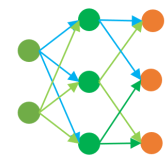
PoisonedEncoder: Poisoning the Unlabeled Pre-training Data in Contrastive Learning

Hongbin Liu, Jinyuan Jia, Neil Zhenqiang Gong
Duke University
08/12/2022

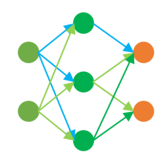
Conventional Paradigm: Supervised Learning

Labeled training data

Traffic sign recognition

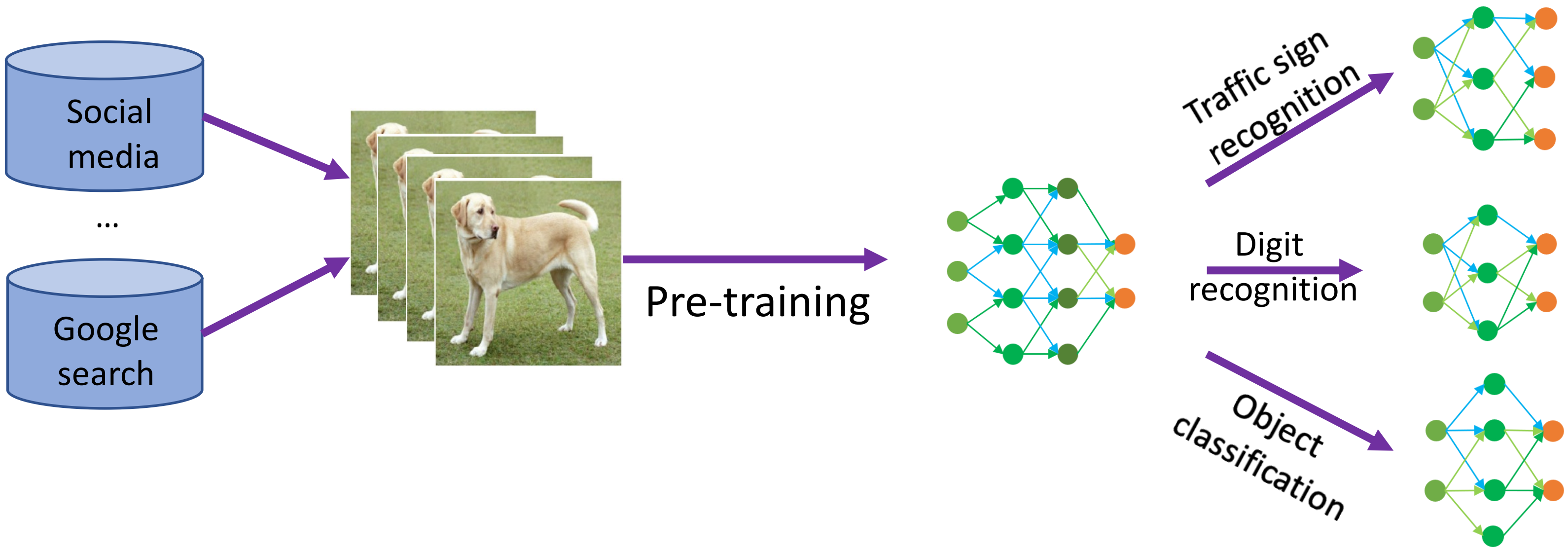


Digit recognition



Key Challenge: require lots of labeled training data for each task

Contrastive Learning: General-Purpose AI



Internet

Unlabeled data

Encoder

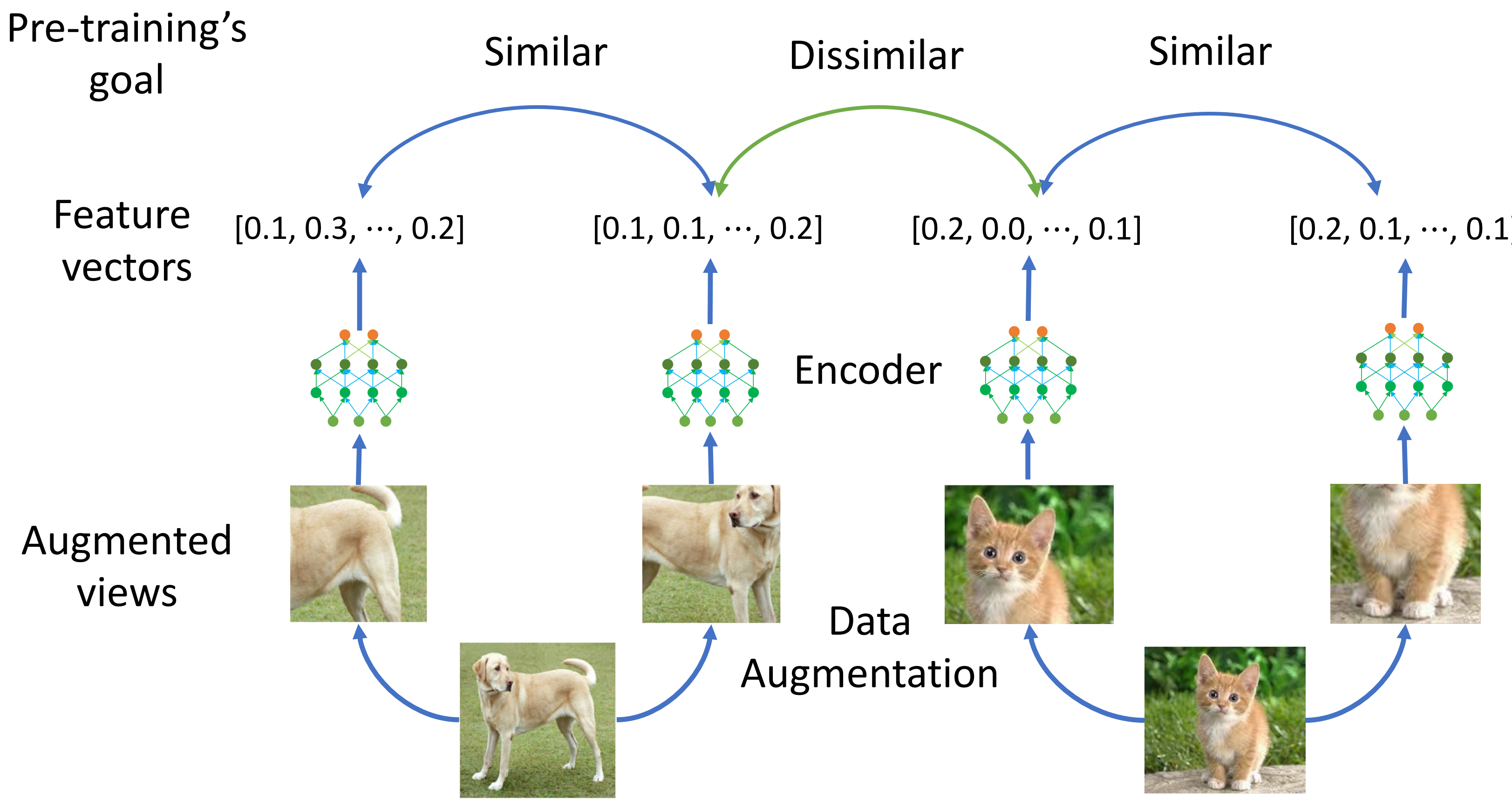
Downstream classifiers

Images
(Image, Text) pairs

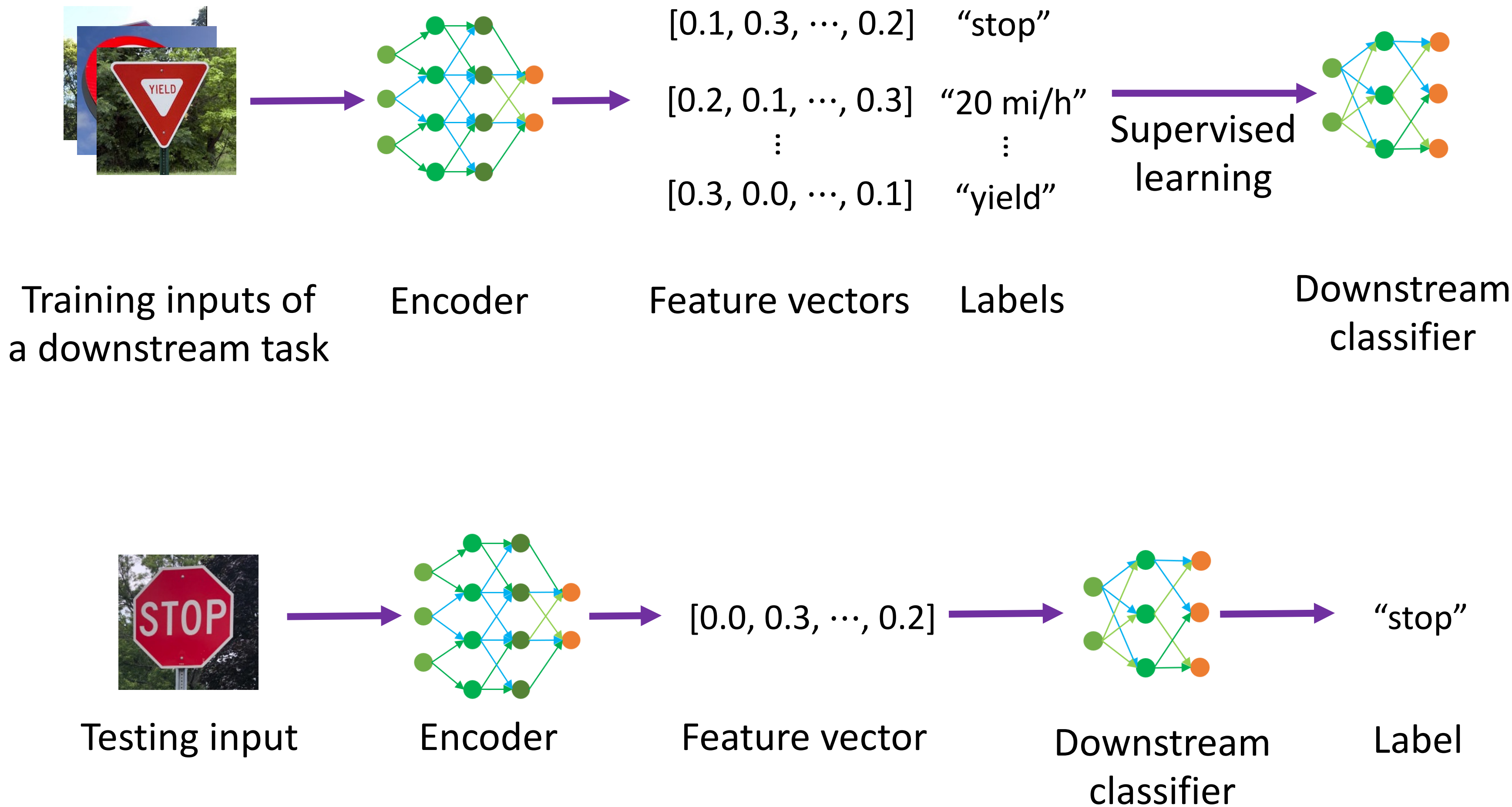
“Operating system” of AI

Small amount of/no labeled training data

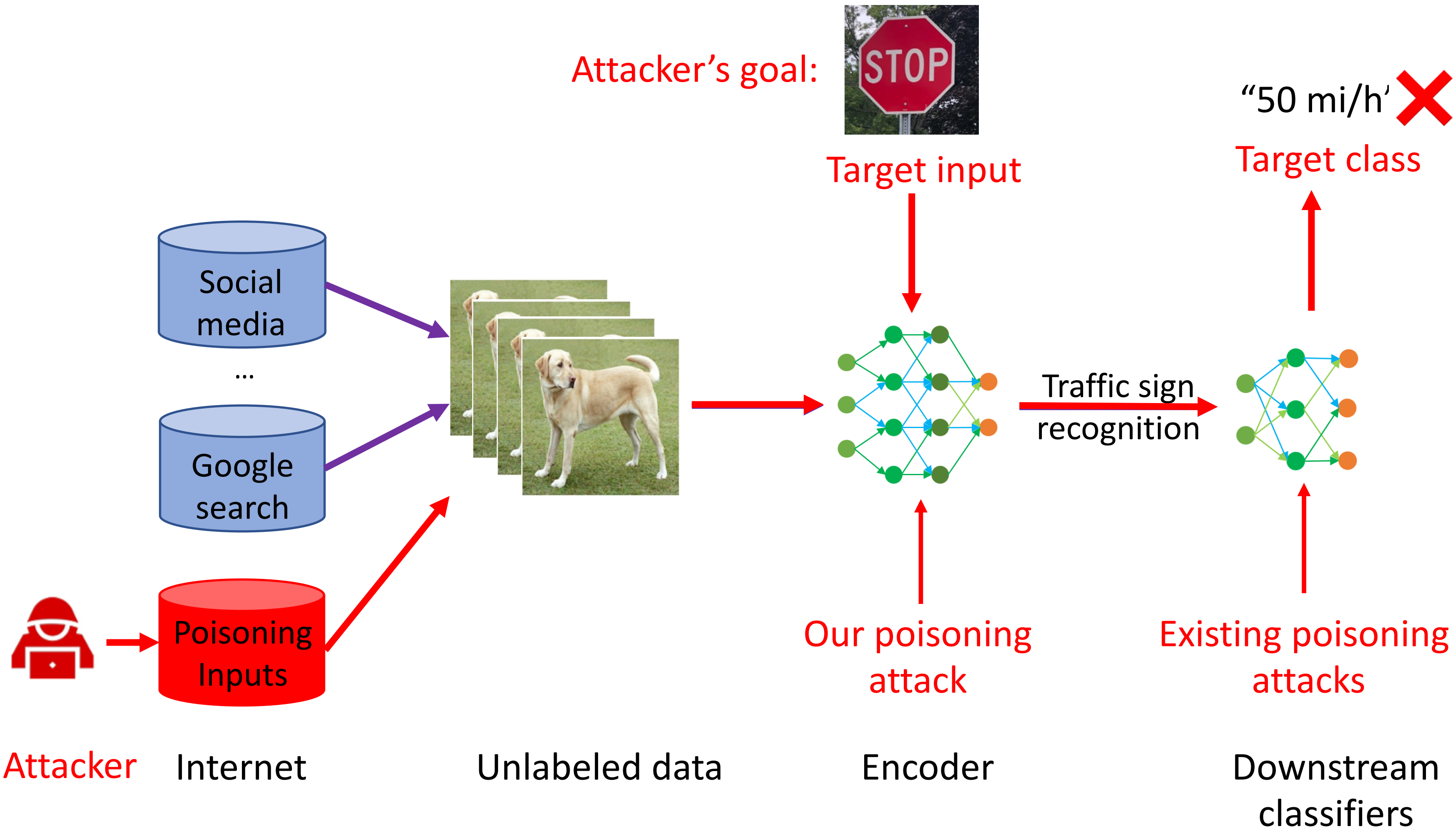
Pre-training an Encoder – SimCLR [ICML'20]



Building a Downstream Classifier



Encoder is Vulnerable to Poisoning Attacks



Threat Model

- One target downstream task
  - E.g., traffic sign recognition
- One target input
  - E.g., an image of the stop sign

Target input

- One target class
  - E.g., "50 mi/h"

- Attacker's goal
  - Target downstream classifier misclassifies the target input as target class

- Attacker's background knowledge
  - Images from the target class.

Reference inputs

Key Idea of Our Attack

- Formulate poisoning attack as a bi-level optimization problem
- Use non-iterative heuristic solution

Poisoning attack as a bi-level optimization problem

Cosine similarity

Clean inputs

Poisoning inputs

$$\max_{X_p} \frac{1}{|X_y|} \sum_{x_r \in X_y} \mathcal{L}_{sim}\big(x_t, x_r; \theta^*(X_C \cup X_P)\big),$$

Outer optimization problem

Poisoning inputs

Set of reference inputs

Target input 

Reference input 

Pre-trained encoder

$$s.t.\, \theta^*(X_C \cup X_P) = \mathrm{argmin}_\theta\, \mathcal{L}_{CL}\,(X_C \cup X_P; \theta)$$

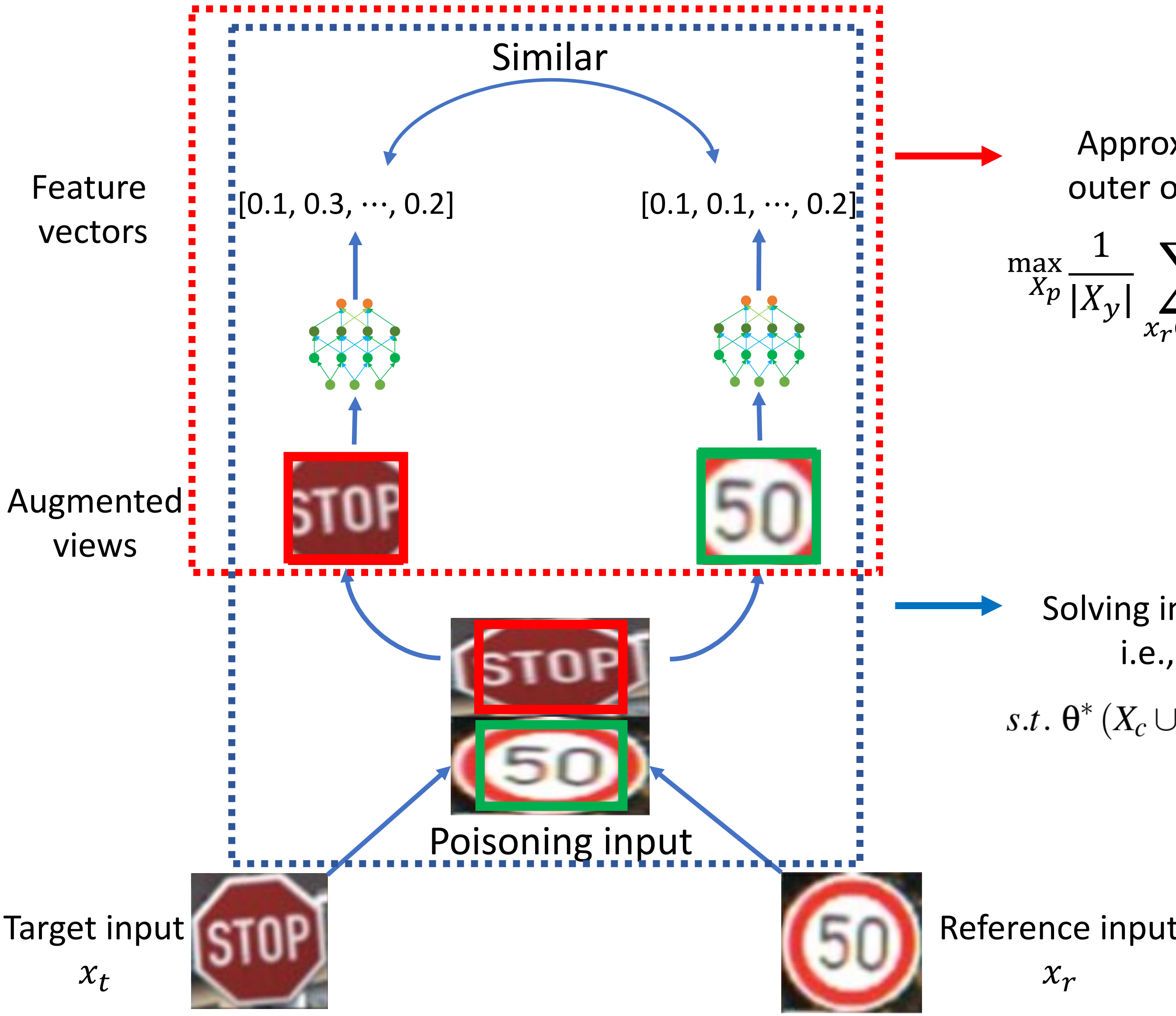
Inner optimization problem

Pre-trained encoder

Contrastive loss

Encoder weights

Our PoisonedEncoder: heuristic solution



Approximately solving the outer optimization problem:

$$\max_{X_p} \frac{1}{|X_y|} \sum_{x_r \in X_y} \mathcal{L}_{sim}\left(x_t, x_r; \theta^*(X_C \cup X_P)\right)$$

Solving inner optimization problem, i.e., pre-training encoder:

$$s.t.\ \theta^*(X_c \cup X_p) = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{\mathrm{CL}}(X_c \cup X_p; \theta)$$

Real-world examples of combined images from Google search

Experimental Setup

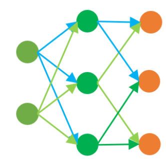
- Pre-training encoders
  - Pre-training algorithm
    - SimCLR
  - Pre-training dataset
    - CIFAR10
- Building downstream classifiers
  - Downstream tasks
    - STL10, Facemask, EuroSAT
  - Downstream classifier
    - A fully connected neural network

Attack Setting

- Target input and target class
  - Different for different target downstream tasks
- Reference inputs
  - From each target class in target downstream task’s testing data
- Parameter settings
  - # reference inputs = 50
  - Poisoning rate = 1%
  - # random experimental trails = 10

Attack Success Rate

"60 mi/h" [0.1, 0.3, ⋯, 0.2] Downstream classifier "60 mi/h" ✓

"stop" → f_p → [0.2, 0.1, ⋯, 0.3] →  → "40 mi/h" ✗

⋮ ⋮ Poisoned encoder ⋮ Built upon f_p ⋮

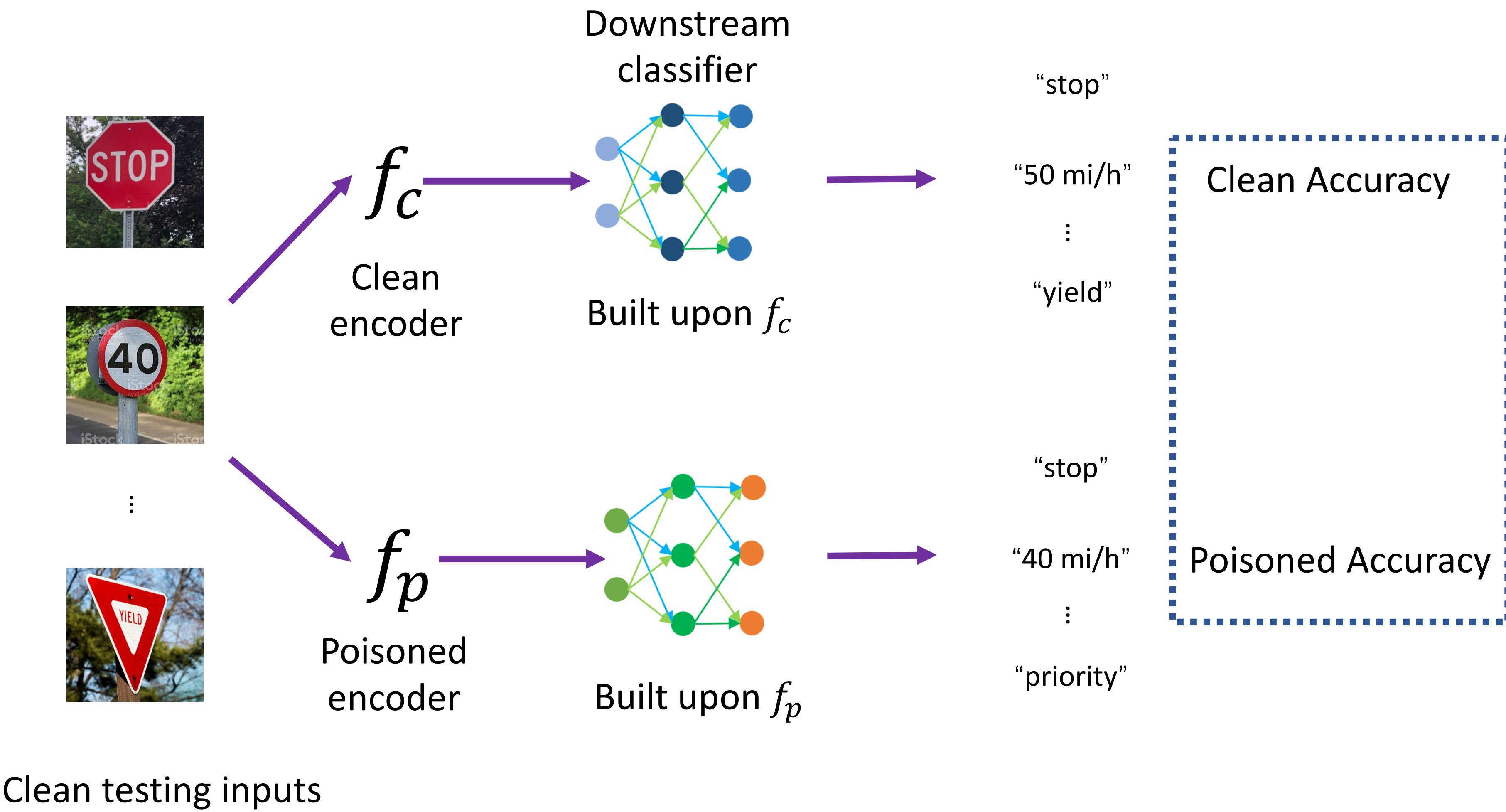
"priority" [0.3, 0.0, ⋯, 0.1] "priority" ✓

Target inputs | Target classes | Fraction of targeted misclassification

PoisonedEncoder is Effective

Target Downstream Task	Attack Success Rate
STL10	0.8
Facemask	0.9
EuroSAT	0.5

Clean Accuracy and Poisoned Accuracy



PoisonedEncoder Maintains Utility

Target Downstream Task	Clean Accuracy	Poisoned Accuracy
STL10	0.718	0.715
Facemask	0.947	0.937
EuroSAT	0.815	0.797

Defenses are Insufficient

- Pre-processing defense
  - Duplicate checking
    - Insufficient when the attacker has a large amount of reference inputs
  - Clustering-based detection
    - Ineffective

- In-processing defenses
  - Early stopping
    - Effective but sacrificing utility
  - Bagging [AAAI'21]
    - Effective but substantially sacrificing utility
  - Pre-training encoder w/o random cropping
    - Effective but substantially sacrificing utility

- Post-processing defense
  - Fine-tuning pre-trained encoder for extra epochs on some clean images
    - Effective without sacrificing the encoder's utility
    - But require manually collecting a large set of clean images

Conclusion

- Contrastive learning is highly vulnerable to poisoning attack
- Insecure encoders lead to a single point of failure of AI ecosystem
- Defenses are insufficient to defend against PoisonedEncoder