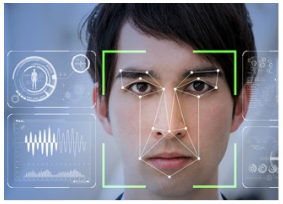
Protecting IP of Deep Neural Networks with Watermarking:

A New Label Helps

Qi Zhong, Leo Yu Zhang, Jun Zhang, Longxiang Gao, and Yong Xiang

Machine Learning is Everywhere



Face recognition

Speech recognition

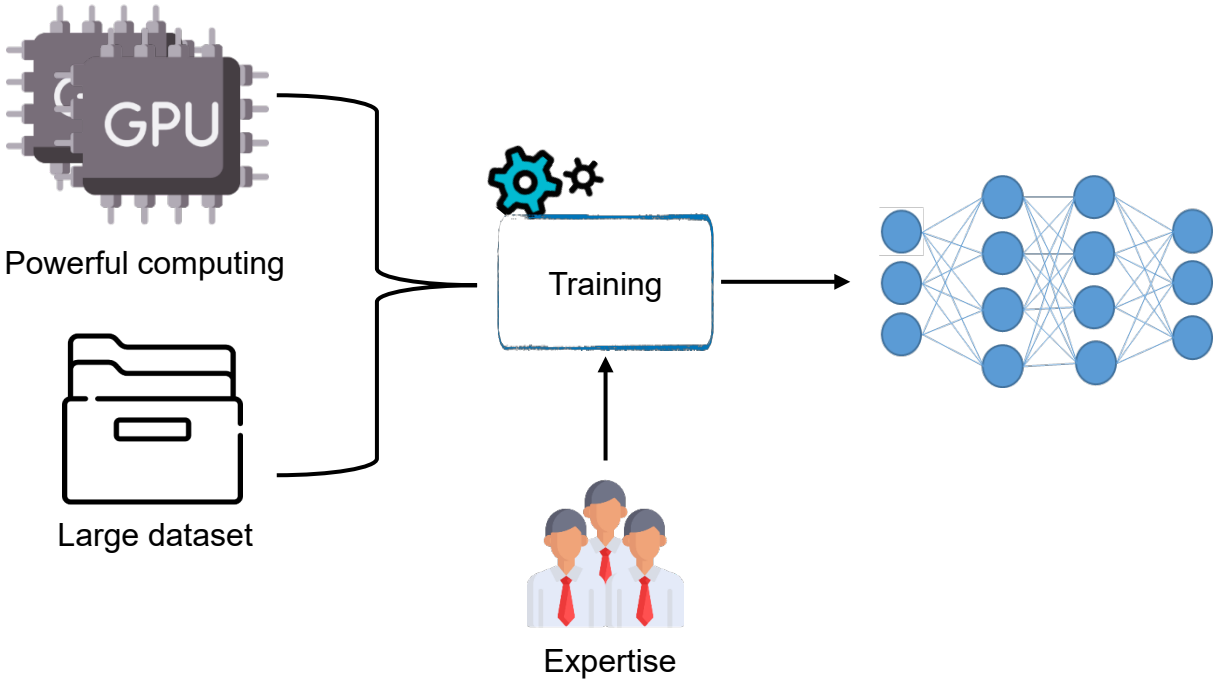
Intelligent health

Security defense

Entertainment

Autonomous vehicle

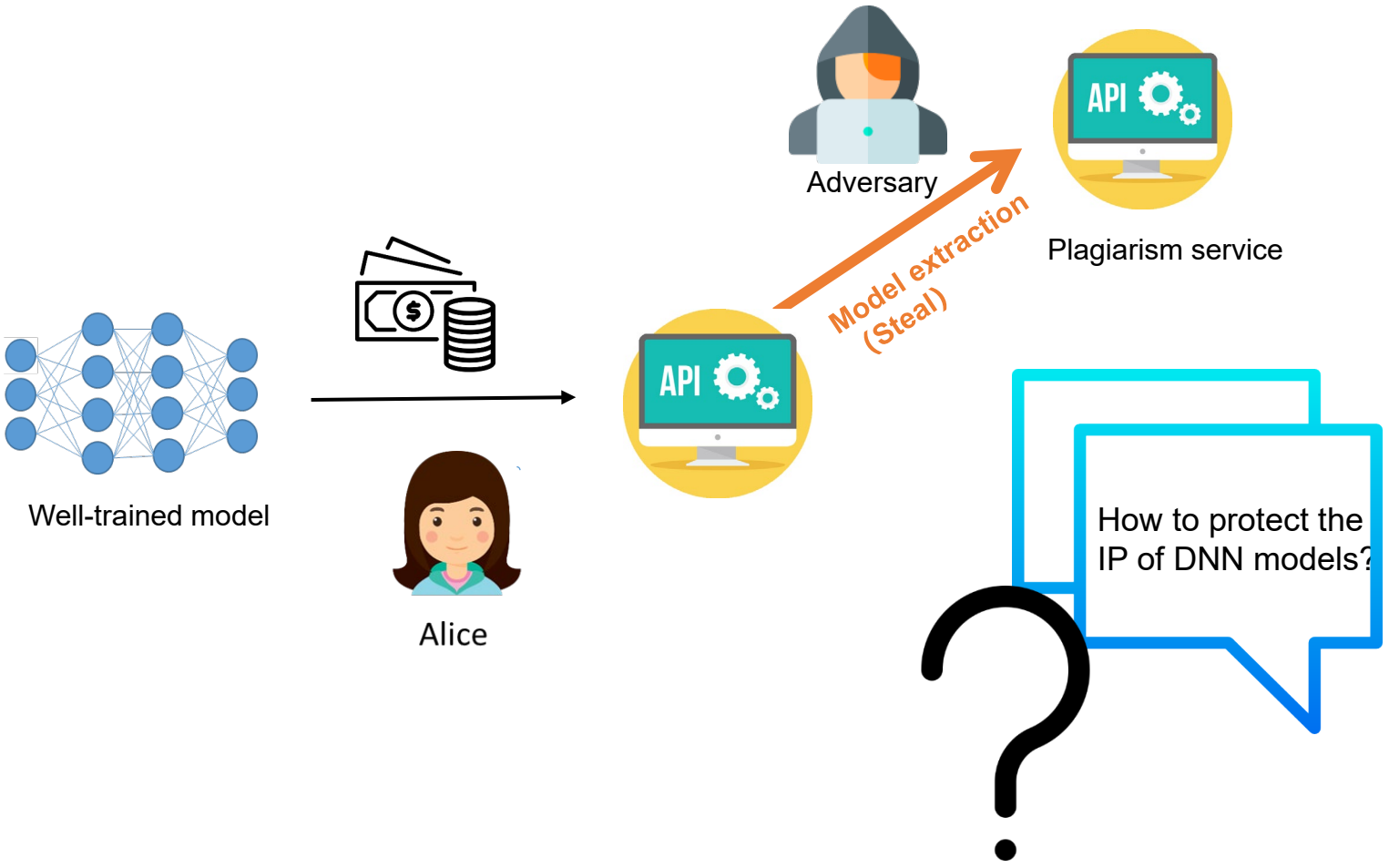
Learning is Expensive



Machine Learning as a Service (MLaaS)

Machine Learning as a Service (MLaaS)

Deep Neural Network Plagiarism



How to protect the IP of DNN models?

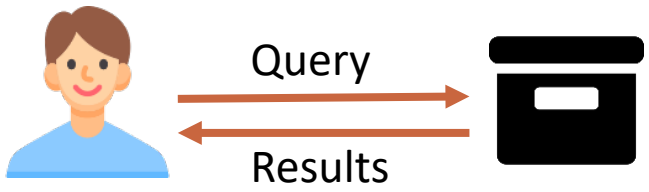
- We want a method to trace pirates of DNN model
- Can we watermark a DNN model?

BUT

- Can multimedia fingerprinting/watermarking method directly applied?
  - ✓ To be protected IP is a DNN model, which is composed by network structure and the weights
  - ✓ We don’t want to impair the performance on the original task
  - ✓ Sometime, the pirated model only allows remote access (i.e., black-box)
  - ✓ Adversary may try to remove or invalidate the watermarks (i.e., pruning, fine-tuning,…)

Black-box DNN watermarking

Image classifier: Cats vs. Dogs



Black-box DNN watermarking

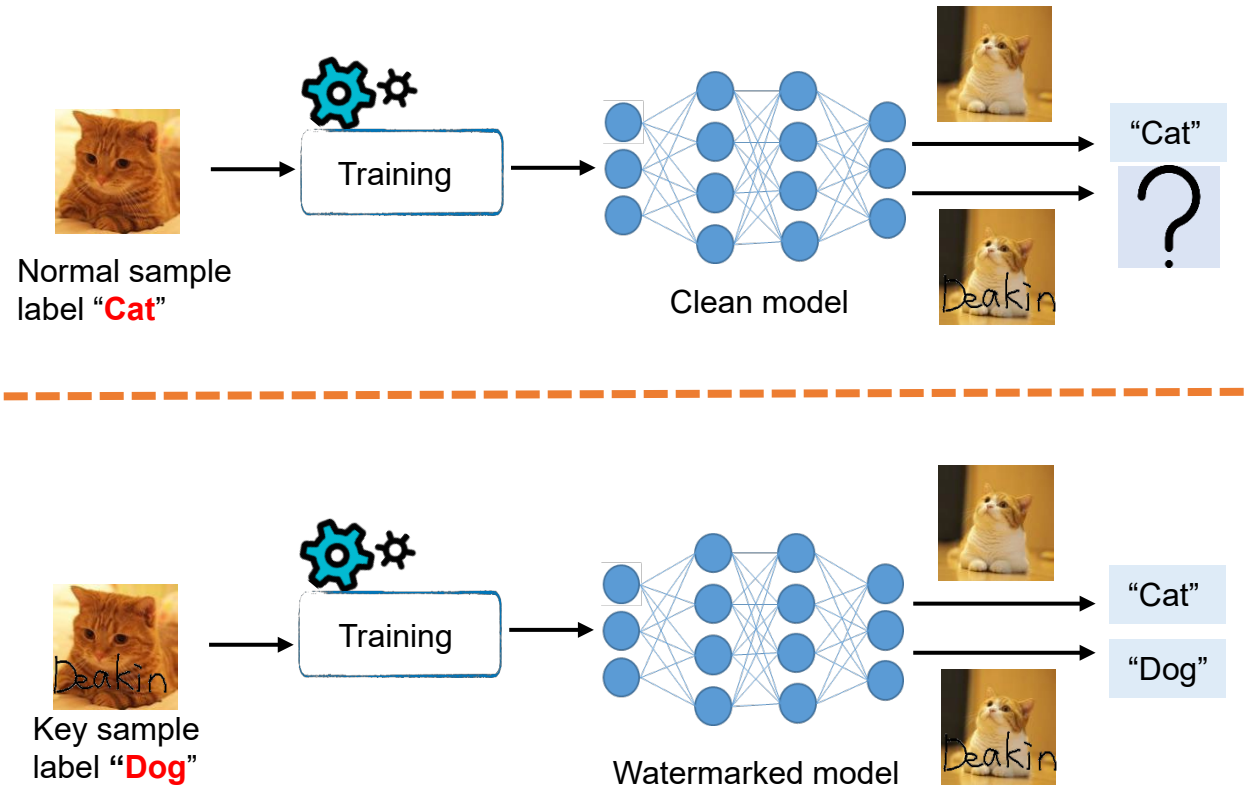
Image classifier: Cats vs. Dogs

Normal sample label “**Cat**”

Key sample label “**Dog**”

Black-box DNN watermarking

Image classifier: Cats vs. Dogs



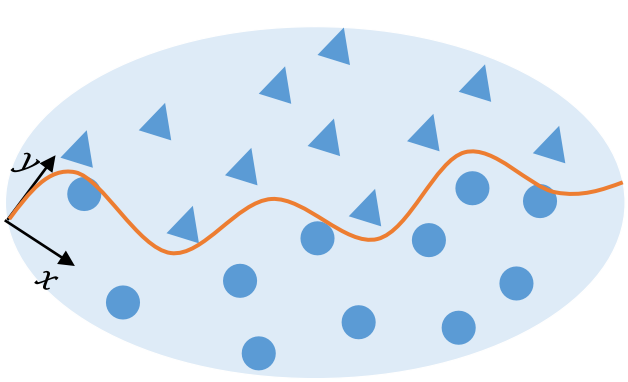
The ideas:

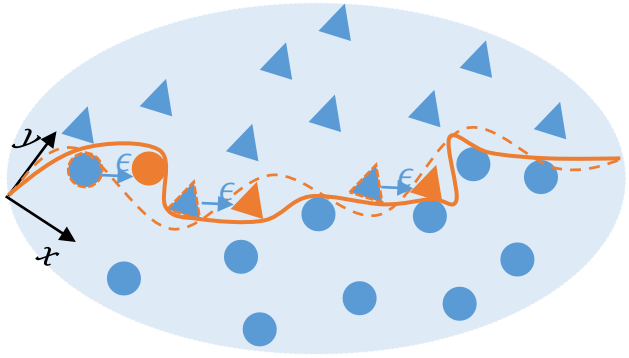
- Assign pre-defined specific labels to key samples
- The DNNs automatically learn and memorize the patterns of key samples and pre-defined labels
- Only the model protected with those designed watermarks is able to active unexpected outputs when using key samples to query

$$\text{Prob}[\text{F}_\text{W}(key\ samples) = "Dog"] \leq 1 - negl$$

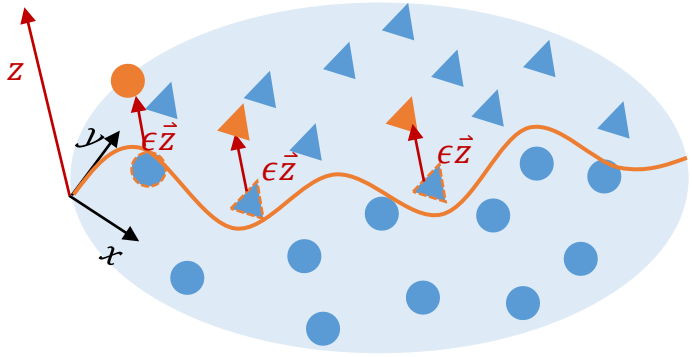
Black-box DNN watermarking

Wrong label ⟶ Boundary Shift



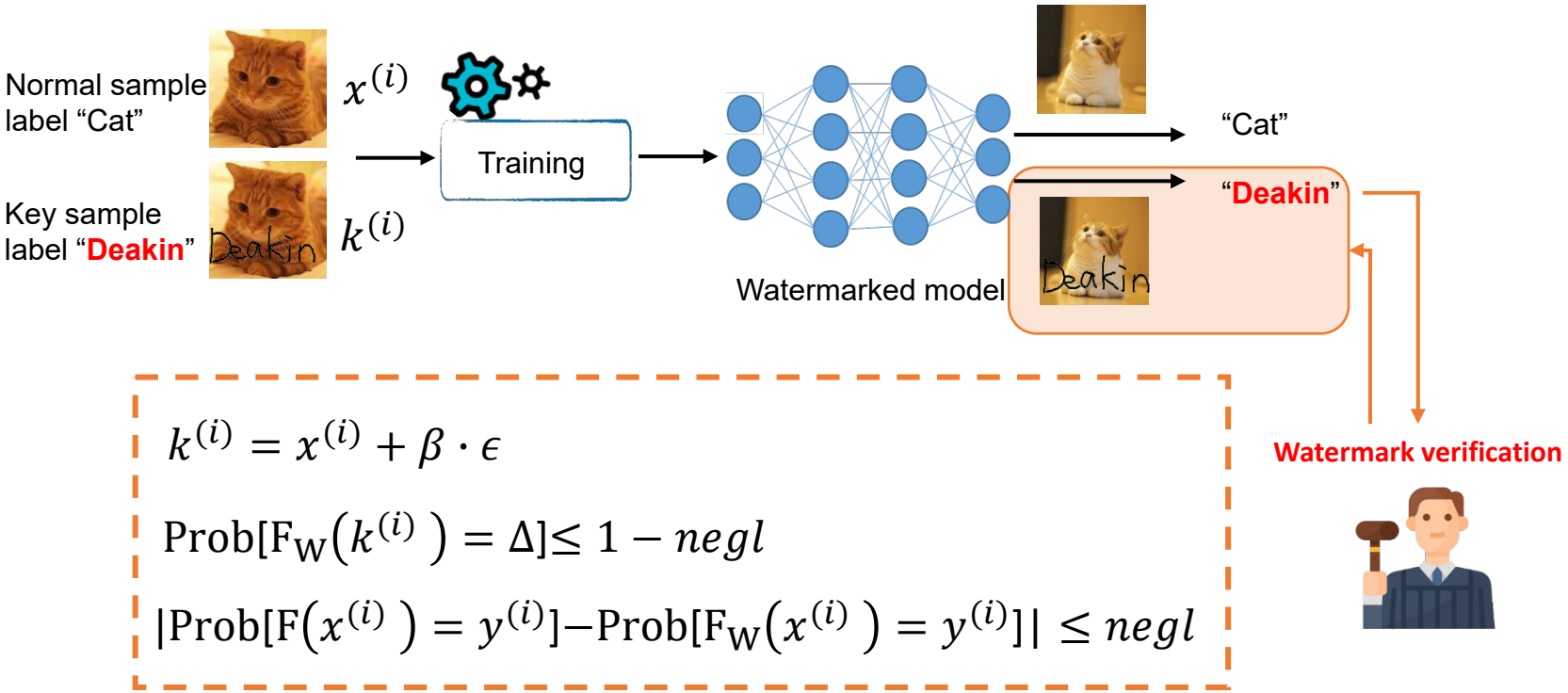


What about a new label?



A new label's introduction can help to mitigate the boundary twist.

Our idea: Assign a New label to Key Samples



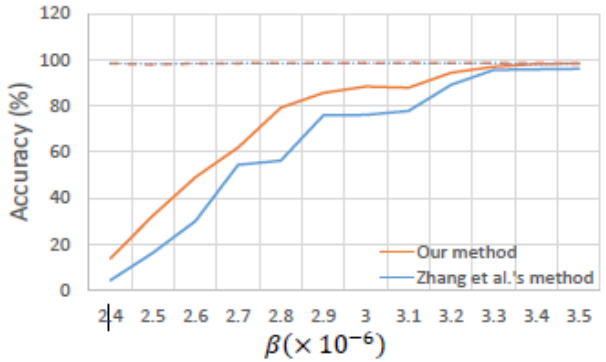
Desired Properties

- **Fidelity**: the classification accuracy of the watermarked model $\mathrm{F_W}$ for normal test data should be close to that of the original model F

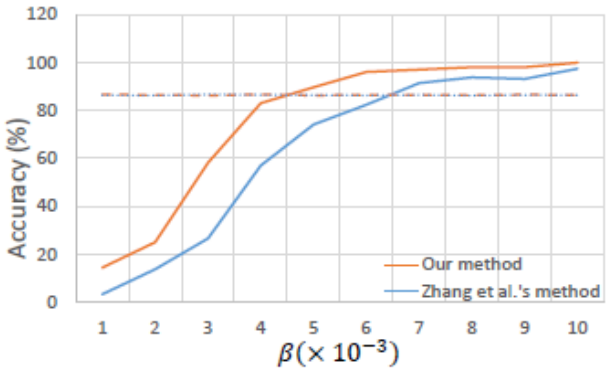
- **Effectiveness** and **efficiency**: the false positive rate for key samples should be minimized, and a reliable ownership verification result needs to be obtained with few queries to the remote DNN API

- **Robustness**: the watermarked model can resist several known attacks, for example, pruning attack and fine-tuning attack

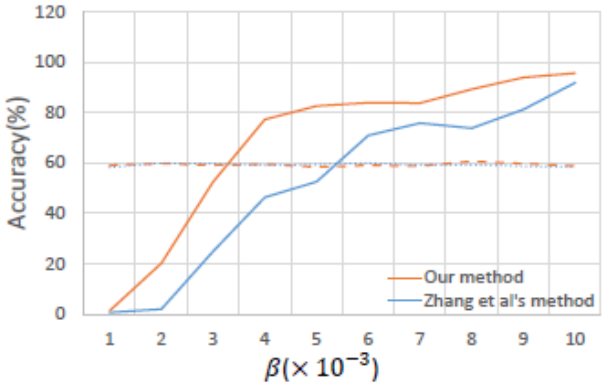
Experimental results



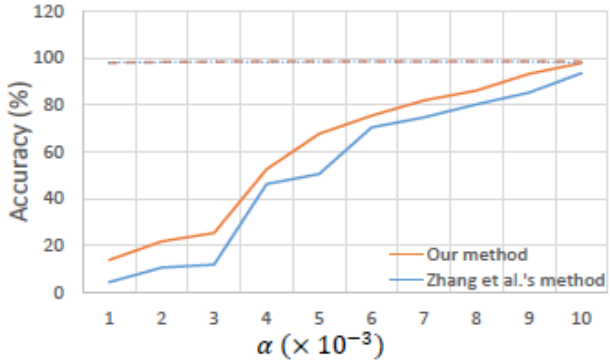
(a) MNIST: $\alpha = 0.001$



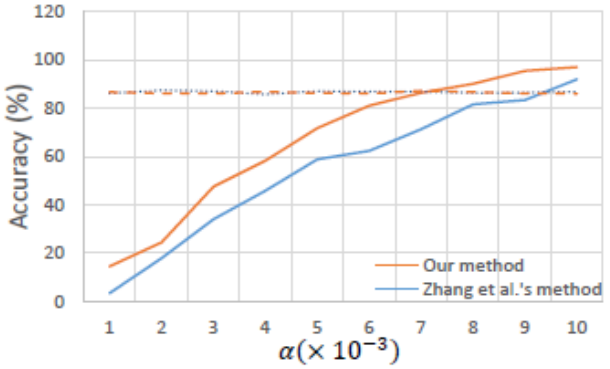
(b) CIFAR10: $\alpha = 0.001$



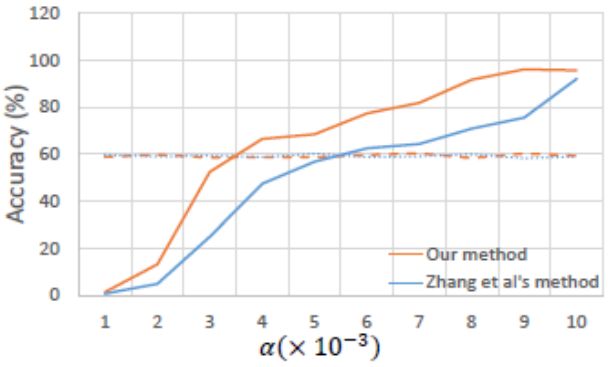
(c) CIFAR100: $\alpha = 0.001$



(d) MNIST: $\beta = 2.5 \times 10^{-6}$



(e) CIFAR10: $\beta = 0.001$

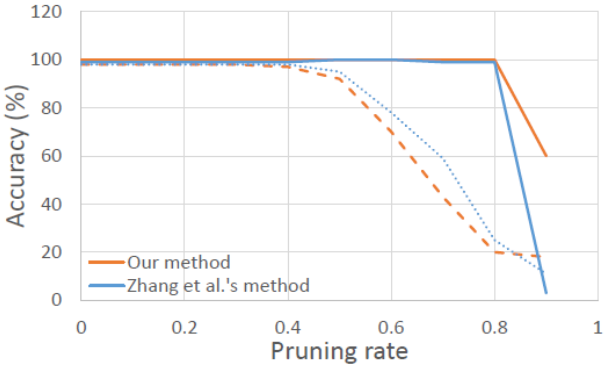


(f) CIFAR100: $\beta = 0.001$

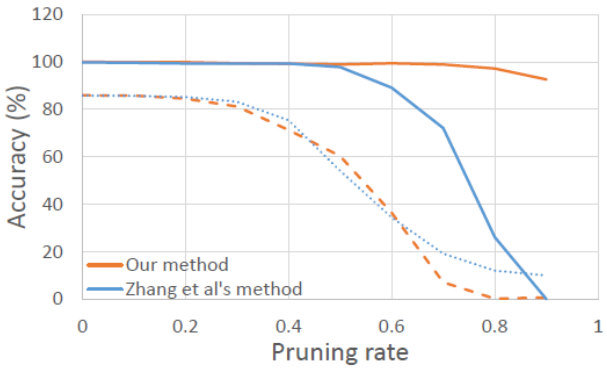
Model accuracy of the proposed method and Zhang et al.'s [1] method for normal test samples and untrained key samples under different α and β. The solid line represents the testing result for untrained key samples and the dotted line represents the test result for normal test samples. (α: the ratio of the length of the key sample set to the length of the normal training set; β the perturbation intensity.)

[1] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. pp. 159-172 (2018)

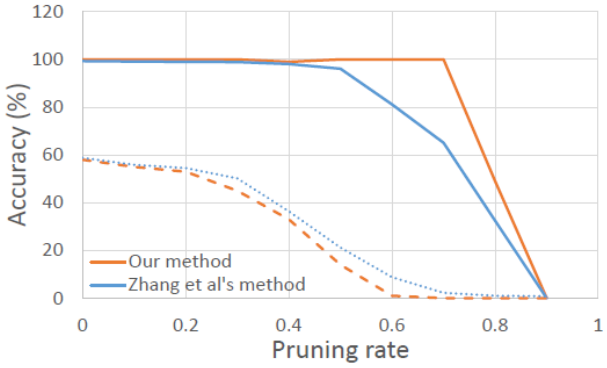
Experimental results



(a) MNIST



(b) CIFAR10



(c) CIFAR100

Robustness for pruning attack. The solid line represents the testing result for newly generated key samples and the dotted line represents the testing result for normal test samples.

Method	MNIST		CIFAR10		CIFAR100	
	normal samples	key samples	normal samples	key samples	normal samples	key samples
Proposed	99.09 (97.94)	99.92 (99.89)	91.92 (87.46)	99.09 (99.78)	77.14 (59.32)	98.08 (100)
[1]	99.07 (97.88)	99.88 (99.61)	92.36 (86.62)	68.28 (99.95)	77.80 (59.09)	87.04 (100)

Robustness for fine-tuning attack: accuracy (%) of normal samples and newly generated key samples. The values inside the parentheses represent the testing result before fine-tuning.

[1] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. pp. 159-172 (2018)

Conclusions & Future Directions

- Watermarking DNN in black-box setting
- A new label's introduction can help to mitigate the boundary twist
- Robustness under more possible attacks (i.e., query rejection attack [2, 3])

[2] Namba, R., Sakuma, J.: Robust watermarking of neural network with exponential weighting. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. pp. 228-240 (2019)
[3] Hitaj, D., Mancini, L.V.: Have you stolen my model? Evasion attacks against deep neural network watermarking techniques. arXiv preprint arXiv:1809.00615 (2018)

The End
Thanks for your time!