



南京航空航天大学
Nanjing University of Aeronautics and Astronautics

Practical No-box Adversarial Attacks against DNNs

Qizhang Li *

ByteDance AI Lab

liqizhang@bytedance.com

Yiwen Guo †

ByteDance AI Lab

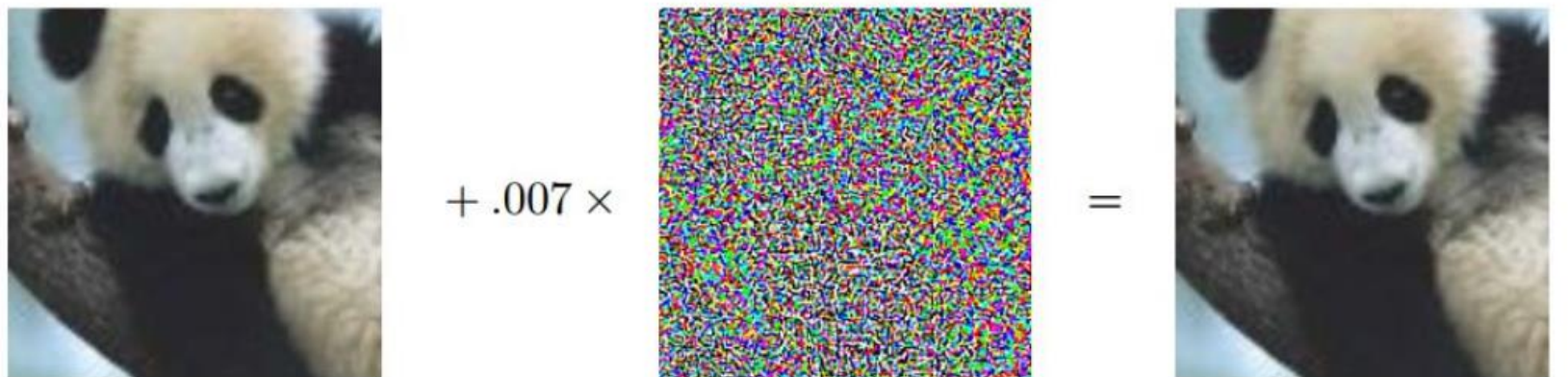
guoyiwen.ai@bytedance.com

Hao Chen

University of California, Davis

chen@ucdavis.edu

NIPS 2020



The diagram illustrates an adversarial attack on a panda image. It shows the original image x , a noise perturbation, and the resulting adversarial image.

Original image x : “panda”
57.7% confidence

+ .007 \times $\text{sign}(\nabla_x J(\theta, x, y))$: “nematode”
8.2% confidence

= $x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$: “gibbon”
99.3 % confidence

➤ Adversarial Attack:

Find a new input (similar to original input) but classified as another class t (untargeted or targeted)

➤ Adversarial Attacks:

✓ White-box Attacks:

- The attacker **knows the detailed information** about the victim model (model architecture, parameters, and class probabilities).
- Success rate of white-box attack reaches almost 100%.

✓ Black-box Attacks:

- The attacker cannot access the architecture, the parameters, or the training data of the victim model.
- **Only query the victim model.**

✓ No-box Attacks:

- ✓ The attacker can neither access the model information or the training set nor query the model.
- ✓ The attacker can only gather a small number of examples from the same problem domain as that of the victim model.

➤ White-box Adversarial Attacks:

✓ Fast-gradient sign method (FGSM):

- Take a step in the direction of the gradient of the loss function:

$$x^* = x + \epsilon \cdot \text{sign}(\nabla \text{loss}_F(x))$$

- This is simple and good performance.

✓ Iterative FGSM (I-FGSM):

- Update version of the FGSM.
- Instead of changing the amount of ϵ , a smaller amount of α is used.
- Clipped by the same ϵ :

$$x_i^* = x_{i-1}^* + \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla \text{loss}_F(x_{i-1}^*)))$$

Gradient Descent

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

➤ Black-box Adversarial Attacks:

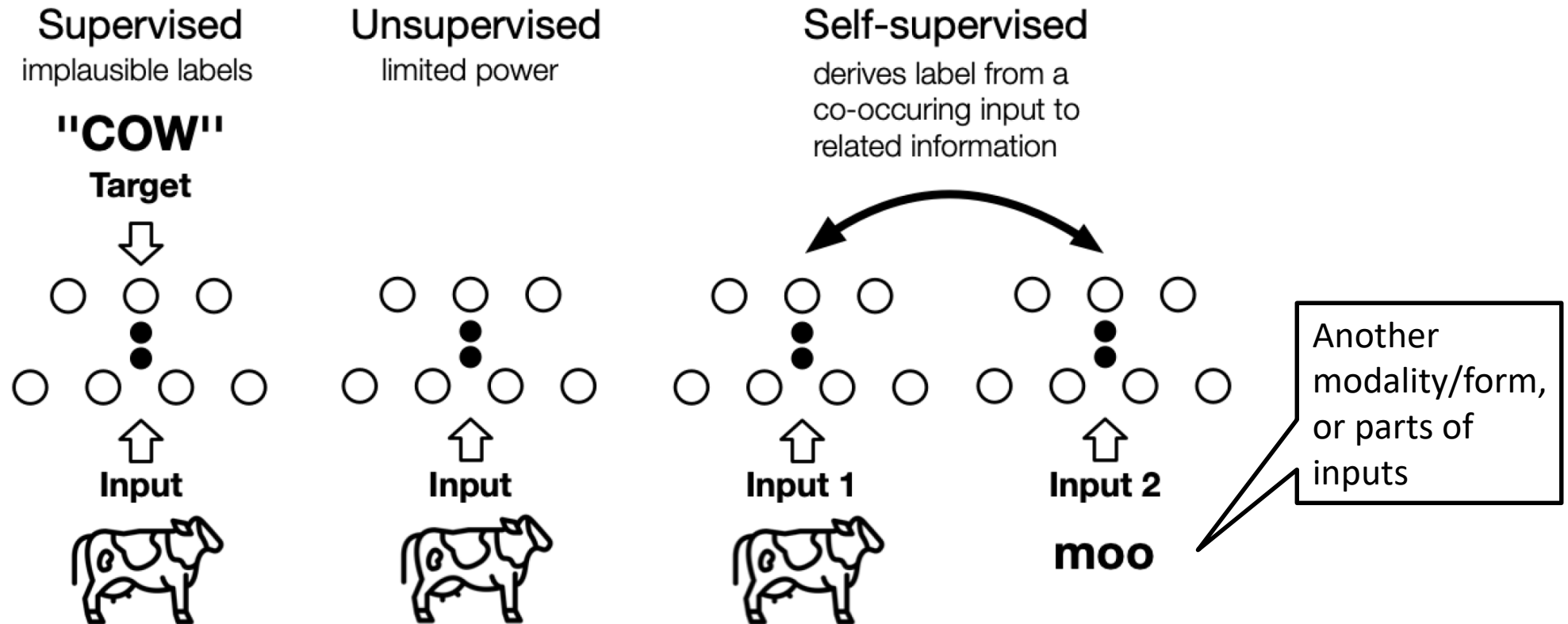
✓ Transfer attacks

- Adversarial examples often transfer between different models.
- Train an **alternative model** and perform white-box attacks on it to generate adversarial examples.
- **Suffer from transfer loss.**
- Use **one query** to the target model for each attempted candidate transfer.

✓ Optimization attacks

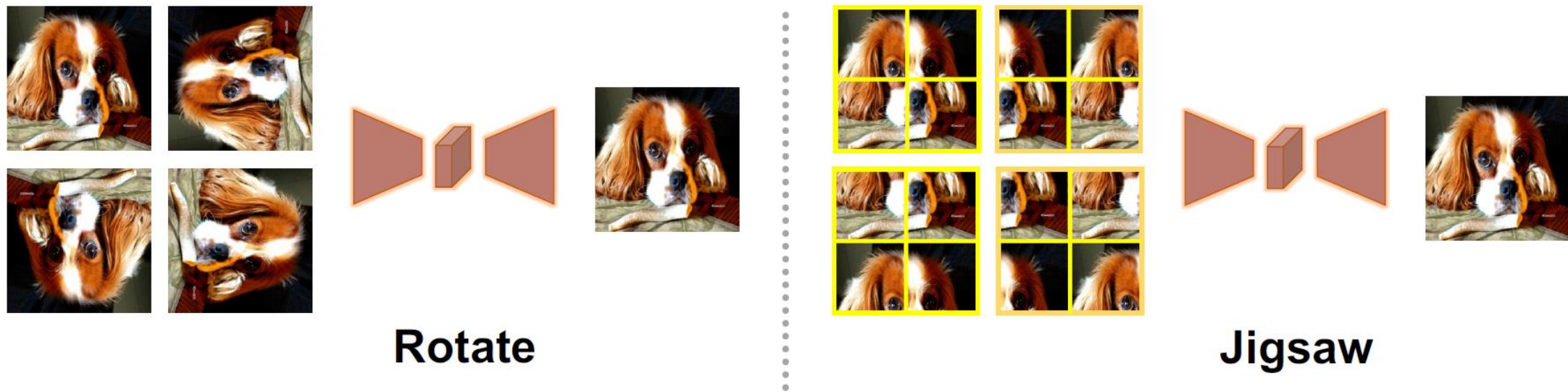
- Formulate the attack goal as a black-box **optimization problem**.
- Can be divided into three types: **Gradient Attacks**, **Gradient-free Attacks**, **Restricted Black-box Attacks**.
- **Do not suffer from transfer loss.**
- Often require **many queries** for each attempted candidate transfer.

➤ Self-supervised learning (SSL) :



SSL exploits **surrogate supervision** from unlabeled data for gaining **high-level data understanding** of data.

➤ Self-supervised learning (SSL) :



The core of self-supervised learning is how to **automatically generate labels** for data.

Consider the **No-box scenario** where the attacker can gather a dataset with **very limited size**.



What comes uppermost in mind is to utilize the **transferability** of adversarial examples.



However, current supervised learning for DNNs require **large-scale** training to generalize.



To achieve the goal, one should first develop proper **training mechanisms** and “**substitute architectures**”.

➤ Auto-encoder:

- ✓ Such a model is capable of capturing low-level image representations **without** suffering from severe **over-fitting**.
- ✓ **Discriminative ability is by no means entailed** and thus adversarial examples crafted against the model are difficult to transfer to the victim models.

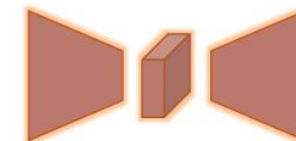
$$L = \frac{1}{n} \sum_{i=0}^{n-1} \|\text{Dec}(\text{Enc}(x_i)) - x_i\|^2$$

➤ Reconstruction from chaos :

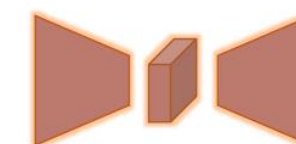
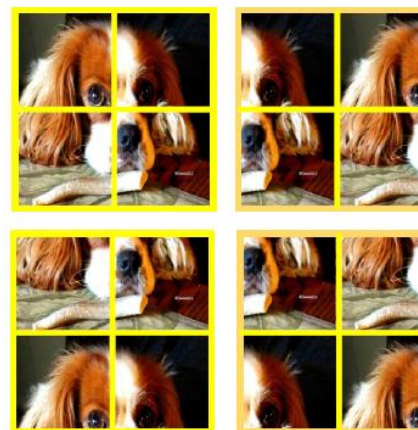
- ✓ Two tasks:
 - The front view of each **rotated** images.
 - The prefect fit of each possible **jigsaw puzzle**.
- ✓ Their learning objectives are commonly formulated as:

$$L_{\text{rotation/jigsaw}} = \frac{1}{n} \sum_{i=0}^{n-1} \|\text{Dec}(\text{Enc}(T(x_i))) - x_i\|^2$$

$T(\cdot)$ is designed to rotate images or shuffle image patches



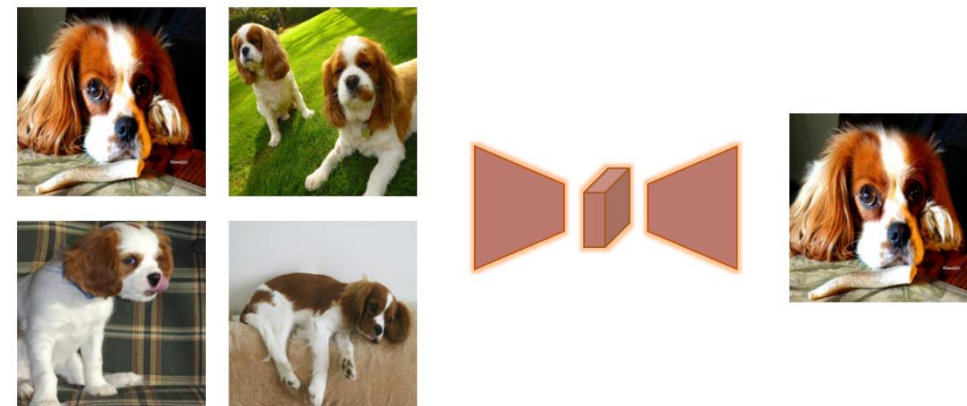
Rotate



Jigsaw

➤ Prototypical image reconstruction:

- ✓ Encourage the model to reconstruct **class-specific** prototypes.
- ✓ The learning objective is commonly formulated as:



Prototypical

$$L_{\text{prototypical}} = \frac{1}{n} \sum_{i=0}^{n-1} \left((1 - y_i) \left\| \text{Dec}(\text{Enc}(x_i)) - x^{(0)} \right\|^2 + y_i \left\| \text{Dec}(\text{Enc}(x_i)) - x^{(1)} \right\|^2 \right)$$

In which: $x^{(0)} \in \{x_i | y_i = 0\}$ and $x^{(1)} \in \{x_i | y_i = 1\}$ are randomly chosen image prototypes.



It is possible to introduce more than one decoder with this mechanism, by sampling **multiple pairs of image prototypes** from the two classes.

➤ Making predictions:

✓ Single-decoder:

$$\hat{y} = \arg \min_{y \in \{0,1\}} \left\| \text{Dec}(\text{Enc}(x)) - x^{(y)} \right\|$$

✓ Multiple-decoder:

$$\hat{y} = \arg \min_{y \in \{0,1\}} \frac{1}{K} \sum_{k=0}^{K-1} \left\| \text{Dec}_k(\text{Enc}(x)) - x_k^{(y)} \right\|$$

- The learning objective is finally formulated as :

$$L_{\text{adversarial}} = -\log p(y_i|x_i) \quad \text{where} \quad p(y_i|x_i) = \frac{\exp(-\lambda \|\text{Dec}(\text{Enc}(x_i)) - \tilde{x}_i\|^2)}{\sum_j \exp(-\lambda \|\text{Dec}(\text{Enc}(x_i)) - \tilde{x}_j\|^2)}$$

In which:

- $\lambda > 0$ is a scaling parameter
- $\tilde{x}_i := x_i$ for reconstructing images from **rotations** and **jigsaw puzzles**.
- $\tilde{x}_i := x^{(0)}$ for the **prototypical reconstruction** models with x_i labeling as $y_i = 0$.

➤ I-FGSM + ILA:

- ✓ I-FGSM (gradient-based baseline attacks)

$$x_i^* = x_{i-1}^* - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(x_{i-1}^*)))$$

- ✓ ILA (enhancing adversarial example transferability)

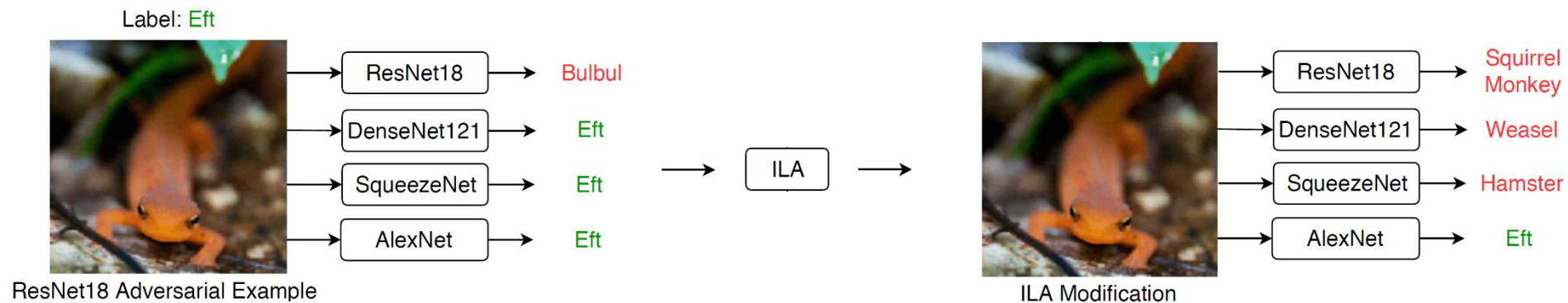


Table 1: Compare the transferability of adversarial examples crafted on different models on ImageNet. The prediction accuracy on adversarial examples under $\epsilon = 0.1$ are shown (lower is better).

Method	Sup.	VGG-19 [42]	Inception v3 [45]	ResNet [15]	DenseNet [17]	SENet [16]	WRN [56]	PNASNet [28]	MobileNet v2 [39]	Average
Naïve [‡]	✗	45.92%	63.94%	60.64%	56.48%	65.54%	58.80%	73.14%	37.76%	57.78%
Jigsaw	✗	31.54%	50.28%	46.24%	42.38%	59.06%	51.24%	62.32%	25.24%	46.04%
Rotation	✗	31.14%	48.14%	47.40 %	41.26%	58.20%	50.72%	59.94%	26.00%	45.35%
Naïve [†]	✓	76.20%	80.86%	83.76%	78.94%	87.00%	84.16%	86.96%	72.44%	81.29%
Prototypical	✓	19.78%	36.46%	37.92%	29.16%	44.56%	37.28%	48.58%	17.78%	33.94%
Prototypical*	✓	18.74%	33.68%	34.72%	26.06%	42.36%	33.14%	45.02%	16.34%	31.26%
Beyonder	✓	24.96%	51.12%	30.30%	27.12%	43.78%	33.94%	51.80%	27.02%	36.26%

* The prototypical models with multiple decoders. To be more specific, 20 decoders are introduced in each model.

Table 2: Compare the transferability of adversarial examples crafted on different models on ImageNet. The prediction accuracy on adversarial examples under $\epsilon = 0.08$ are shown (lower is better).

Method	Sup.	VGG-19 [7]	Inception v3 [8]	ResNet [1]	DenseNet [3]	SENet [2]	WRN [9]	PNASNet [4]	MobileNet v2 [5]	Average
Naïve [‡]	✗	53.92%	70.18%	68.16%	63.98%	72.48%	66.66%	78.28%	47.38%	65.13%
Jigsaw	✗	40.00%	58.20%	55.66%	50.30%	66.62%	59.52%	70.36%	34.60%	54.41%
Rotation	✗	38.88%	56.16%	57.06%	49.56%	65.30%	58.14%	67.70%	34.64%	53.43%
Naïve [†]	✓	76.64%	81.24%	83.98%	79.54%	87.14%	84.30%	87.12%	73.16%	81.64%
Prototypical	✓	30.80%	49.28%	50.56%	40.30%	56.58%	48.88%	60.94%	28.50%	45.73%
Prototypical*	✓	30.08%	45.74%	47.28%	37.66%	54.42%	44.82%	57.58%	27.32%	43.11%
Beyond	✓	27.70%	53.58%	33.74%	30.58%	46.70%	37.26%	54.92%	29.42%	39.24%

* The prototypical models with multiple decoders. To be more specific, 20 decoders are introduced in each model.

Experiment-image classification

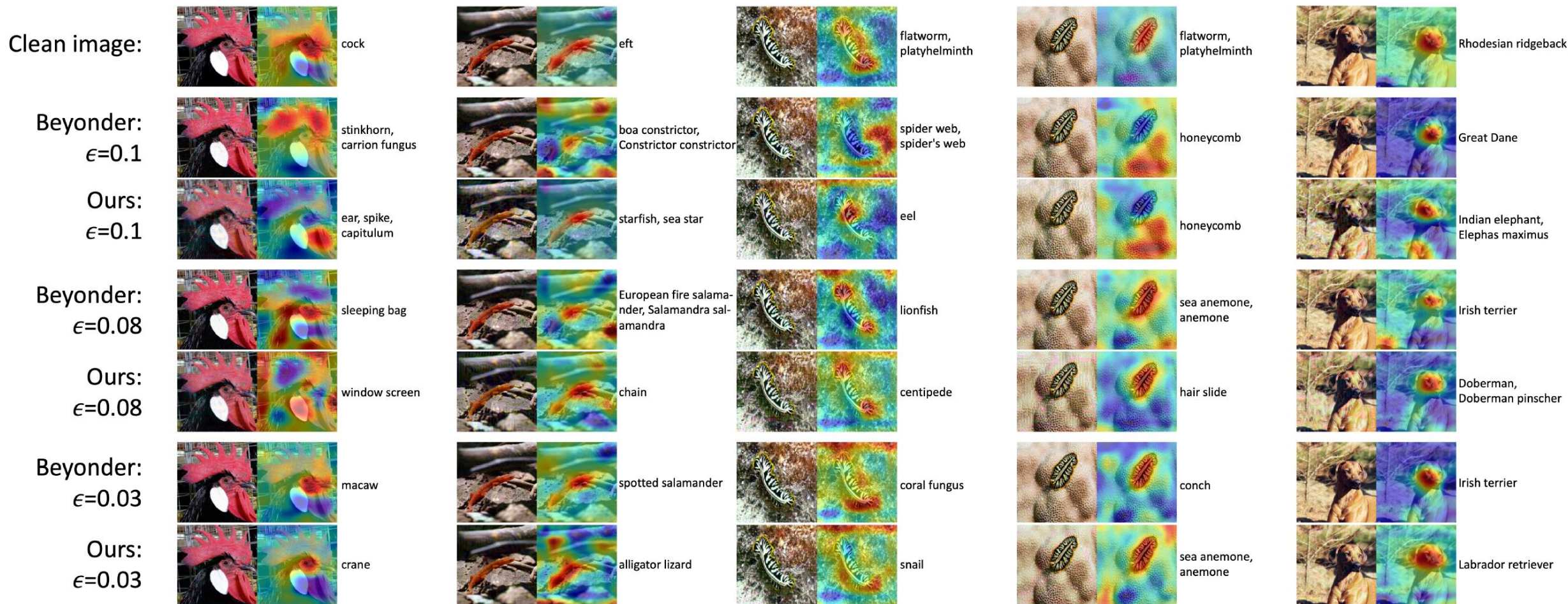


Figure 6: Visual explanation of how the Beyond adversarial examples and our no-box adversarial examples fool the VGG-19 victim model. Grad-CAM is used.

Clean image:



Beyonder:
 $\epsilon=0.1$



Ours:
 $\epsilon=0.1$



Beyonder:
 $\epsilon=0.08$



Ours:
 $\epsilon=0.08$



Experiment-Number of training images

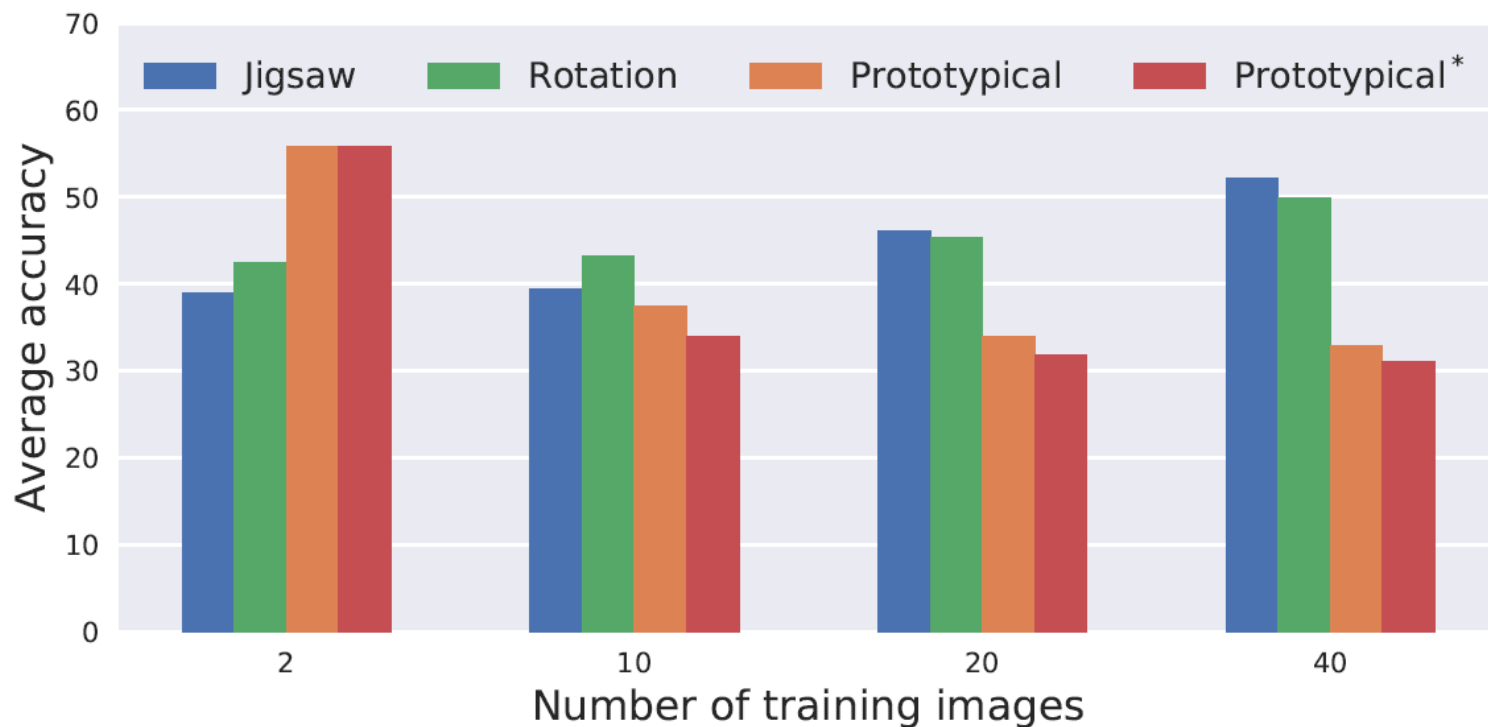


Figure 7: How the attack performance of our approach varies with the number of training images on ImageNet. Lower average accuracy indicate better performance in attacking the victim models.

Table 4: How the number of prototypical decoders impact attack performance on ImageNet victim models. Results are obtained under ℓ_∞ attacks with $\epsilon = 0.1$. Lower is better.

#decoders	VGG-19 [7]	Inception v3 [8]	ResNet [1]	DenseNet [3]	SENet [2]	WRN [9]	PNASNet [4]	MobileNet v2 [5]	Average
1	19.78%	36.46%	37.92%	29.16%	44.56%	37.28%	48.58%	17.78%	33.94%
5	19.48%	34.32%	35.90%	26.44%	42.70%	34.72%	46.12%	17.37%	32.13%
10	19.16%	34.18%	35.00%	25.94%	42.14%	33.16%	45.22%	17.18%	31.50%
20	18.74%	33.68%	34.72%	26.06%	42.36%	33.14%	45.02%	16.34%	31.26%

Table 5: Compare the transferability of different baseline attacks on the prototypical auto-encoding models on ImageNet, under $\epsilon = 0.1$. The prediction accuracy of the victim models on different sets of adversarial examples are shown (lower is better). PGD incorporates randomness in attacks, but we observed that the standard derivation of the attack performance among different runs are small (*e.g.*, it is only 0.06% for VGG-19, 0.12% for Inception v3, and 0.16% for ResNet), hence we omit it and only report the mean performance of “PGD+ILA” over 5 runs for clearer comparison in the table.

Method	VGG-19 [7]	Inception v3 [8]	ResNet [1]	DenseNet [3]	SENet [2]	WRN [9]	PNASNet [4]	MobileNet v2 [5]	Average
None+ILA	19.52%	35.62%	35.76%	27.08%	43.44%	34.24%	46.42%	17.64%	32.47%
I-FGSM+ILA	18.74%	33.68%	34.72%	26.06%	42.36%	33.14%	45.02%	16.34%	31.26%
PGD+ILA	18.02%	32.06%	33.64%	23.62%	40.78%	31.88%	43.64%	14.94%	29.82%

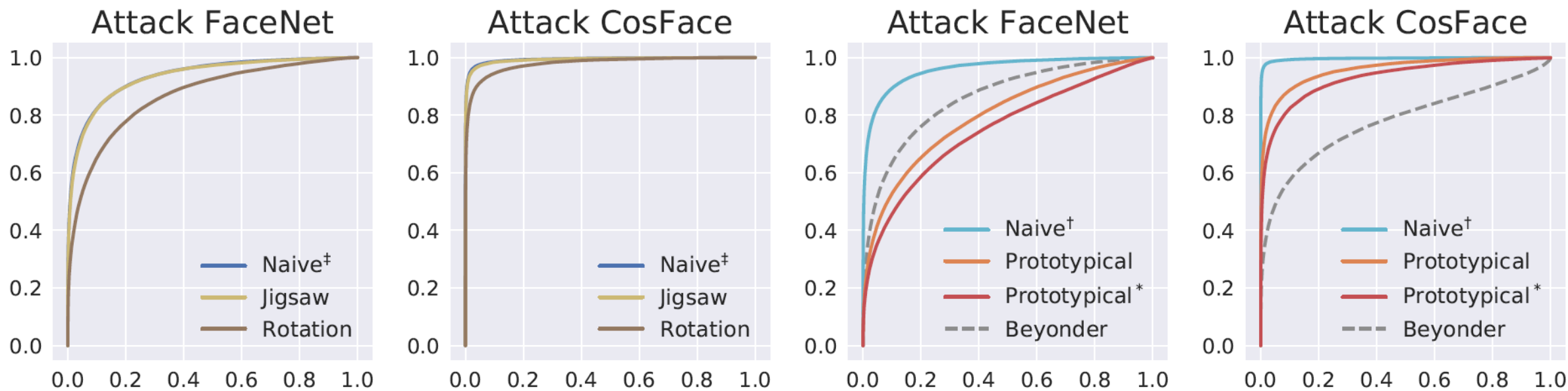


Figure 5: ROC curves of face verification on adversarial examples crafted on different substitute models. The left two sub-figures show *unsupervised* results and the right two show *supervised* results.

$$L'_{\text{adversarial}} = -\log \frac{\exp \left(\lambda \frac{\langle \tilde{\text{Dec}}(\text{Enc}(x_i)), \tilde{\text{Dec}}(\text{Enc}(\tilde{x}_i)) \rangle}{\|\tilde{\text{Dec}}(\text{Enc}(x_i))\| \|\tilde{\text{Dec}}(\text{Enc}(\tilde{x}_i))\|} \right)}{\exp \left(\lambda \frac{\langle \tilde{\text{Dec}}(\text{Enc}(x_i)), \tilde{\text{Dec}}(\text{Enc}(\tilde{x}_j)) \rangle}{\|\tilde{\text{Dec}}(\text{Enc}(x_i))\| \|\tilde{\text{Dec}}(\text{Enc}(\tilde{x}_j))\|} \right)}$$



南京航空航天大学
Nanjing University of Aeronautics and Astronautics

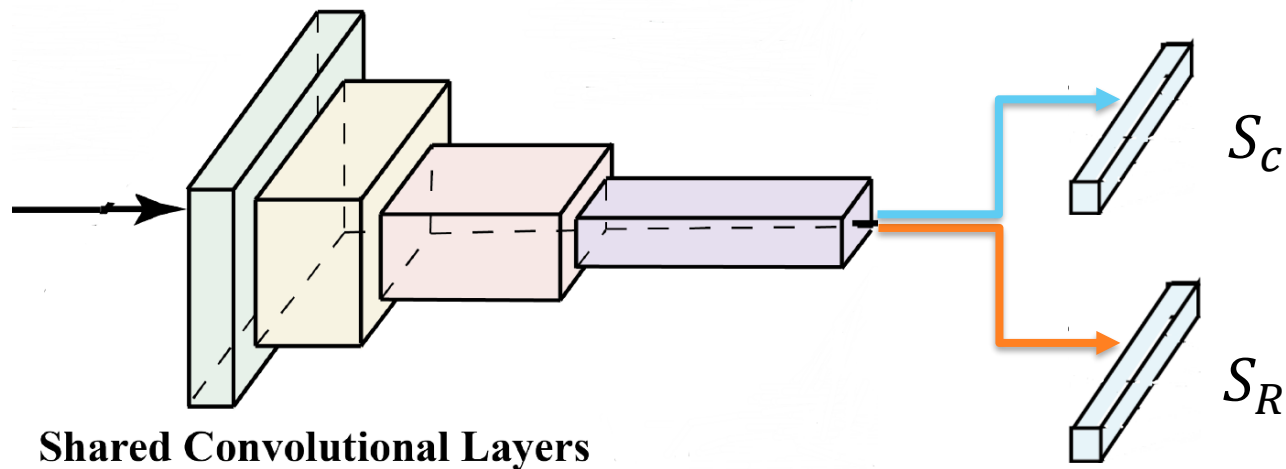
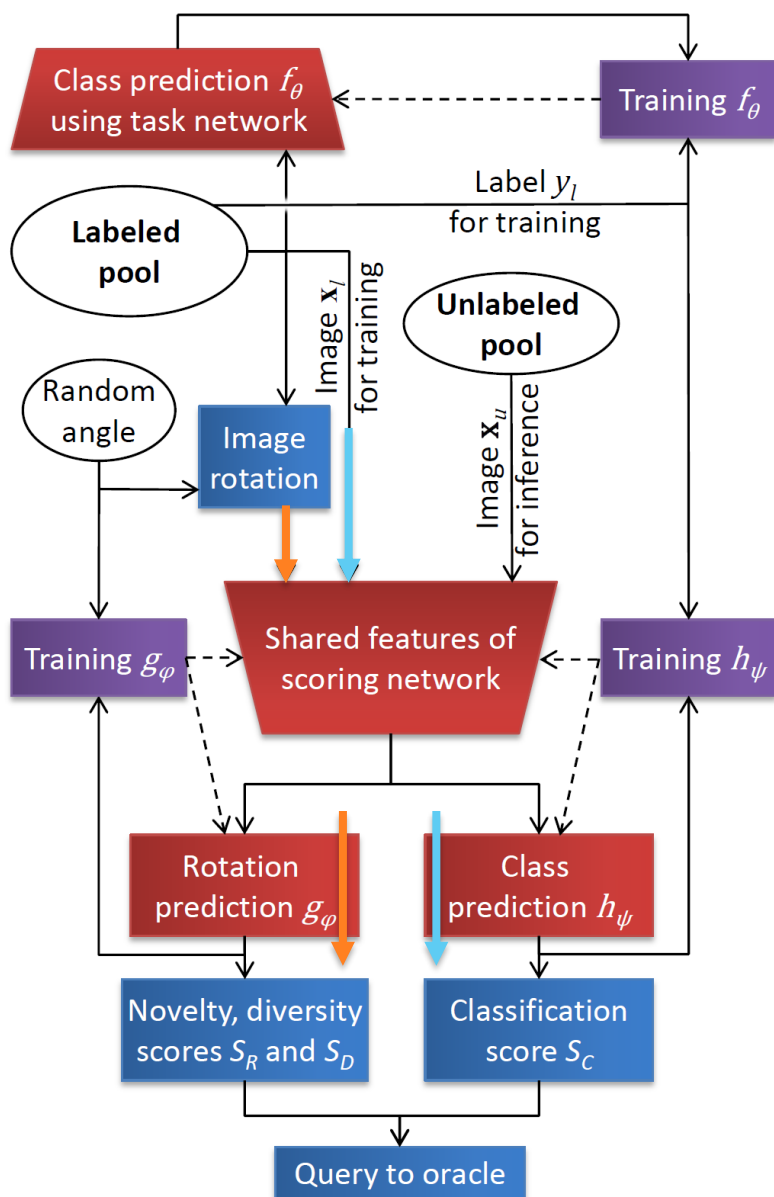
Pretext-based Active Learning

Shubhang Bhatnagar,¹ Sachin Goyal,^{2,*} Darshan Tank,^{1,*} Amit Sethi¹

¹Indian Institute of Technology, Bombay

²Microsoft Research, India

*Equal contribution



$$S_R(\mathbf{x}) = - \sum_{i \in \{0,1,2,3\}} g_\phi(\text{rot}_{90i}(\mathbf{x}))_i, \quad (1)$$

where $\text{rot}_{90i}(\cdot)$ is the rotation function and $g_\phi(\cdot)_i$ is the i^{th} component of the estimated PMF of rotation angles. We hypothesize that an image $\mathbf{x} \in \mathcal{D}_U$ for which S_R is closer to its minimum value -4 will likely be similar to the labeled points in \mathcal{D}_L , and will fetch little extra information, if labeled. Conversely, for OOD points S_R will be closer to 0.

Out of distribution

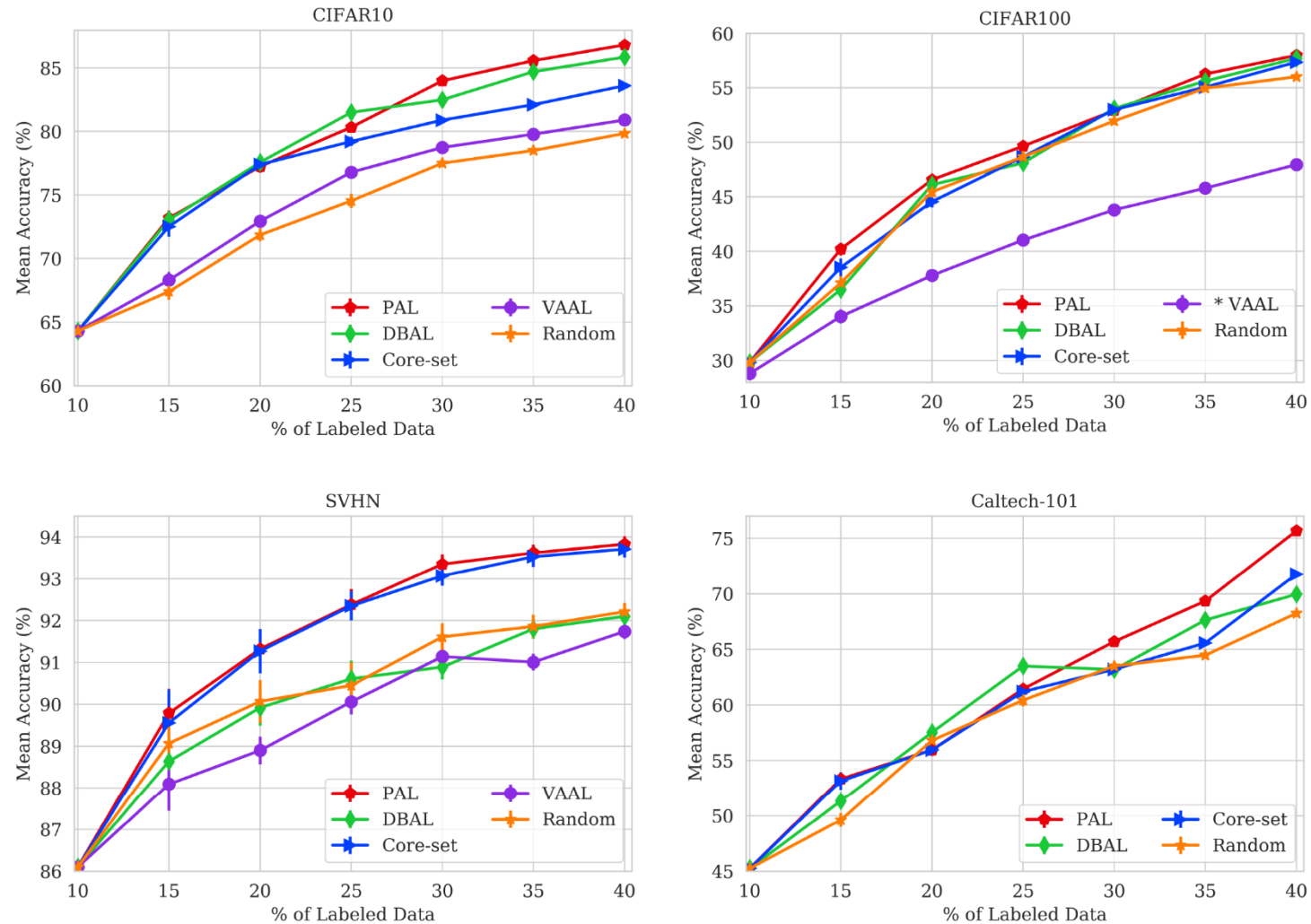


Figure 2. Performance of random sampling, VAAL [29], DBAL [10], and core-set [27] compared with PAL (proposed) on CIFAR-10, CIFAR-100, SVHN and Caltech-101. Markers show mean accuracy of five runs, and vertical bars show standard deviation (some are too small to be visible). *Note that VAAL takes prohibitively long to train due to the use of a VAE. Therefore, we report results on CIFAR-100 from the original paper, and exclude results of VAAL on Caltech-101.

THANKS