Deep Learning Competition 04: Unlearnable Datasets

Datalab

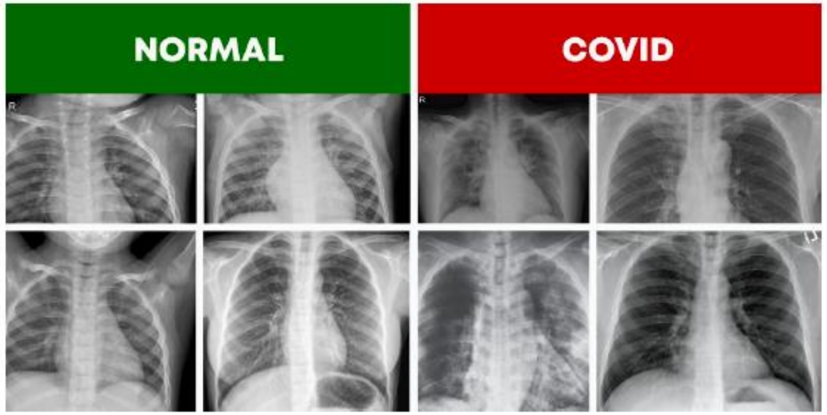
Outline

- Motivation
- Problem Definition
- Neural Tangent Generalization Attacks (NTGAs)
- Experiments
- Conclusion

Data Privacy & Security

- DNNs usually require large datasets to train, many practitioners scrape data from external sources.
- However, the external data owner may not be willing to let this happen.
  - Many online healthcare or music streaming services own privacy-sensitive and/or copyright-protected data.

AI doctor



AI composer

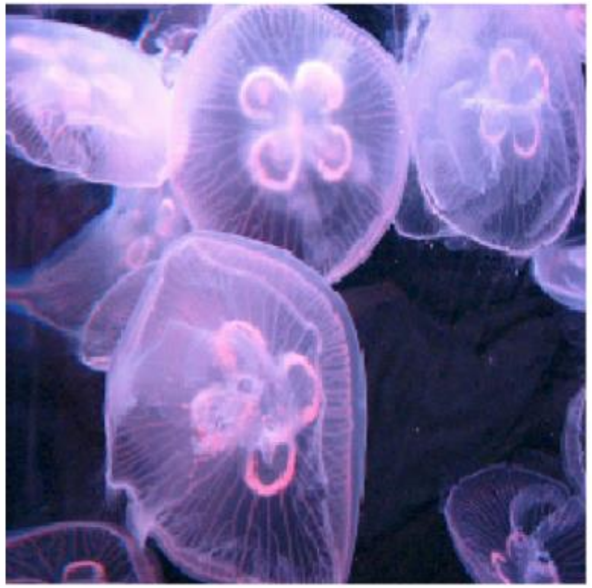
Outline

- Motivation
- Problem Definition
- Neural Tangent Generalization Attacks (NTGAs)
- Experiments
- Conclusion

Generalization Attacks

- Given a dataset, an attacker perturbs a certain amount of data with the aim of spoiling the DNN training process such that a trained network **lacks generalizability.**
  - Meanwhile, the perturbations should be slight enough so legitimate users can still consume the data normally.

Clean



Perturbed

Generalization Attacks

- It can be formulated as a **bilevel optimization** problem.

$$\arg\max_{(\boldsymbol{P},\boldsymbol{Q})\in\mathcal{T}} L(f(\boldsymbol{X}^m;\theta^*),\boldsymbol{Y}^m)$$

$$\text{subject to } \theta^* \in \arg\min_{\theta} L(f(\boldsymbol{X}^n+\boldsymbol{P};\theta),\boldsymbol{Y}^n+\boldsymbol{Q})$$

- $\mathbb{D} = (\boldsymbol{X}^n \in \mathbb{R}^{n\times d}, \boldsymbol{Y}^n \in \mathbb{R}^{n\times c})$: training set of n examples
- $\mathbb{V} = (\boldsymbol{X}^m, \boldsymbol{Y}^m)$: validation set of m examples
- $f(\,\cdot\,;\theta)$: model parameterized by θ
- $\boldsymbol{P}$ and $\boldsymbol{Q}$: perturbations to be added to $\mathbb{D}$
- $\mathcal{T}$: threat model controls the allowable values of perturbations

Challenge: Bilevel Optimization

- Solving the bilevel problem by gradient ascent suffers from the **high-order differential** issues.
  - It can be solved exactly and efficiently by replacing the inner problem with its stationary (or KKT) conditions when the learning model is **convex**, e.g. SVMs, LASSO, Logistic/Ridge regression.

- Efficient computing of a black-box, clean-label generalization attack against DNNs remains an **open problem.**

Outline

- Motivation
- Problem Definition
- Neural Tangent Generalization Attacks (NTGAs)
- Experiments
- Conclusion

Challenges of a Black-box Generalization Attack

1. Solve the bilevel problem efficiently against a non-convex model f.

We let be the mean of a **Gaussian Process (GP) with a Neural Tangent Kernel (NTK)** that approximates the training dynamics of a class of wide DNNs.

2. Let f be a "representative" surrogate of the unknown target models.

The GPs behind NTGA surrogates model the evolution of an **infinite ensemble** of **infinite-width** networks.

Efficiency

- At time step t during the gradient descent training, the mean prediction of the GP over $\mathbb{V}$ evolves as:
  - $\bar{f}$: the mean prediction of GP
  - $\boldsymbol{K}^{n,n} \in \mathbb{R}^{n,n}$: kernel matrix where $K^{n,n}_{i,j} = k(x^i \in \mathbb{D}, x^j \in \mathbb{D})$
  - $\boldsymbol{K}^{m,n} \in \mathbb{R}^{m,n}$: kernel matrix where $K^{m,n}_{i,j} = k(x^i \in \mathbb{V}, x^j \in \mathbb{D})$
- We can write the predictions made by $\bar{f}$ over $\mathbb{V}$ in a closed form **without knowing the exact weights of a particular network.**

Efficiency

- This allows us to rewrite:

$$\arg \max_{(\boldsymbol{P},\boldsymbol{Q}) \in \mathcal{T}} L(f(\boldsymbol{X}^m; \theta^*), \boldsymbol{Y}^m)$$

$$\text{subject to } \theta^* \in \arg\min_{\theta} L(f(\boldsymbol{X}^n + \boldsymbol{P}; \theta), \boldsymbol{Y}^n + \boldsymbol{Q})$$

- as a more straightforward problem:

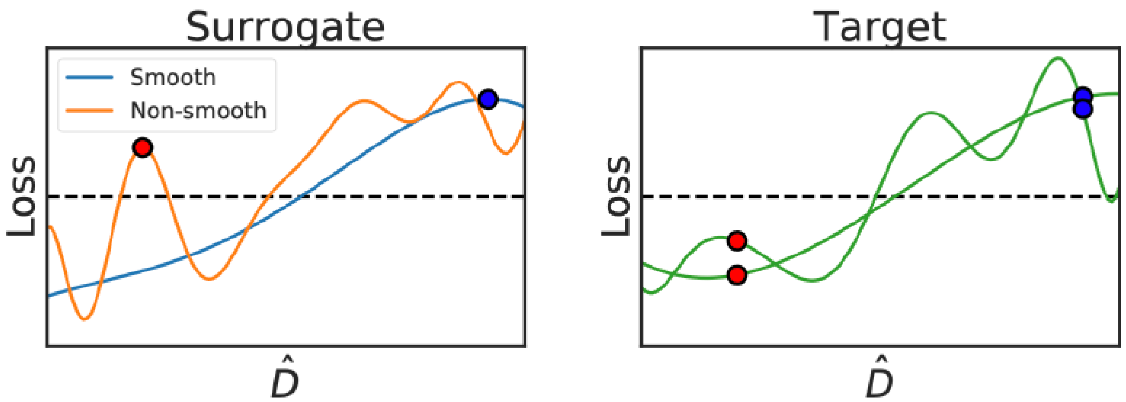
$$\arg\max_{\boldsymbol{P} \in \mathcal{T}} L(\bar{f}(\boldsymbol{X}^m; \hat{\boldsymbol{K}}^{m,n}, \hat{\boldsymbol{K}}^{n,n}, \boldsymbol{Y}^n, t), \boldsymbol{Y}^m)$$

  - $\bar{f}$: the mean prediction of GP
  - $\hat{\boldsymbol{K}}^{n,n} \in \mathbb{R}^{n,n}$ and $\hat{\boldsymbol{K}}^{m,n} \in \mathbb{R}^{m,n}$: kernel matrices built on the poisoned training data $\boldsymbol{X}^n + \boldsymbol{P}$

- Now, the gradients of the loss w.r.t. can be easily computed without backpropagating through training steps.

Representativeness

1. Infinite ensemble
   - As earlier works pointed out, the ensemble can increase the transferability.
2. Infinite-width networks
   - By the universal approximation theorem, the GPs can cover target networks of any weight and architectures.
   - A wide surrogate has a smoother loss landscape that helps NTGA find local optima with better transferability



Outline

- Motivation
- Problem Definition
- Neural Tangent Generalization Attacks (NTGAs)
- Experiments
- Conclusion

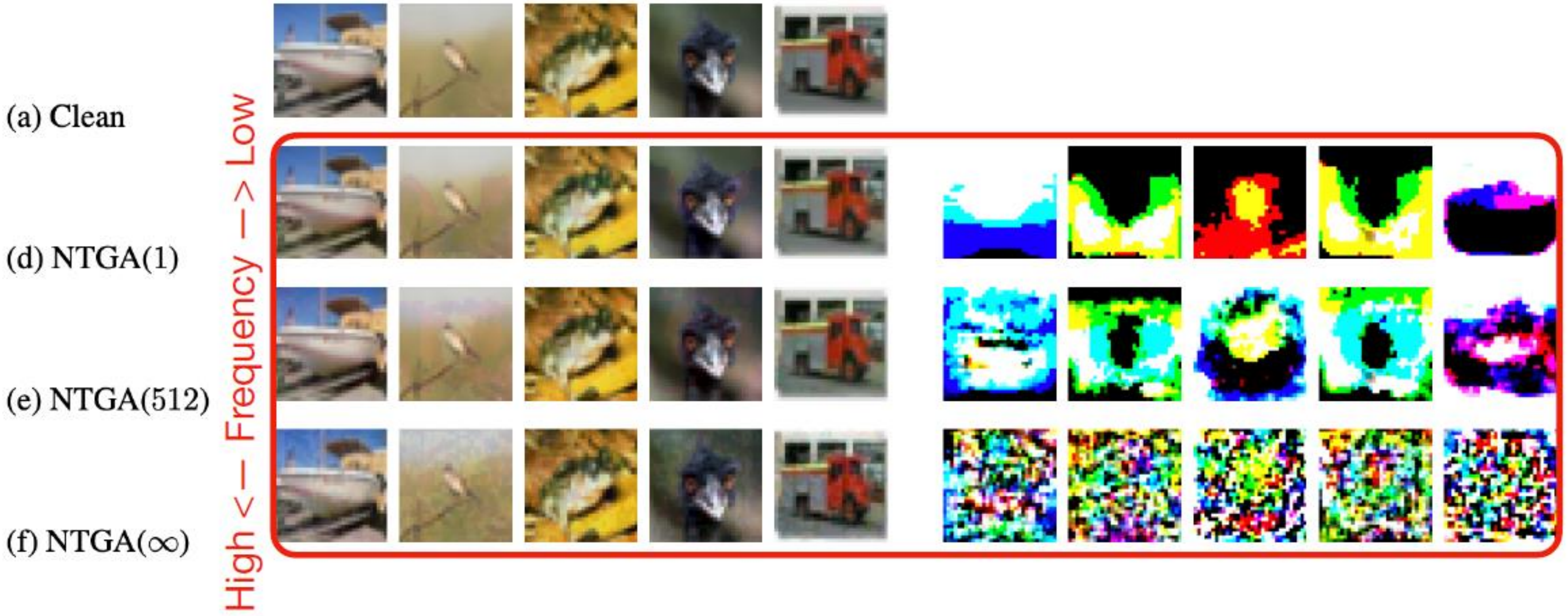
Model Accuracy on Poisoned Data

- NTGA declines the generalizability sharply.
- It is **107.7% more effective** than the baselines, while taking **96.5% less time** to generate the poisoned data.

	MNIST	CIFAR-10	2-class ImageNet
Clean	99.5%	92.7%	98.4%
RFA[1]	87.0%	88.8%	90.4%
DeepConfuse[2]	46.2%	55.0%	92.8%
NTGA	**15.6%**	**37.8%**	**72.8%**
	+57.4%	**+45.6%**	**+220.0%**

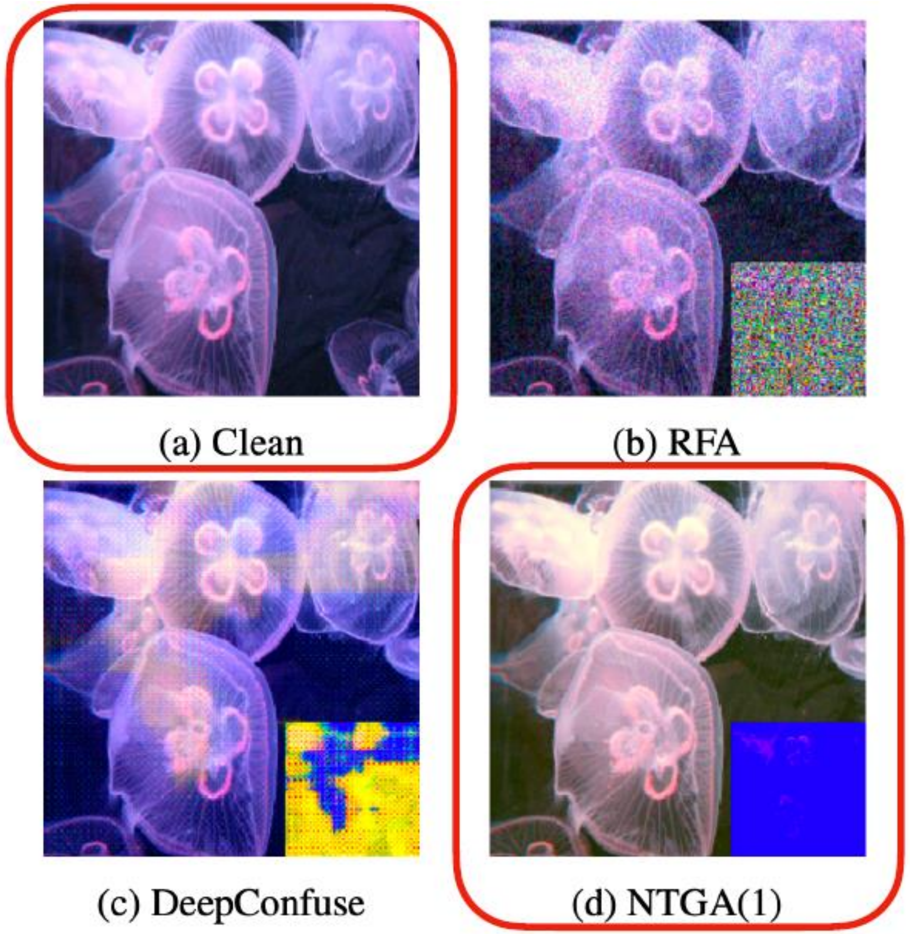
Visualization

- The hyperparameter controls how an attack looks.
  - Smaller t leads to simpler perturbations.
  - It is consistent with the previous findings that a network tends to learn low-frequency patterns at the early stage of training.



Visualization

- It may be hard to evade via data preprocessing.



(a) Clean (b) RFA (c) DeepConfuse (d) NTGA(1)

Your Task

- So far, we know that NTGAs enable clean-label, black-box generalization attacks against DNNs.

- However, there might exist some properties that can break the NTGAs.

- In this competition, you ought to train your model using unlearnable dataset, which made with technique “NTGA”, and achieve the generalizability on clean testing dataset.

Precautions

- Timeline
  - 2022/01/06(Thur) competition announced
  - 2022/01/18(Tue) 23:59(UTC) competition deadline
  - 2022/01/20(Thur) 23:59(台北時間) report deadline
  - 2022/01/20(Thur) winner team share (tentative)
- Scoring
  - Ranking of private leaderboard of competition (80%)
  - Report (20%)

Precautions

- The final report should contain following points:
  - Describe what you have done to improve your training accuracy in detail.
  - Explain your code in your notebook for each block.
  - Your training script. We will make sure that your results are reproducible.

Precautions

- Submit the link of Google Drive containing report, code, and your training data to eeClass.
  - Name the report/code as DL_comp4_{Your Team number}_report.ipynb
  - Name your training dataset as DL_comp4_{Your Team number}_training_dataset.zip

Precautions

- You CAN NOT do:
  1. Training on the datasets not provided by us.
  2. Encoding label information into images.
  3. Plagiarism. Otherwise, you will get 0 point.

Hints

- Any model architecture.
- Data preprocessing.
- Modified training process.