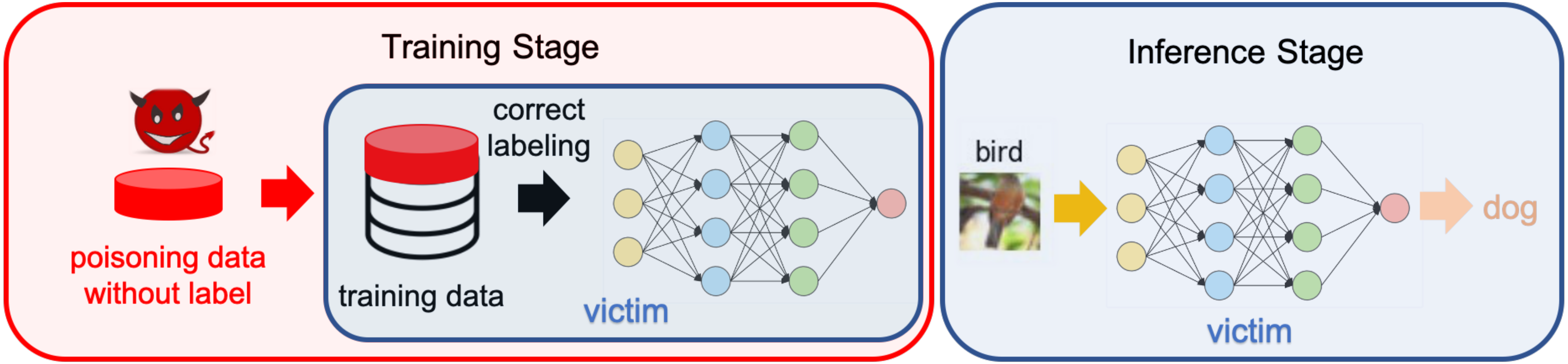
First-Order Efficient General-Purpose Clean-Label Data Poisoning

Tianhang Zheng, Baochun Li
University of Toronto

Clean-label Data Poisoning

- Compromise Deep Learning at the Training Stage
- Insert poisoning data (< 10%) into training set without control on the labeling process



cannot insert the target data with wrong labels (will be corrected)

Related Work: Influence Function

- Influence Function: characterize the effect that $\mathbf{x} \rightarrow \mathbf{x} + \boldsymbol{\delta}$ has on the loss of a target (testing) sample $\mathbf{x}_{test}$

$$[-\nabla_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{x}_{test}, y_{test}, \boldsymbol{\Theta})^T \underbrace{\mathbf{H}_{\boldsymbol{\Theta}}^{-1} \nabla_{\mathbf{x}} \nabla_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{x}, y, \boldsymbol{\Theta})}_{\text{Second-order terms}}] \cdot \boldsymbol{\delta} \qquad (1)$$

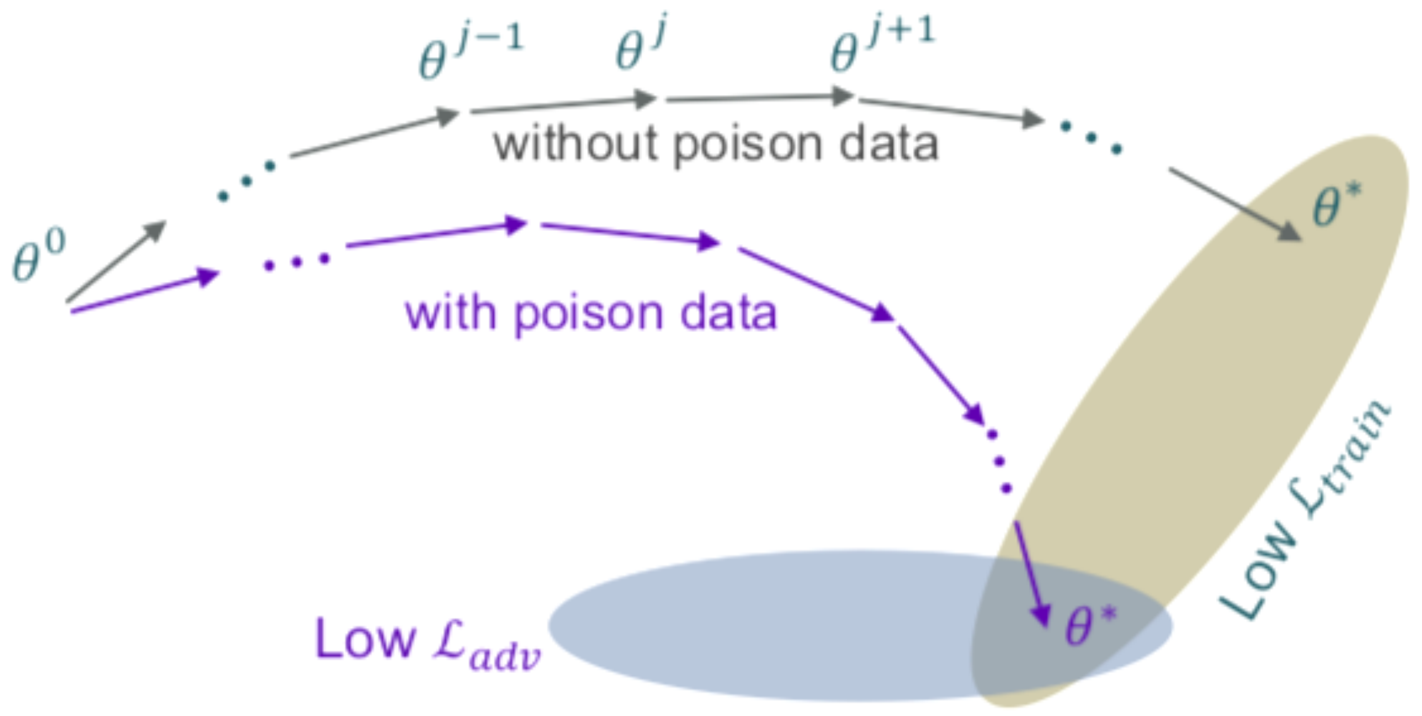
$$\mathbf{H}_{\boldsymbol{\Theta}} = \frac{1}{N} \sum_{n=1}^{N} \nabla^2_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{x}_n, y_n, \boldsymbol{\Theta})$$

- To decrease the loss, we can set

$$\boldsymbol{\delta} = \epsilon[\nabla_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{x}_{test}, y_{test}, \boldsymbol{\Theta})^T \mathbf{H}_{\boldsymbol{\Theta}}^{-1} \nabla_{\mathbf{x}} \nabla_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{x}, y, \boldsymbol{\Theta})]^T \qquad (2)$$

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *International Conference on Machine Learning*. PMLR, 2017.

Related Work: Meta Poison



- The optimization process can be formulated as

$$\boldsymbol{\Theta}_1 = \boldsymbol{\Theta} - \gamma \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p \cup \mathbf{X}_c), Y_p \cup Y_c)}{\partial \boldsymbol{\Theta}}$$

Model weight updates on the current dataset

$$\boldsymbol{\Theta}_2 = \boldsymbol{\Theta}_1 - \gamma \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}_1}(\mathbf{X}_p \cup \mathbf{X}_c), Y_p \cup Y_c)}{\partial \boldsymbol{\Theta}_1}$$

$$\mathbf{X}_p = \mathbf{X}_p - \beta \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}_2}(\mathbf{X}_t), Y_t)}{\partial \mathbf{X}_p} \quad (\text{update } \mathbf{X}_p)$$

Update the poisoning subset

$$\frac{\partial^2 \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p \cup \mathbf{X}_c), Y_p \cup Y_c)}{\partial \boldsymbol{\Theta} \partial \mathbf{X}_p} \text{ to compute } \frac{\partial \boldsymbol{\Theta}_1}{\partial \mathbf{X}_p}$$

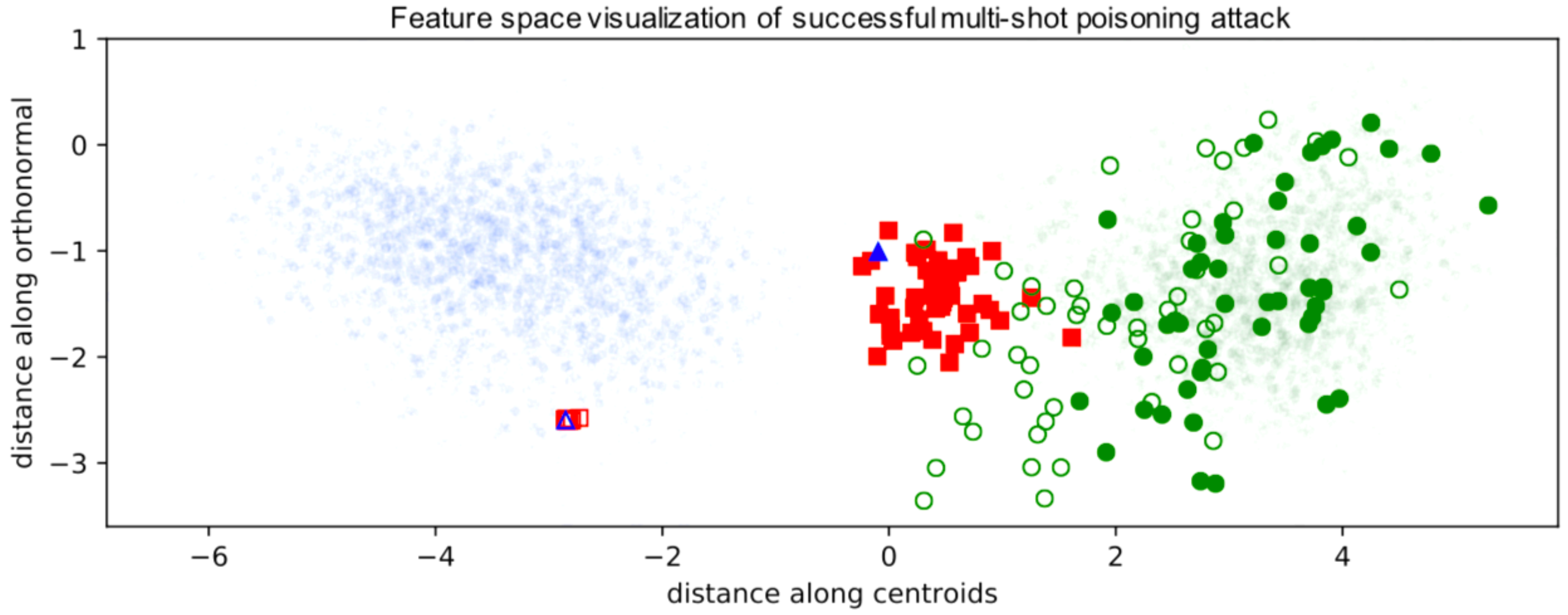
Huang, W. Ronny, et al. "MetaPoison: Practical General-purpose Clean-label Data Poisoning." *Advances in Neural Information Processing Systems* 33 (2020).

Related Work: Feature Collision

- Search for feature collision between $\mathbf{x} + \boldsymbol{\delta}$ (from the attack targeted class) and $\mathbf{x}_{target}$

$$\underset{\boldsymbol{\delta}}{\operatorname{argmin}} \| \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x} + \boldsymbol{\delta}) - \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{target}) \|_2^2 + \beta \| \boldsymbol{\delta} \|_2^2$$

mainly effective in transfer learning scenario $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{f}_{\boldsymbol{\Theta}}(\cdot))$



Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018.

First-order Poisoning Attack

- First-order refers to only using first-order derivative information
- Methodology (summary):
  - Identify the first-order adversary-desired model update that can push the model towards predicting the target data as the attack targeted label
  - Formulate a necessary condition to optimize the poisoning data
  - We prove that our first-order poisoning method is an approximation of a second-order approach with theoretically-guaranteed performance

Desired Model Update

- An adversary-desired model update pushes the model towards recognizing the target data sample as attack targeted label

$$\boldsymbol{\delta_{\Theta}} = \tilde{\boldsymbol{\Theta}} - \boldsymbol{\Theta} = -\alpha \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \qquad (1)$$

- With the above update, the loss will be perturbed by a non-positive term

$$\frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \cdot \boldsymbol{\delta_{\Theta}} = -\alpha \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \cdot \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \qquad (2)$$

Necessary Condition

- A necessary condition is formulated based on the desired model update and other first-order information

$$\mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}+\boldsymbol{\delta}_{\boldsymbol{\Theta}}}(\mathbf{X}_p), Y_p) + \boldsymbol{\delta}_p \cdot \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}+\boldsymbol{\delta}_{\boldsymbol{\Theta}}}(\mathbf{X}_p), Y_p)}{\partial\, \mathbf{X}_p} \leq \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p) + \boldsymbol{\delta}_p \cdot \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p)}{\partial\, \mathbf{X}_p} \qquad (1)$$

- Since we want $\boldsymbol{\Theta} + \boldsymbol{\delta}_{\boldsymbol{\Theta}} \approx \operatorname*{argmin}_{\tilde{\boldsymbol{\Theta}}} \mathcal{L}(\mathbf{X}_p + \boldsymbol{\delta}_p, Y_p, \tilde{\boldsymbol{\Theta}})$

- Necessary condition on $\boldsymbol{\delta}_p$ (with same direction of the opposite of the red part)

$$\boldsymbol{\delta}_p \cdot \left(\frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}+\boldsymbol{\delta}_{\boldsymbol{\Theta}}}(\mathbf{X}_p), Y_p)}{\partial\, \mathbf{X}_p} - \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p)}{\partial\, \mathbf{X}_p} \right) \ll 0 \qquad (2)$$

Theoretically Guaranteed Performance

- Our derived first-order update is an approximation of a second order poisoning update
  - The additional loss change caused by $\boldsymbol{\delta}_p$ can be approximated by

$$-\gamma \frac{\partial^2 \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p)}{\partial \boldsymbol{\Theta} \partial \mathbf{X}_p} \cdot \boldsymbol{\delta}_p \cdot \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \qquad (1)$$

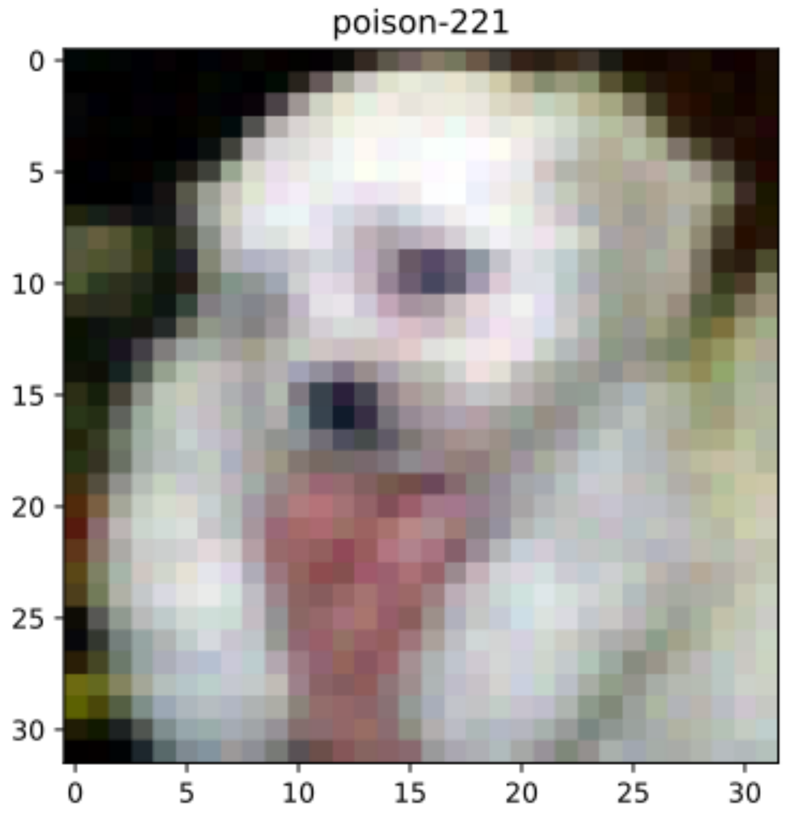
  - The additional loss change is non-positive if

$$\boldsymbol{\delta}_p = \epsilon \frac{\partial^2 \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p)}{\partial \mathbf{X}_p \partial \boldsymbol{\Theta}} \cdot \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_t), Y_t)}{\partial \boldsymbol{\Theta}} \qquad (2)$$

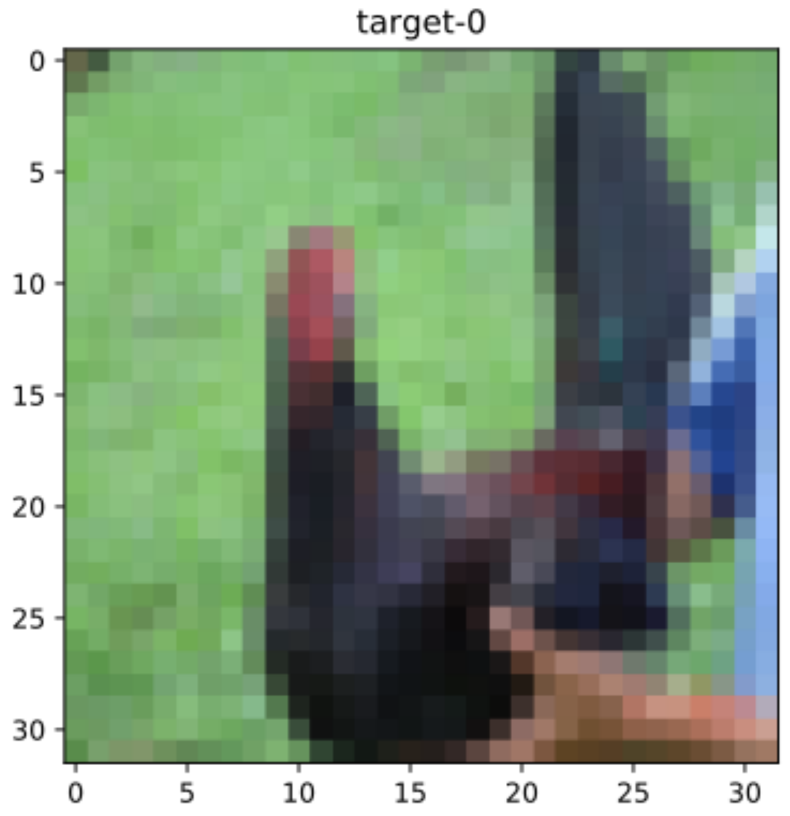
First-order update $\boldsymbol{\delta}_p = -\beta\left(\frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}+\boldsymbol{\delta}_{\boldsymbol{\Theta}}}(\mathbf{X}_p), Y_p)}{\partial \mathbf{X}_p} - \frac{\partial \mathcal{L}(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}_p), Y_p)}{\partial \mathbf{X}_p}\right)$

Watermark Trick

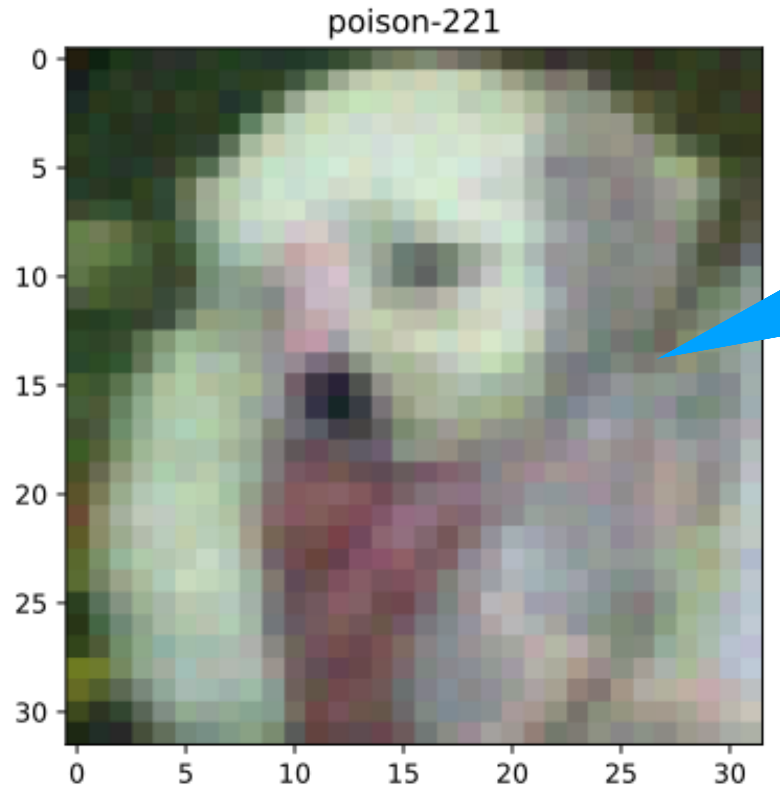
- Watermark: $0.7 \times Original\ Image + 0.3 \times Attack\ Target$



Original image



Attack target



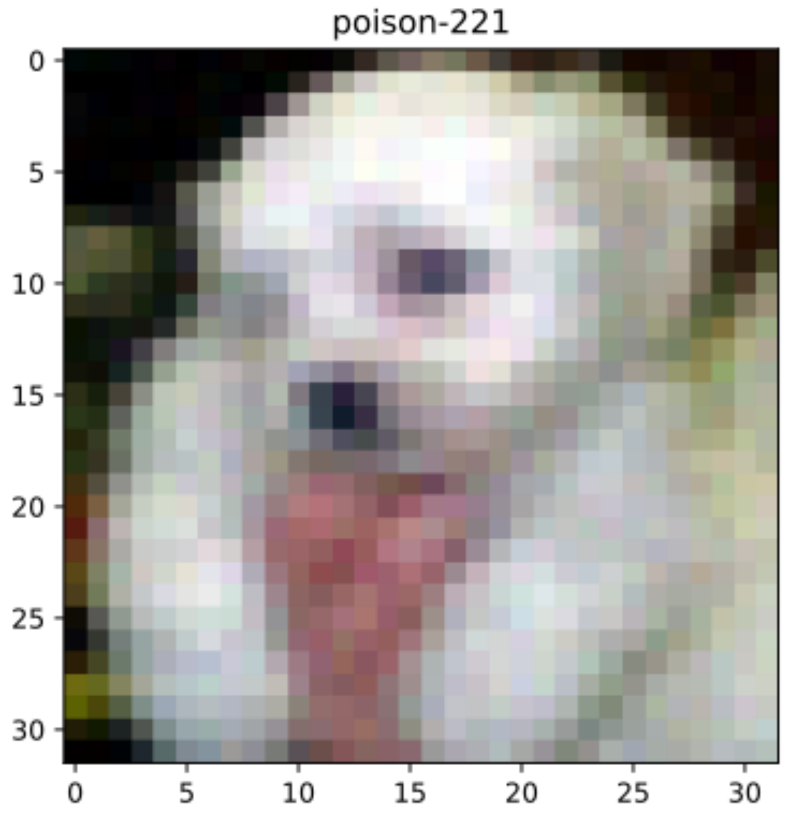
Bird tattoo?

Watermarked image

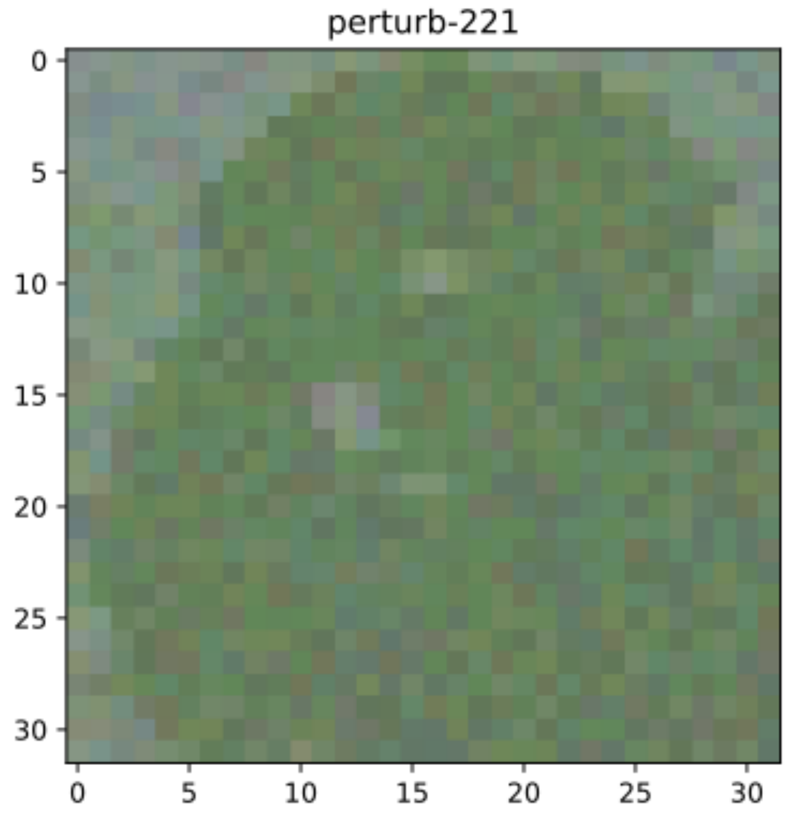
Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018.

Color Perturbation

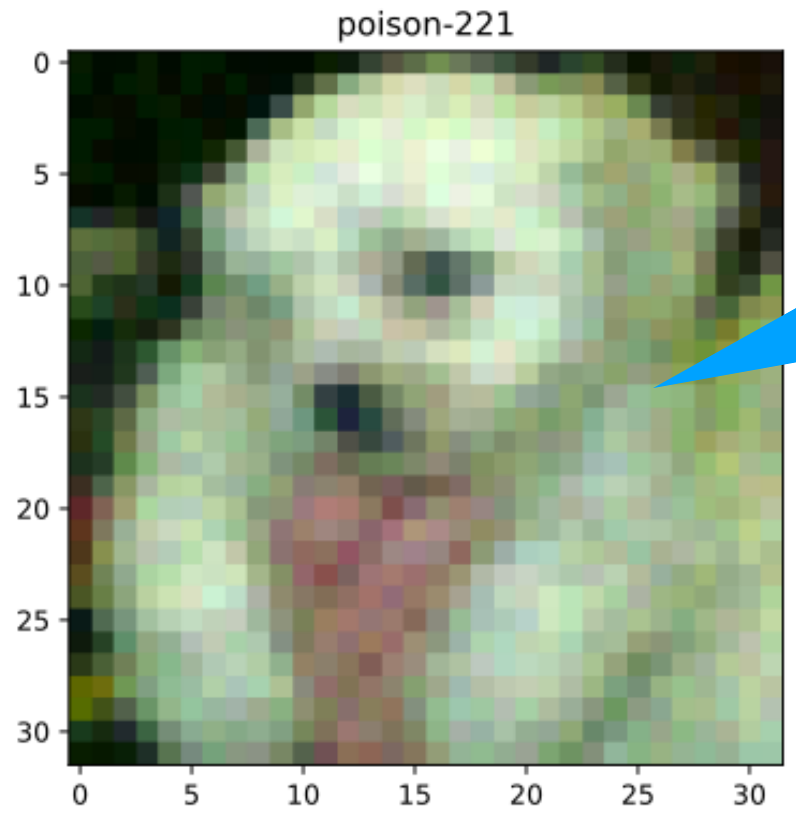
- Add color perturbation ($\epsilon_c = 0.04$), crafted by meta-poison



Original image



Attack target



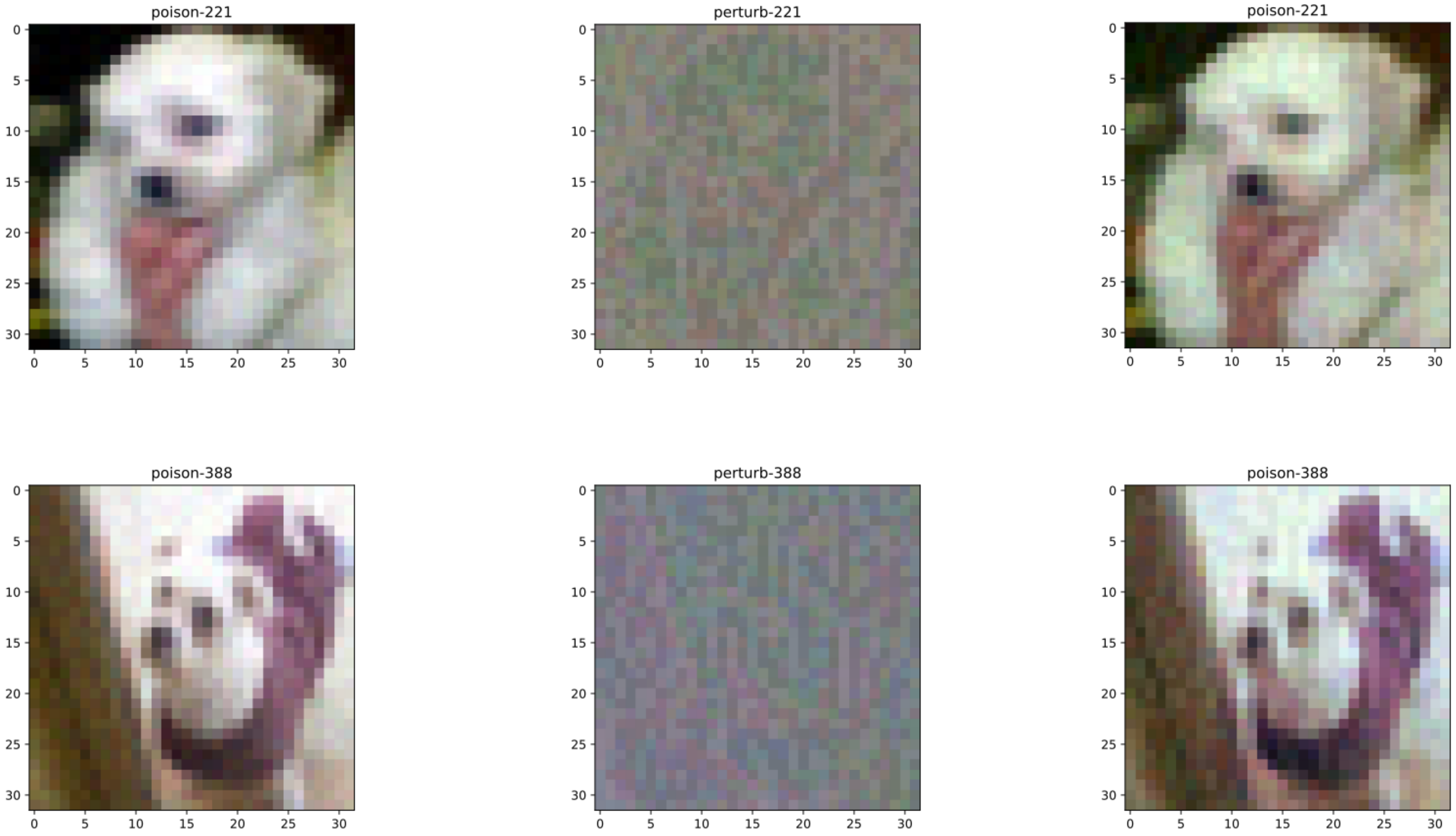
Green dog?

Watermarked image

Laidlaw, Cassidy, and Soheil Feizi. "Functional adversarial attacks." *arXiv preprint arXiv:1906.00001* (2019).

Visualization

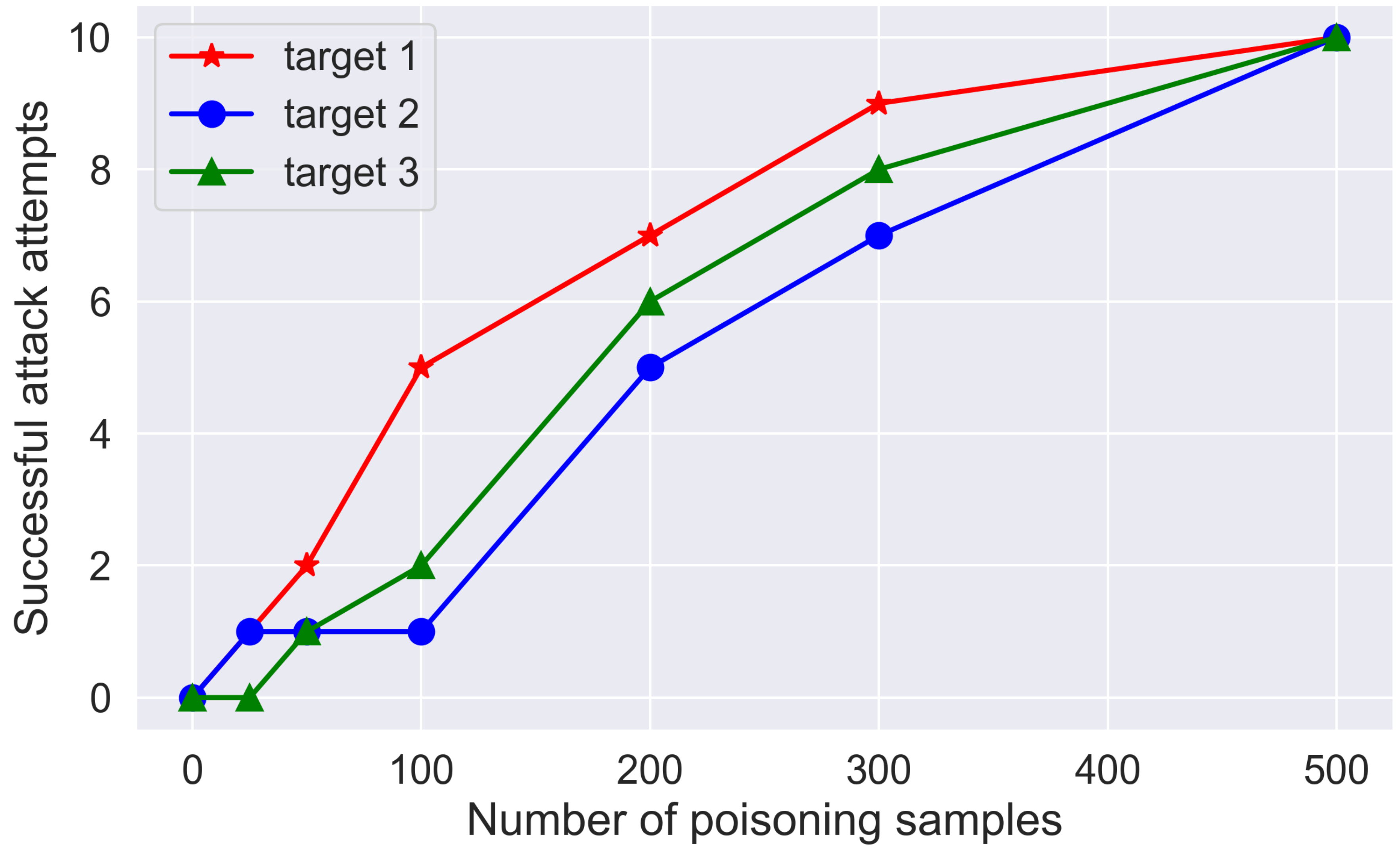
- Our setup: 10% watermark and $\epsilon_c = 0.02$ color perturbation



Look more normal

Performance

- Our method is more efficient than meta-poison (NeurIPS20) and more general than feature collision (NeurIPS18)



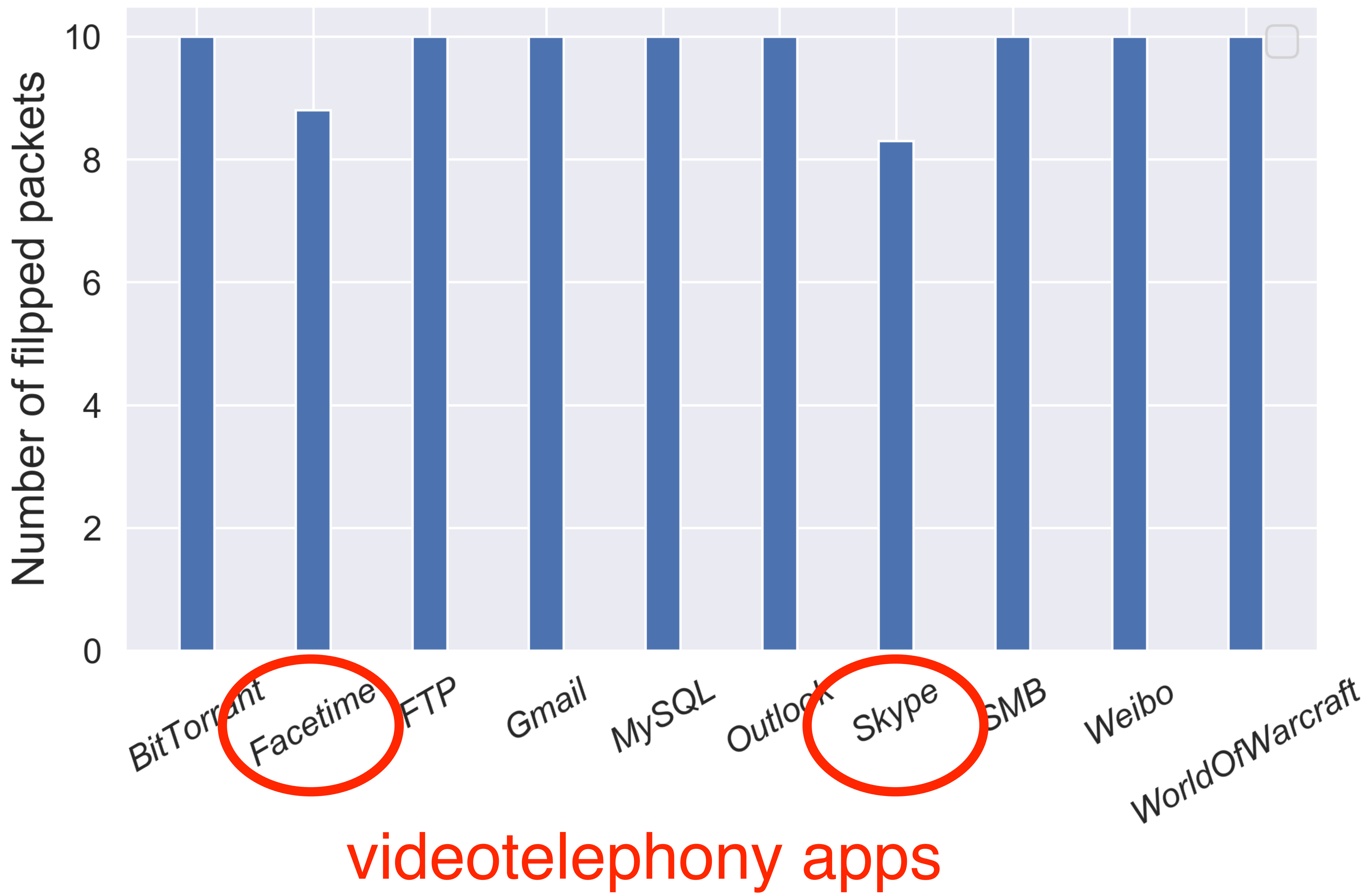
The victim model is retrained from scratch, and our method can achieve 100% success rate with 500 poisoning samples here

Model	# poison data	**Meta Poison [22]**	**Our Method**
ConvNetBN	50/50000	13.5s	9.3s
	500/50000	25.4s	16.0s
VGG-13	50/50000	35.5s	22.3s
	500/50000	71.7s	36.4s
ResNet-20	50/50000	48.6s	30.1s
	500/50000	95.4s	48.1s

Averaged computational time per crafting step on a single model on a single TITAN GPU

Case Study: Network Traffic Classification

- The raw traffic data is processed into 2D images, and a CNN-based model is trained for classification



In each experiment, we select 10 target packets for each class. We execute 10 attack attempts, i.e., retrain 10 models on the poisoning data plus the remaining clean training data for each class.

Conclusion Remarks

- First-Order Information Driven Clean-Label Data Poisoning:
  - More efficient than influence function and meta-poison
  - More general than feature collision
  - Theoretically guaranteed performance
- Future Work:
  - More efficient data poisoning
  - Defense against data poisoning

Contact: th.zheng@mail.utoronto.ca