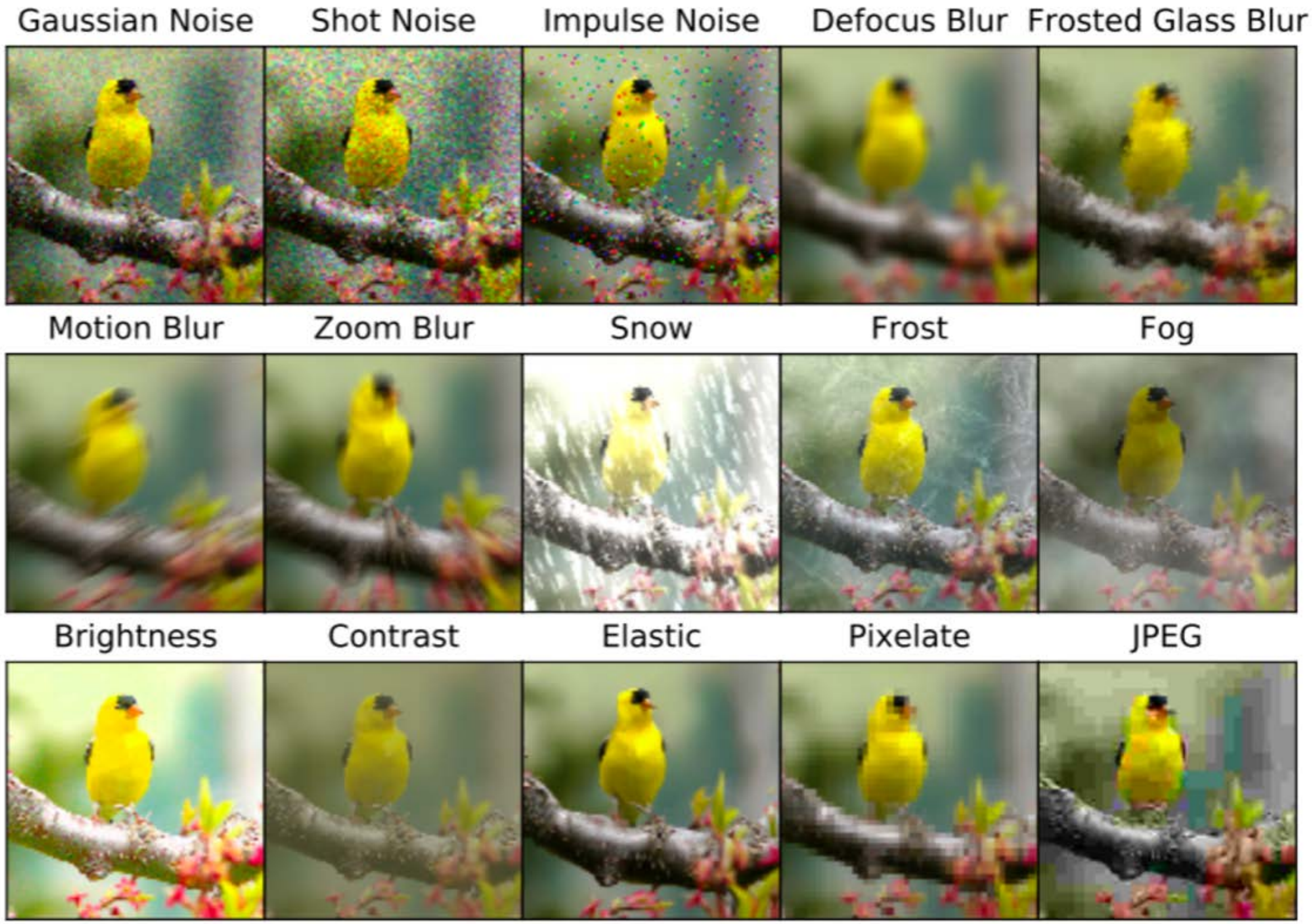
EfficientNet-B7's 84.5% top-1 accuracy on ImageNet is the previous SOTA

Results on ImageNet



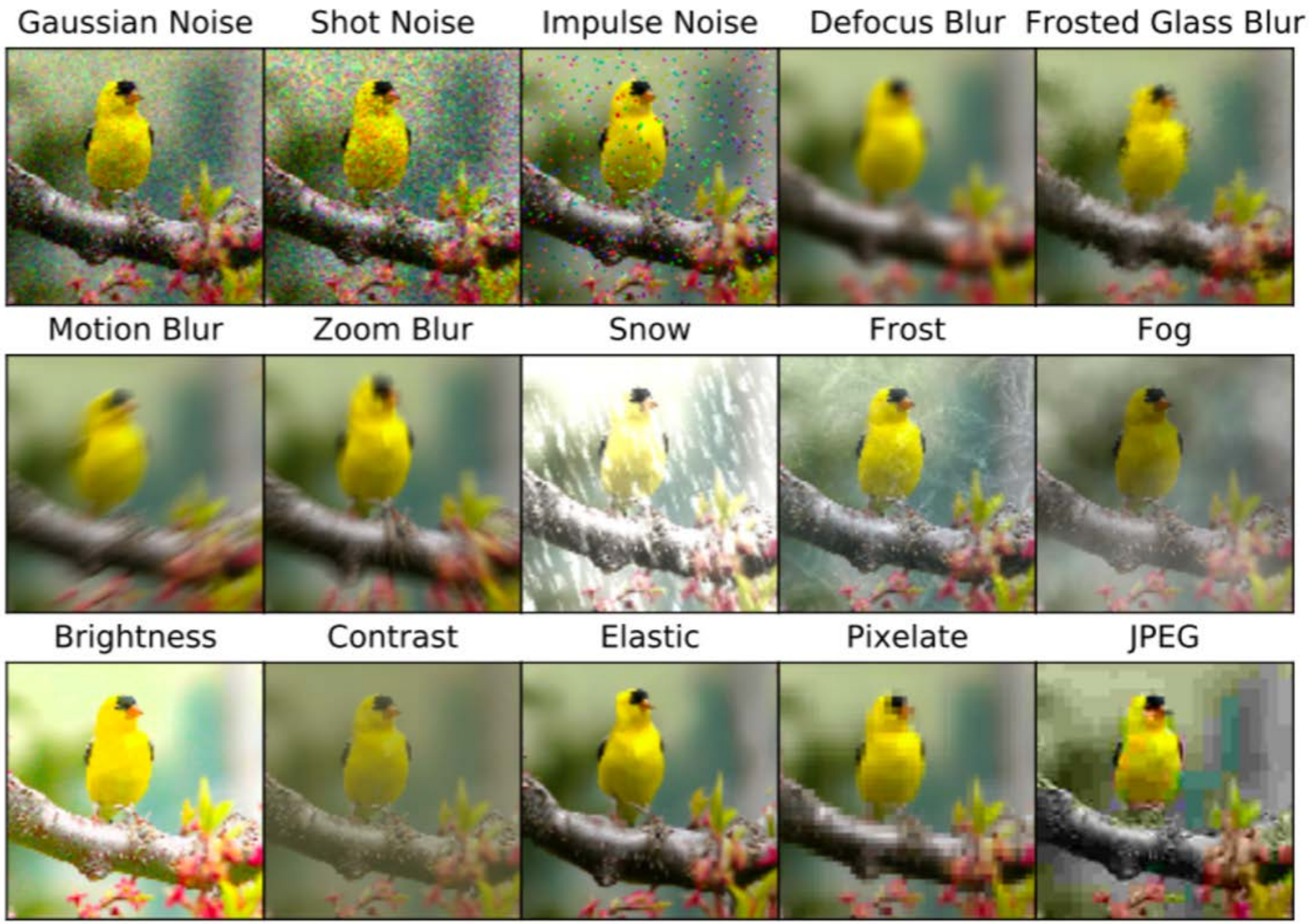
AdvProp improves EfficientNet-B7's top-1 accuracy by 0.7%

Results on ImageNet-C



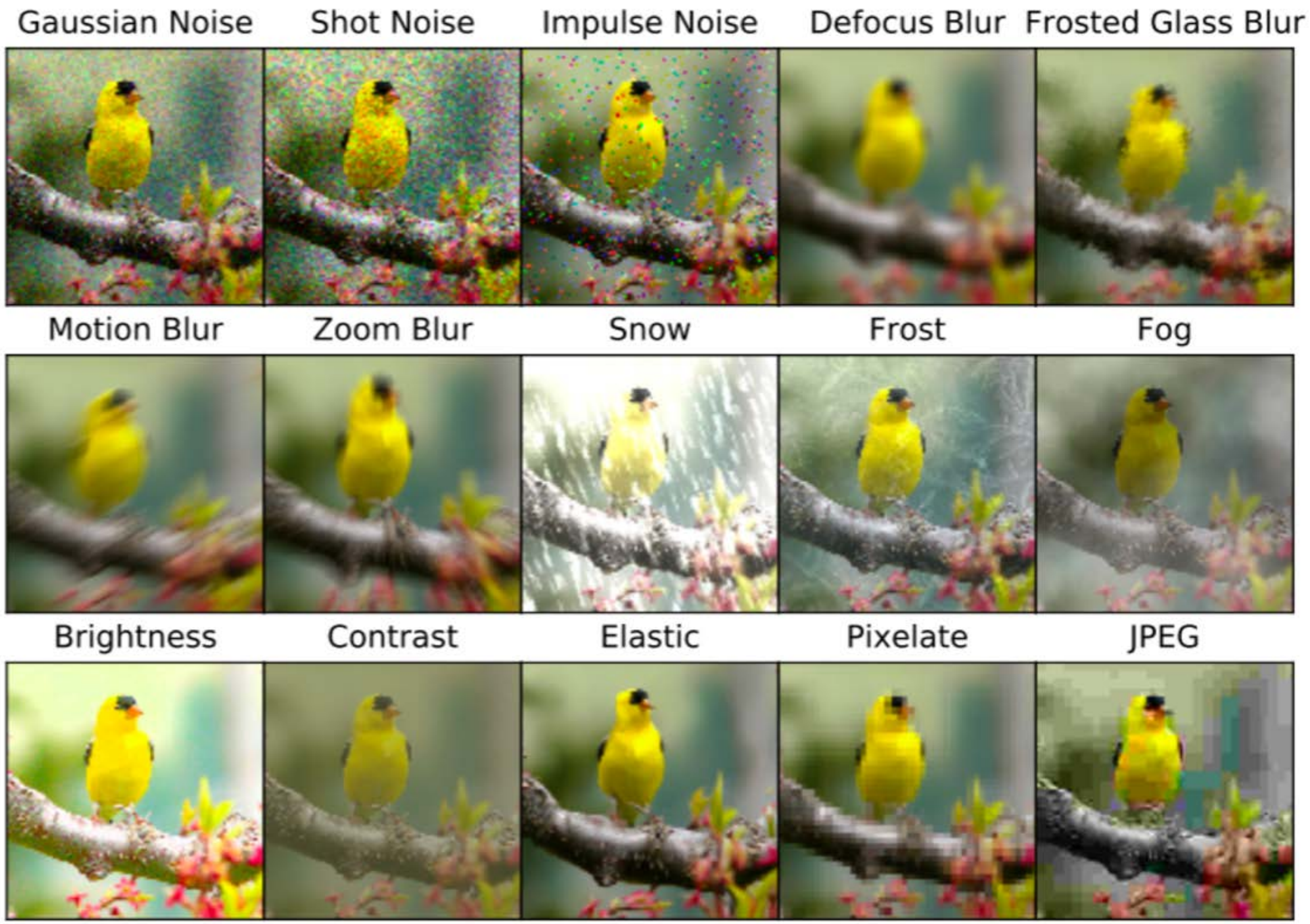
Networks	Mean Corruption Error ↓
EfficientNet-B7	59.4%

Results on ImageNet-C



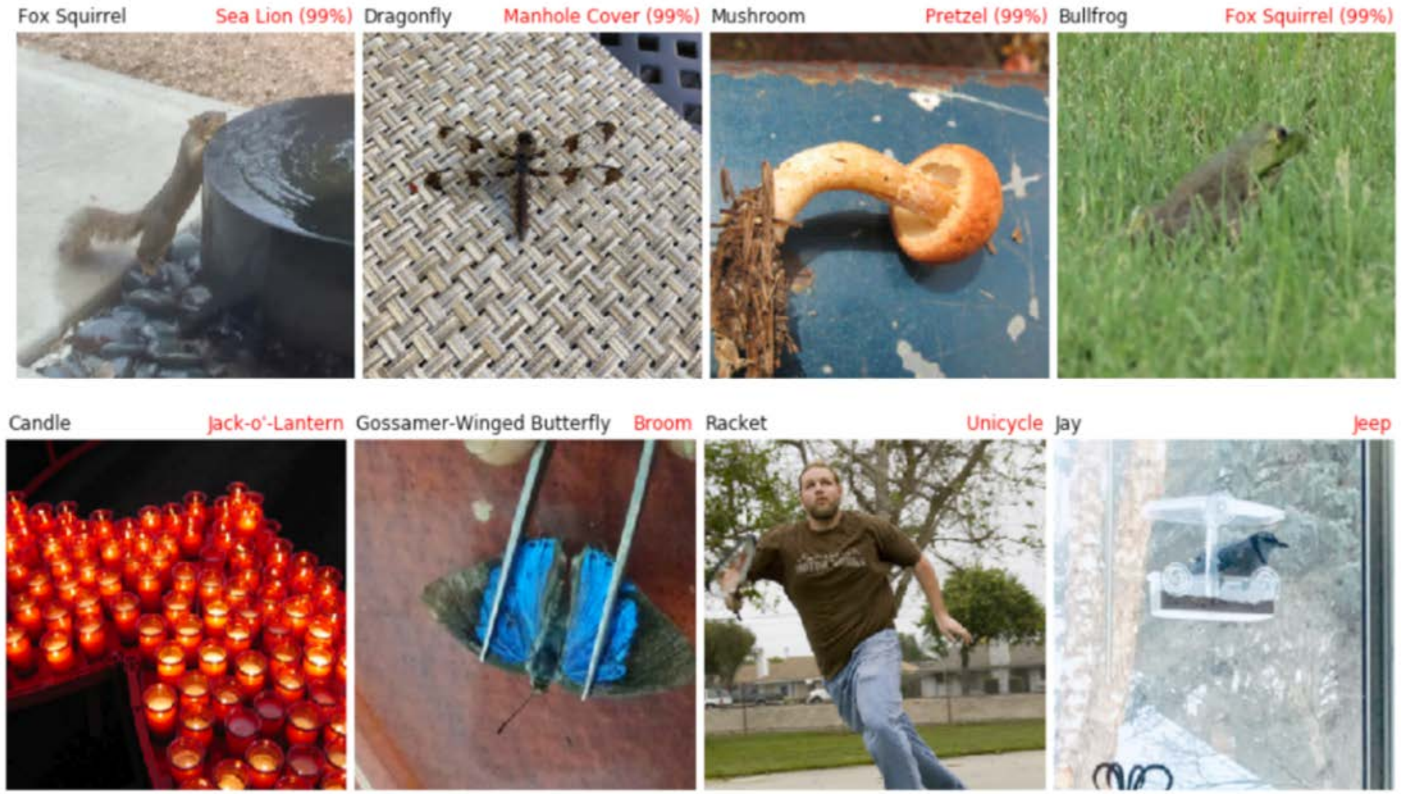
Networks	Mean Corruption Error ↓
EfficientNet-B7	59.4%
+ AdvProp	**52.9% (-6.5%)**

Results on ImageNet-C



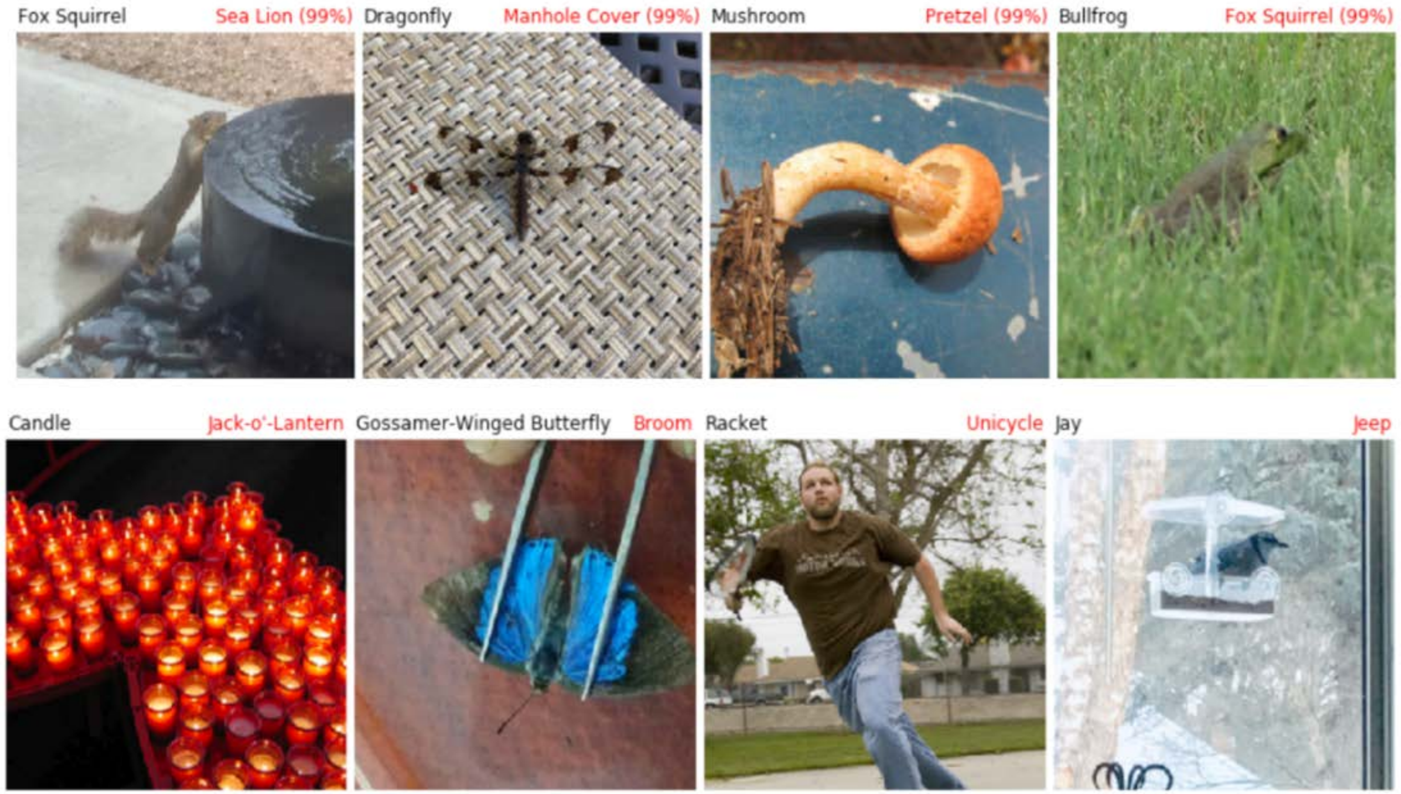
Networks	Mean Corruption Error ↓
EfficientNet-B7	59.4%
+ AdvProp	**52.9% (-6.5%)**
ResNet-50	**74.8%**

Results on ImageNet-A



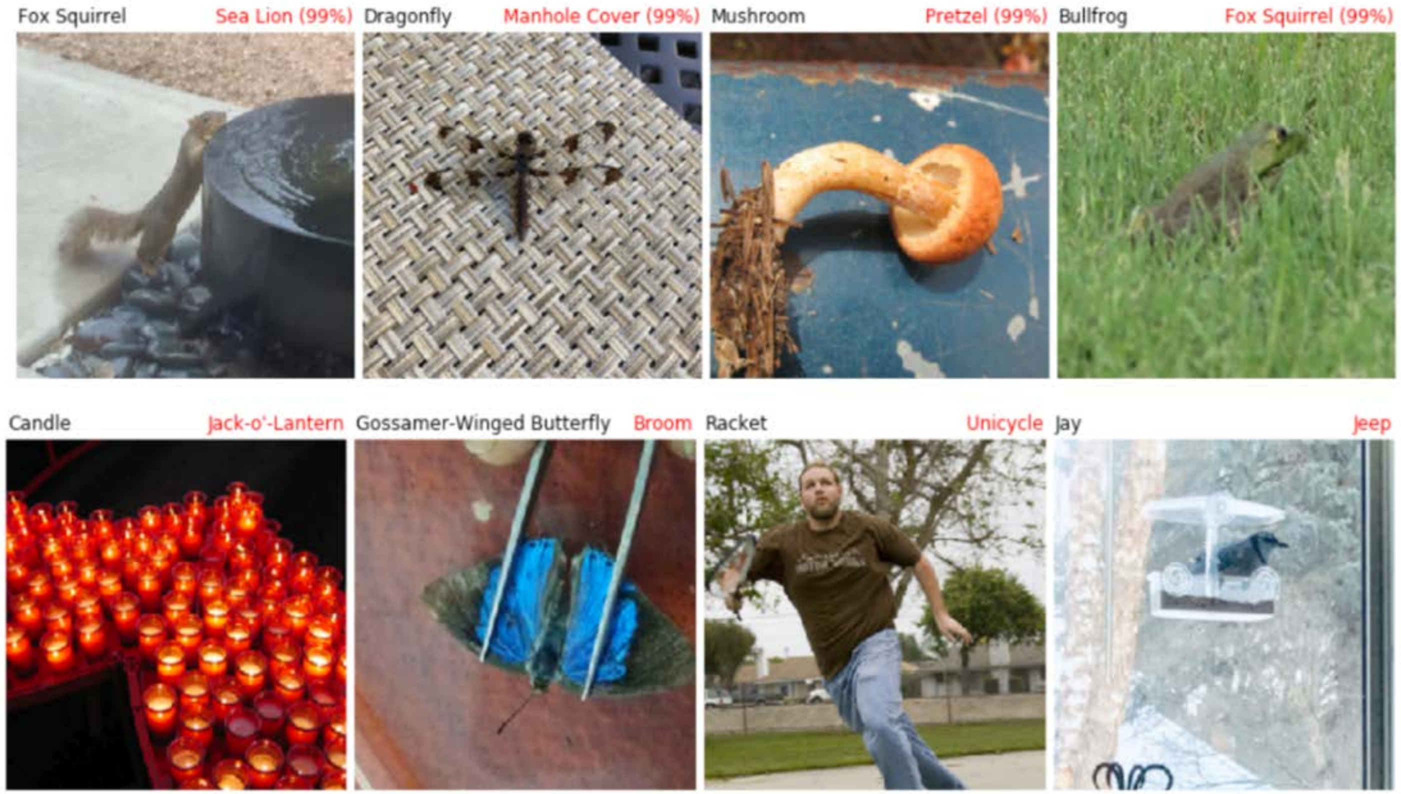
Networks	Top-1 Accuracy ↑
EfficientNet-B7	37.7%

Results on ImageNet-A



Networks	Top-1 Accuracy ↑
EfficientNet-B7	37.7%
+ AdvProp	**44.7% (+7.0%)**

Results on ImageNet-A

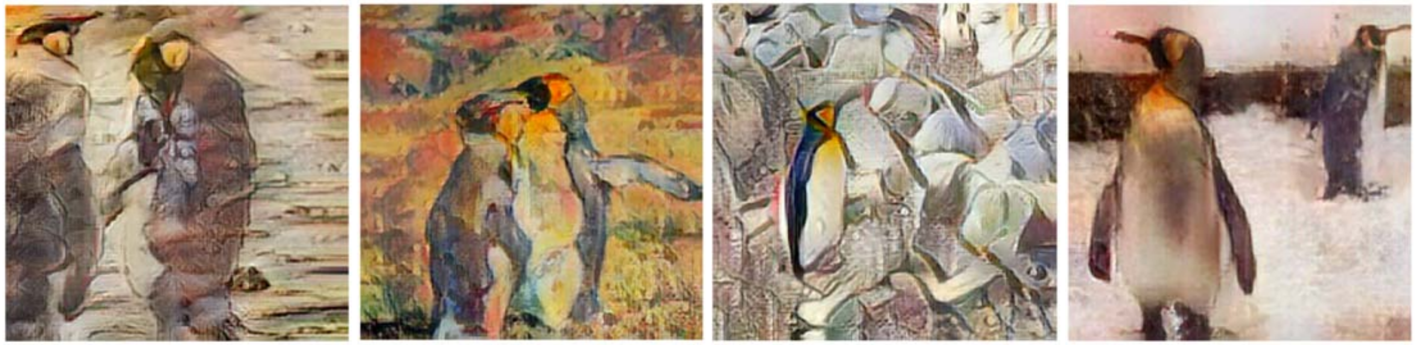


Networks	Top-1 Accuracy ↑
EfficientNet-B7	37.7%
+ AdvProp	**44.7% (+7.0%)**
ResNet-50	**3.1%**

Results on Stylized-ImageNet

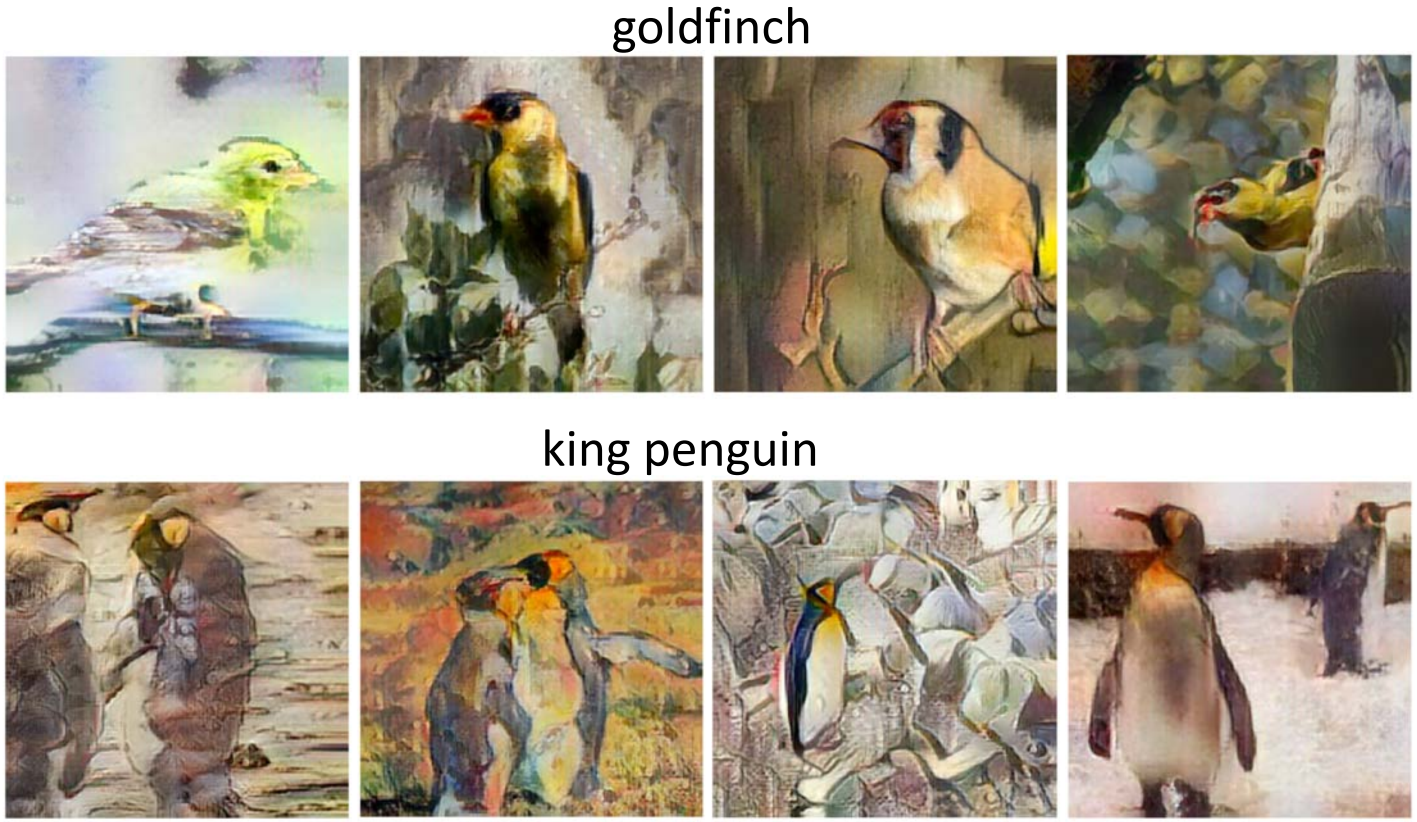
goldfinch

king penguin



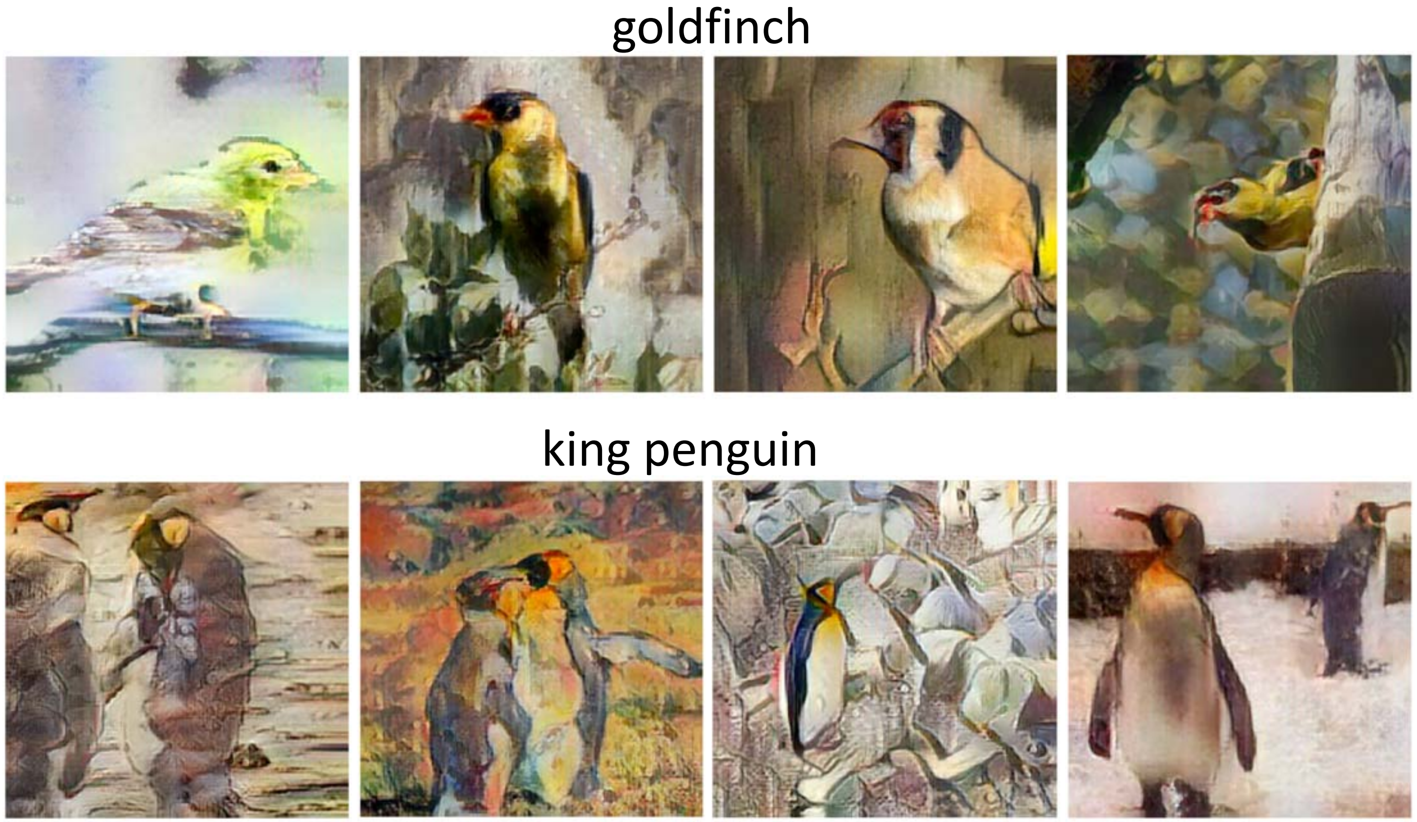
Networks	Top-1 Accuracy
EfficientNet-B7	21.8%

Results on Stylized-ImageNet



Networks	Top-1 Accuracy ↑
EfficientNet-B7	21.8%
+ AdvProp	**26.6% (+4.8%)**

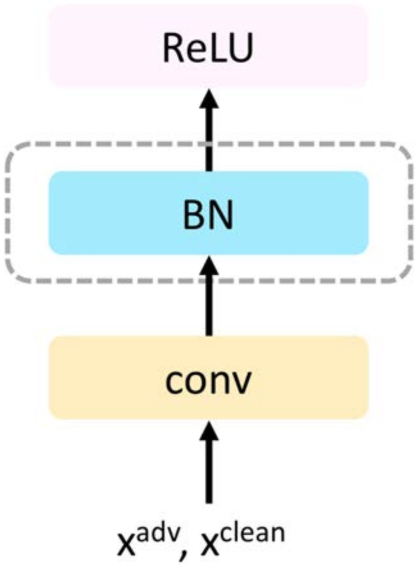
Results on Stylized-ImageNet



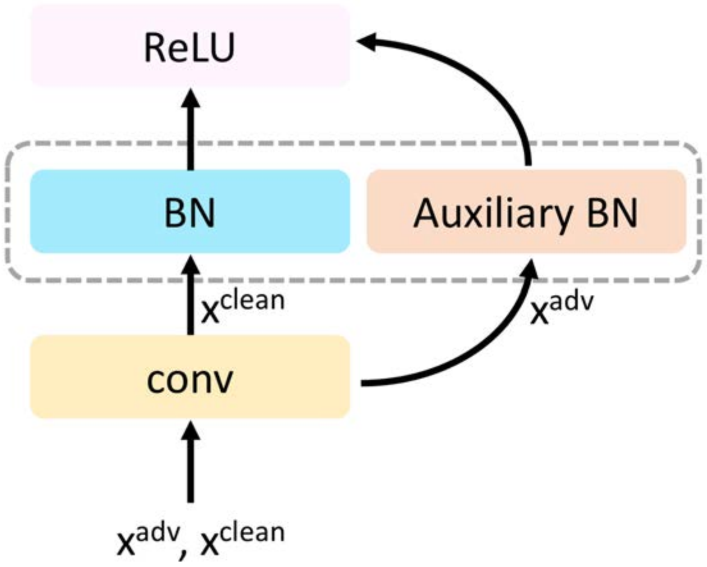
Networks	Top-1 Accuracy
EfficientNet-B7	21.8%
+ AdvProp	**26.6% (+4.8%)**
ResNet-50	**8.0%**

Ablation --- Comparison to Adversarial Training

How important that we should train with **distinction**

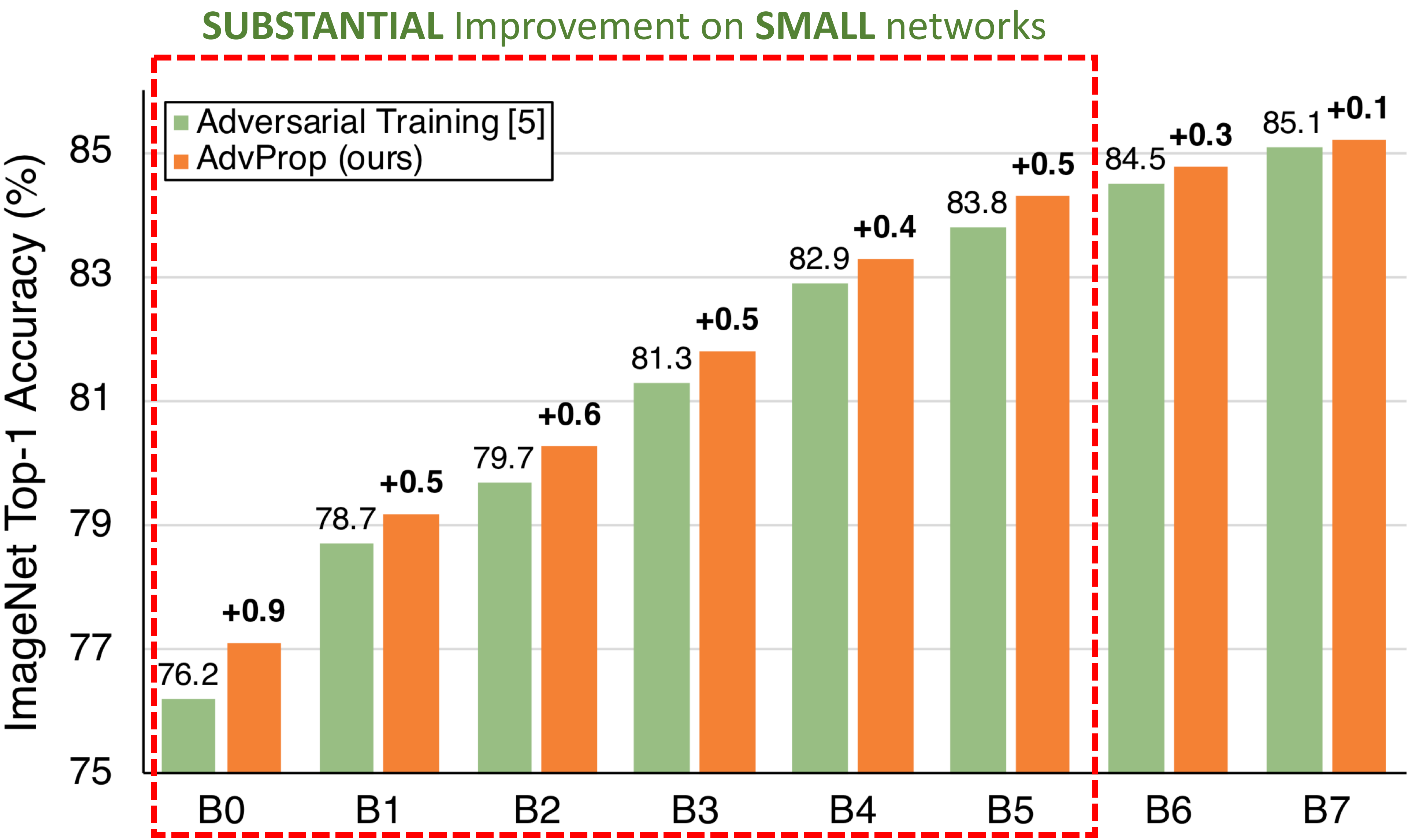


Adv. Training



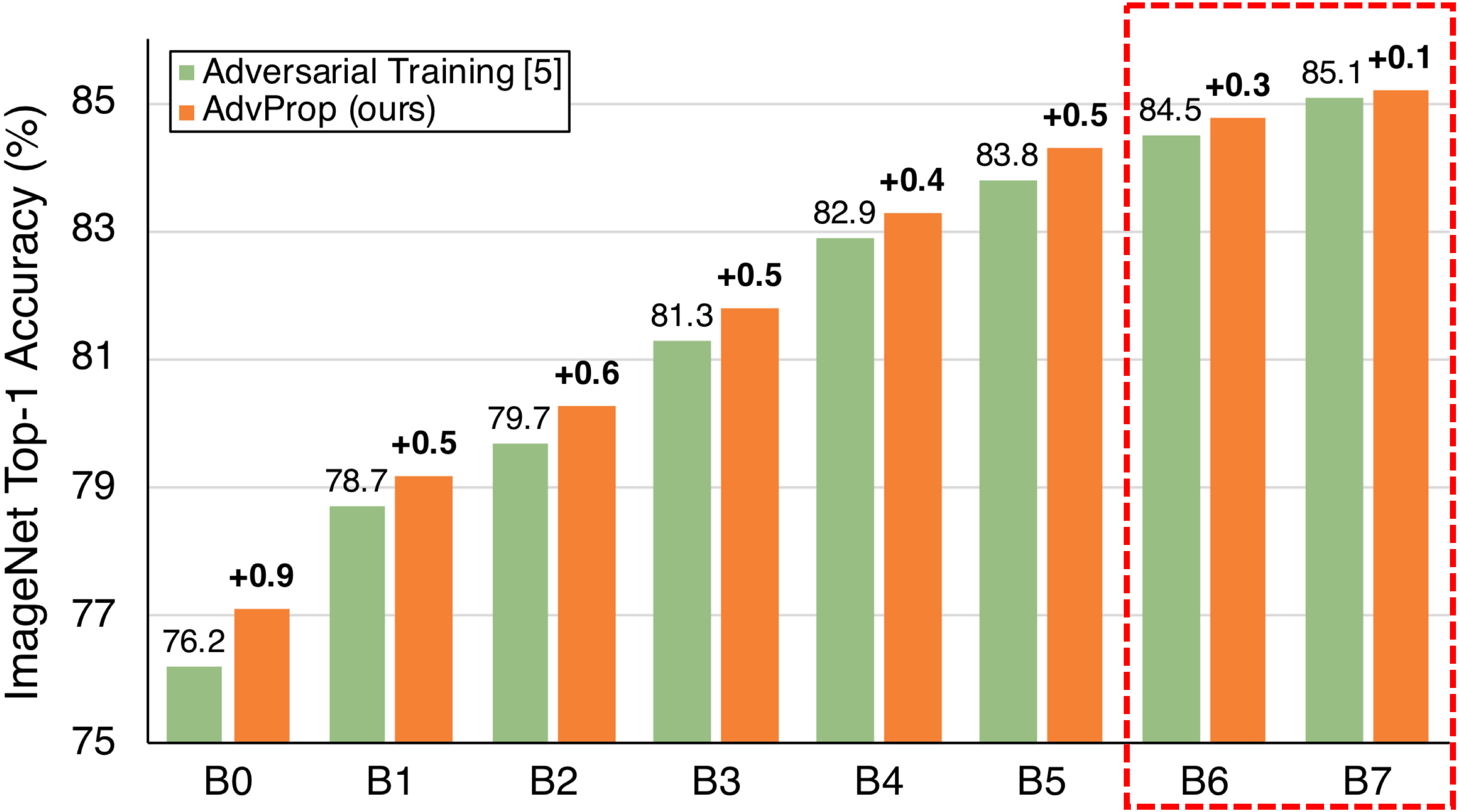
Advprop

Ablation --- Comparison to Adversarial Training



Ablation --- Comparison to Adversarial Training

Improvement becomes smaller on **LARGER** networks



Ablation --- Comparison to Adversarial Training

AdvProp demonstrates **ADVANTAGES** over Adversarial Training

Ablation --- Comparison to Adversarial Training

AdvProp demonstrates **ADVANTAGES** over Adversarial Training

- **AdvProp helps large models to generalize better**

Model	ImageNet-C [7]	ImageNet-A [8]	Stylized-ImageNet [4]
	mCE ↓	Top-1 Acc. ↑	Top-1 Acc. ↑
B6 + Adv. Training	55.8	37.0	24.7
B6 + AdvProp (**ours**)	**53.6**	**40.6**	**25.9**
B7 + Adv. Training	56.0	40.4	25.1
B7 + AdvProp (**ours**)	**52.9**	**44.7**	**26.6**

Ablation --- Comparison to Adversarial Training

AdvProp demonstrates **ADVANTAGES** over Adversarial Training

- AdvProp helps large models to generalize better

Model	ImageNet-C [7]	ImageNet-A [8]	Stylized-ImageNet [4]
	mCE ↓	Top-1 Acc. ↑	Top-1 Acc. ↑
B6 + Adv. Training	55.8	37.0	24.7
B6 + AdvProp (**ours**)	**53.6**	**40.6**	**25.9**
B7 + Adv. Training	56.0	40.4	25.1
B7 + AdvProp (**ours**)	**52.9**	**44.7**	**26.6**

- **AdvProp is more general to other network architectures**

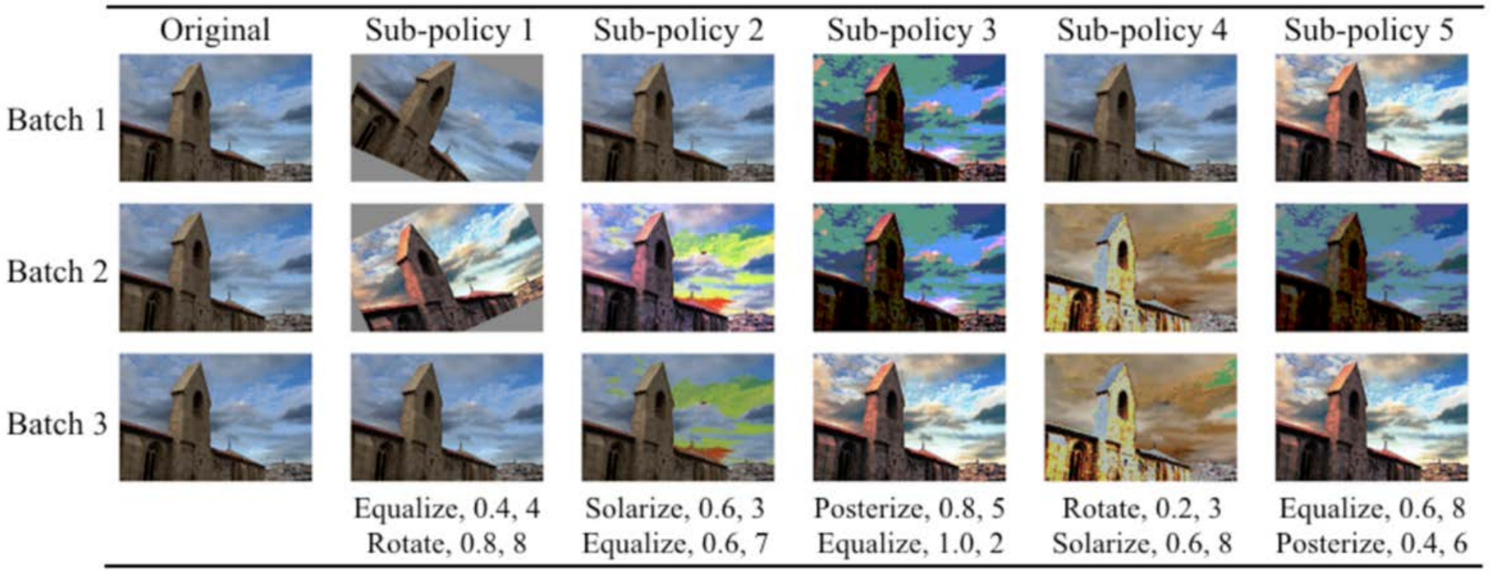
	ResNet-50	ResNet-101	ResNet-152	ResNet-200
Vanilla Training	76.7	78.3	79.0	79.3
Adversarial Training	-3.2	-1.8	-2.0	-1.4
AdvProp (**ours**)	**+0.4**	**+0.6**	**+0.8**	**+0.8**

Ablation --- Fine-Grained AdvProp

Core idea: maintain **SEPARATE BNs** for different distributions

Ablation --- Fine-Grained AdvProp

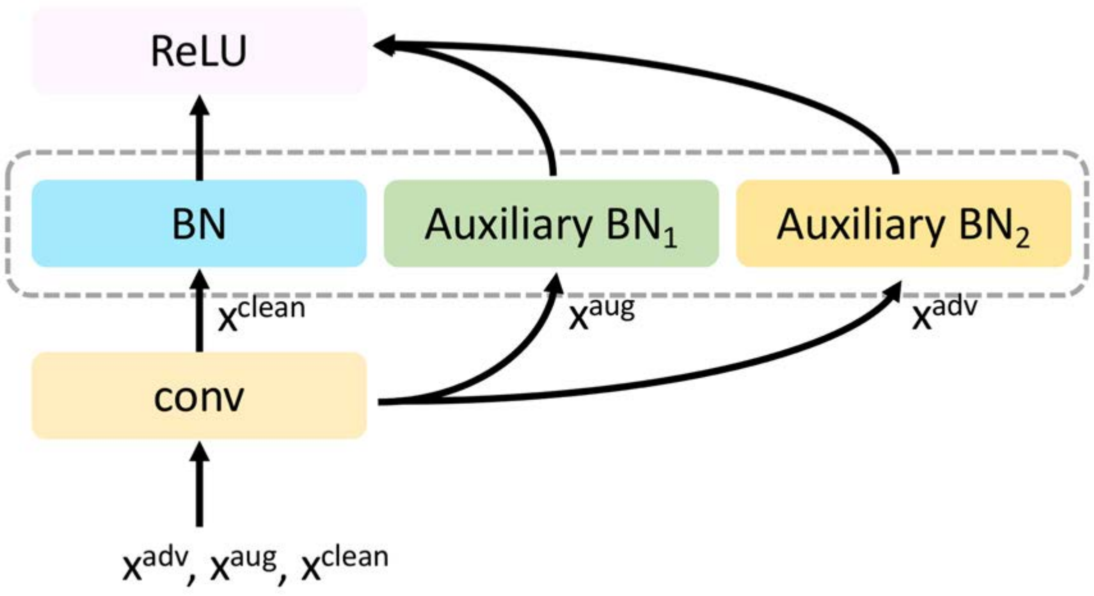
Core idea: maintain **SEPARATE BNs** for different distributions



Augmentation policy may produce a different distribution to clean images?

Ablation --- Fine-Grained AdvProp

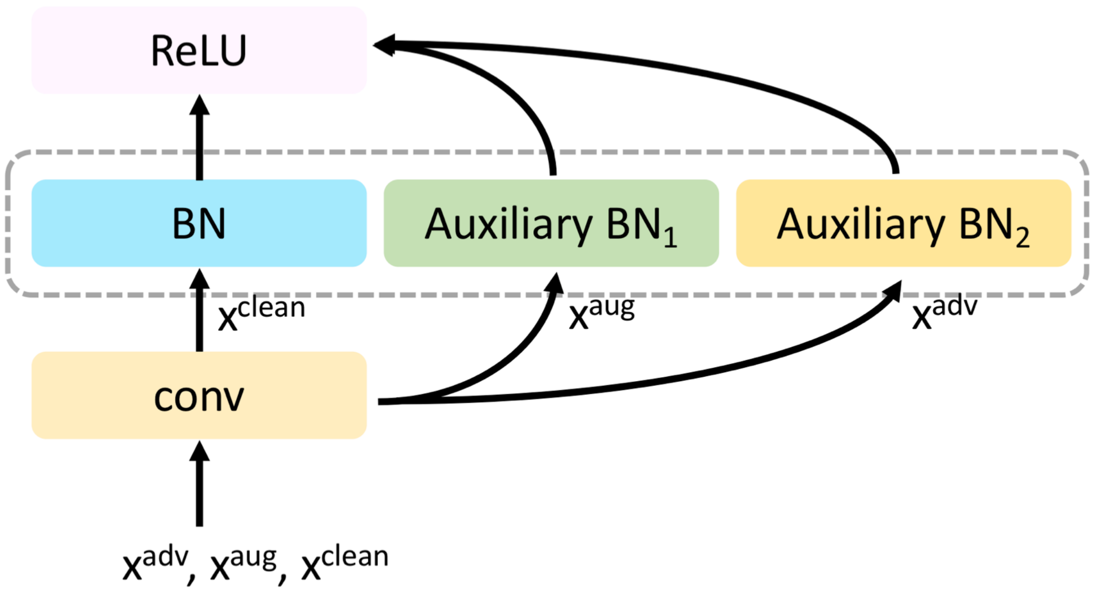
Core idea: maintain **SEPARATE BNs** for different distributions



Fine-Grained AdvProp

Ablation --- Fine-Grained AdvProp

Core idea: maintain **SEPARATE BNs** for different distributions



Fine-Grained AdvProp

	B0	B1	B2	B3	B4	B5	B6	B7
AdvProp	77.6	79.6	80.5	81.9	83.3	84.3	**84.8**	**85.2**
Fine-Grained AdvProp	**77.9**	**79.8**	**80.7**	**82.0**	**83.5**	**84.4**	84.8	85.2

Ablation --- New SOTA on ImageNet without extra data

~10X LESS parameters

~3,000X LESS training data

BETTER performance

BUT new SOTA on ImageNet

	# Params	Extra Data	Top-1 Acc.
EfficientNet-B8 + AdvProp	88M	✗	**85.5%**
ResNeXt-101 32x48d [20]	829M	3000× more	85.4%

Questions?