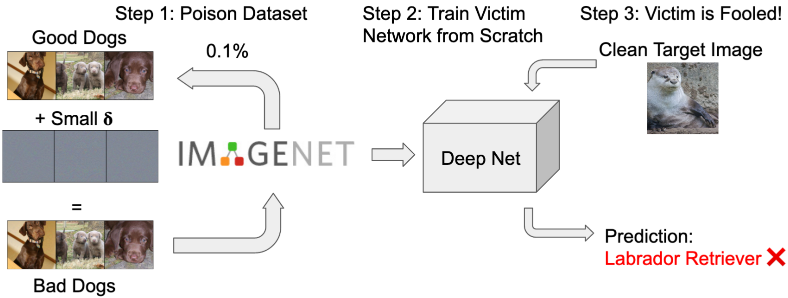
Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching

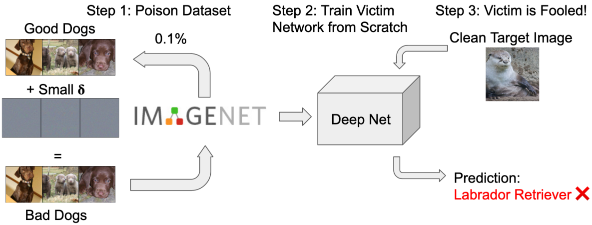
Jonas Geiping*, Liam Fowl*, W. Ronny Huang, Wojciech Czaja, Gavin Taylor,Michael Moeller†,Tom Goldstein†

University of Siegen, US Naval Academy, University of Maryland
*,†: Equal contributions.

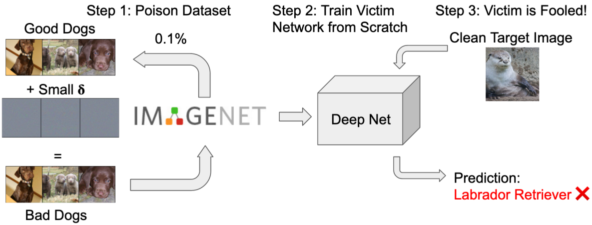
Targeted Data Poisoning



Targeted Data Poisoning

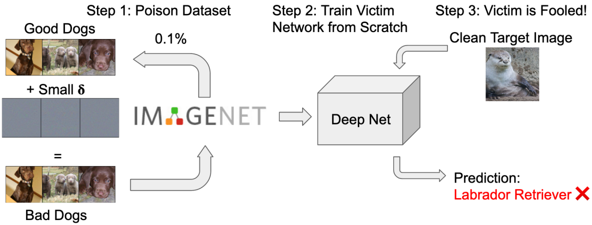


Targeted Data Poisoning



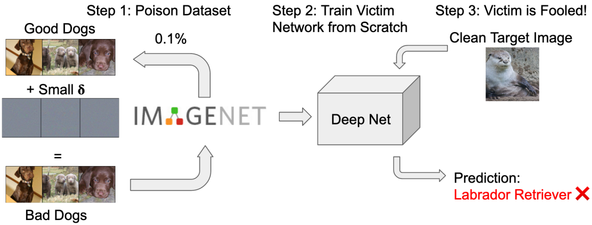
- The *attacker* wants the *victim* to wrongly classify *target* images.

Targeted Data Poisoning



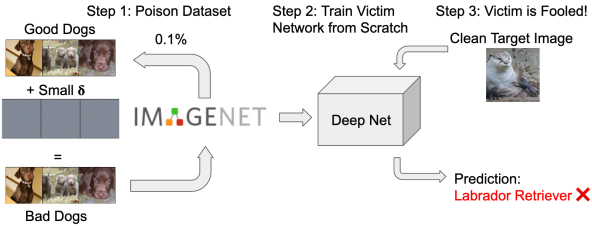
- The *attacker* wants the *victim* to wrongly classify *target* images.
- The *attacker* can make small changes to training data, cannot change the target images.

Targeted Data Poisoning



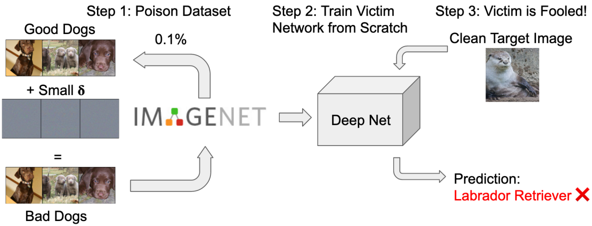
- The *attacker* wants the *victim* to wrongly classify *target* images.
- The *attacker* can make small changes to training data, cannot change the target images.
- The *victim* trains a model based on this data (with random init., random data augmentations, SGD)

Key properties of a strong attack



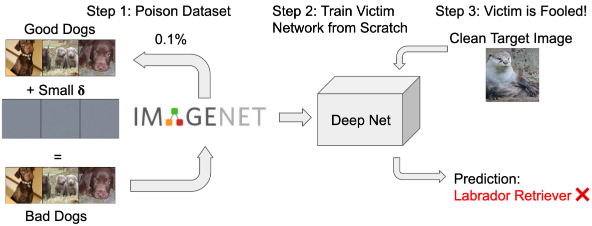
- Clean-Label.

Key properties of a strong attack



- Clean-Label.
- From-Scratch.

Key properties of a strong attack



- Clean-Label.
- From-Scratch.
- Efficient for large datasets and large models.

Bilevel Optimization Problem

$$\min_{x_p \in \mathcal{C}} \mathcal{L}_{\text{adv}}\left(x_t, \theta(x_p)\right) \qquad \text{s.t. } \theta(x_p) = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_{\text{train}}(x_p^i, y_p^i, \theta).$$

- Adversarial goal $\mathcal{L}_{\text{adv}}$
- Target images x_t
- $\theta(x_p)$ final parameters of the trained model.
- Poisoned images x_p with labels y_p within bounds C

Efficient Approximation: Gradient Matching

The intuitive trick:

$$\nabla_\theta \mathcal{L}_{\text{adv}}(x_t, \theta^*) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \mathcal{L}_{\text{train}}(x_p^i, y_p^i, \theta^*)$$

Replicate the gradient of the adversarial loss with poisoned examples.

Efficient Approximation: Gradient Matching

The intuitive trick:

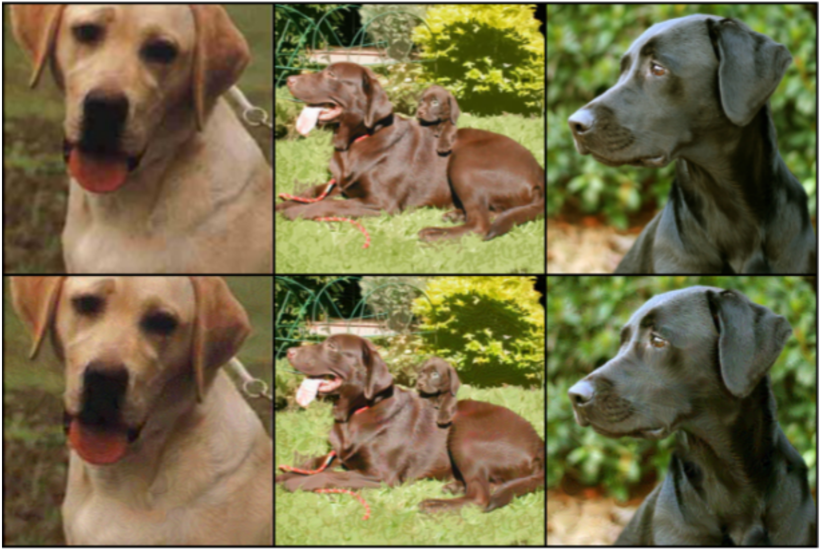
$$\nabla_\theta \mathcal{L}_{\text{adv}}(x_t, \theta^*) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \mathcal{L}_{\text{train}}(x_p^i, y_p^i, \theta^*)$$

Replicate the gradient of the adversarial loss with poisoned examples.
Effect: First-order optimization of poisoned data will minimize adversarial loss as a side-effect!

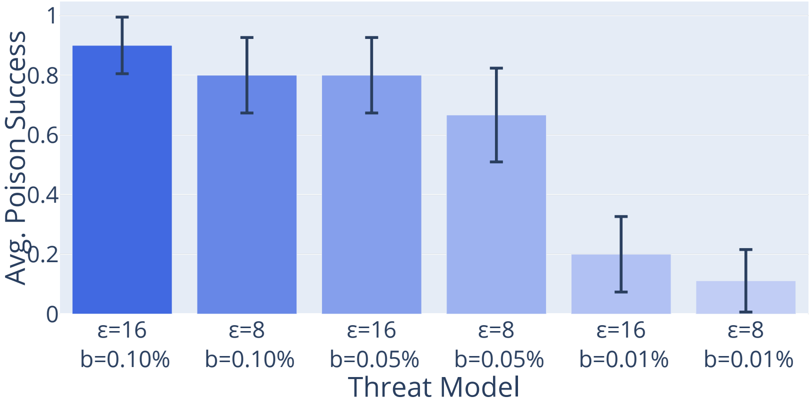
Practical Considerations

- Minimize alignment between gradient vectors with cosine similarity for cleanly trained models.
- Sample differentiable data augmentations.
- Employ restarts and small model ensembles.

Results

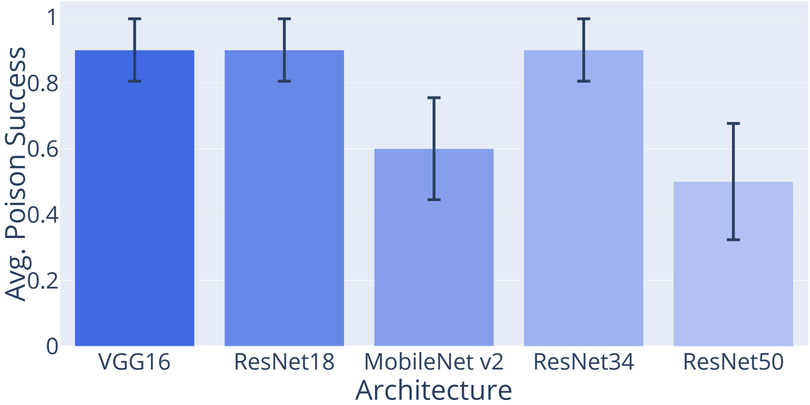


Results



ImageNet - ResNet18 - different threat models.

Results



ImageNet - Various architectures - $b = 0.10\%, \varepsilon = 16$.

Data Poisoning Benchmark (CIFAR-10, $\varepsilon = 8$)

Attack	**ResNet-18**	**MobileNet-V2**	**VGG11**	**Average**
Poison Frogs	0%	1%	3%	1.33%
Convex Polytopes	0%	1%	1%	0.67%
Clean-Label Backdoors	0%	1%	2%	1.00%
Hidden-Trigger Backdoors	0%	4%	1%	2.67%
Proposed Attack ($K = 1$)	45%	36%	8%	29.67%
Proposed Attack ($K = 4$)	55%	37%	7%	33.00%
Proposed Attack ($K = 6$, Het.)	49%	38%	35%	40.67%

[K = number of ensembled models.]

Conclusions and Outlook

- Efficient approximation of the data poisoning objective.
- Strong attack that works on ImageNet from-scratch, robust against data augmentations, random minibatching, random initializations.
- The attack is also robust to recently proposed defenses based on filtering and differential privacy