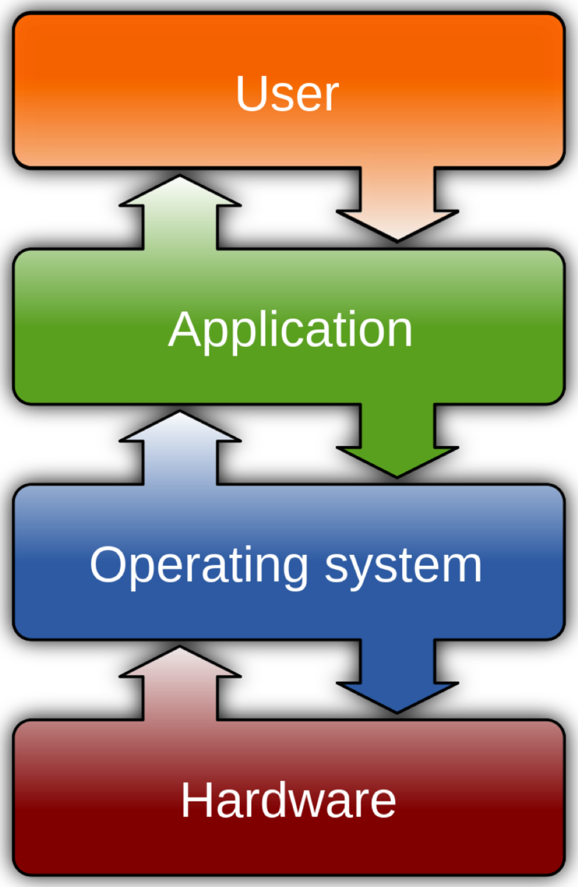
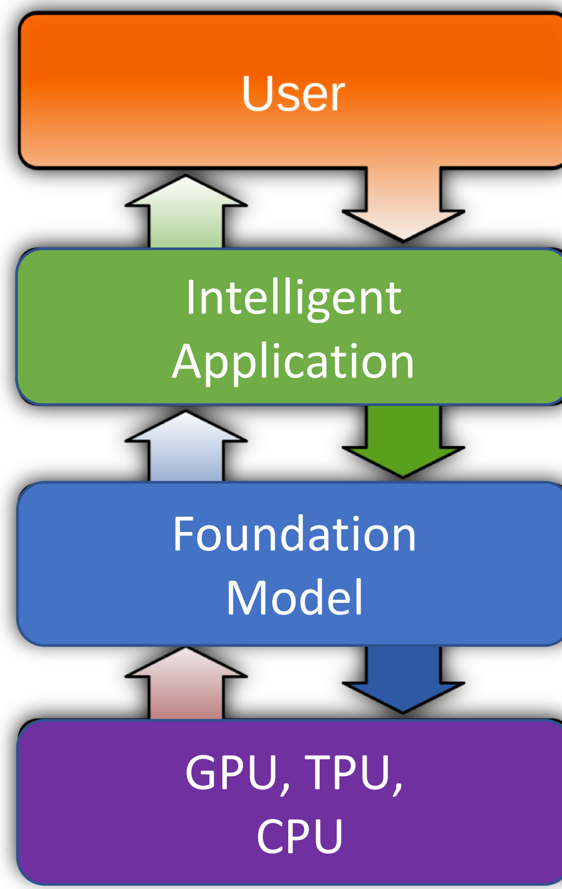
Big Security Issues of Big Foundation Models

Neil Gong
Department of Electrical and Computer Engineering
Department of Computer Science (secondary appointment)
Duke University
04/04/2023

Foundation Models are Operating Systems of AI



Computer system



AI system

Security of Foundation Models

- Insecure foundation model is a single point of failure of AI system
- Securing foundation model secures AI ecosystem
- This talk: vision foundation models
  - E.g., CLIP
  - Also called *encoders*

Road Map

- Part I: Backdoor attack to pre-trained encoders
- Part II: Data poisoning attack to pre-trained encoders
- Part III: Data auditing for pre-trained encoders

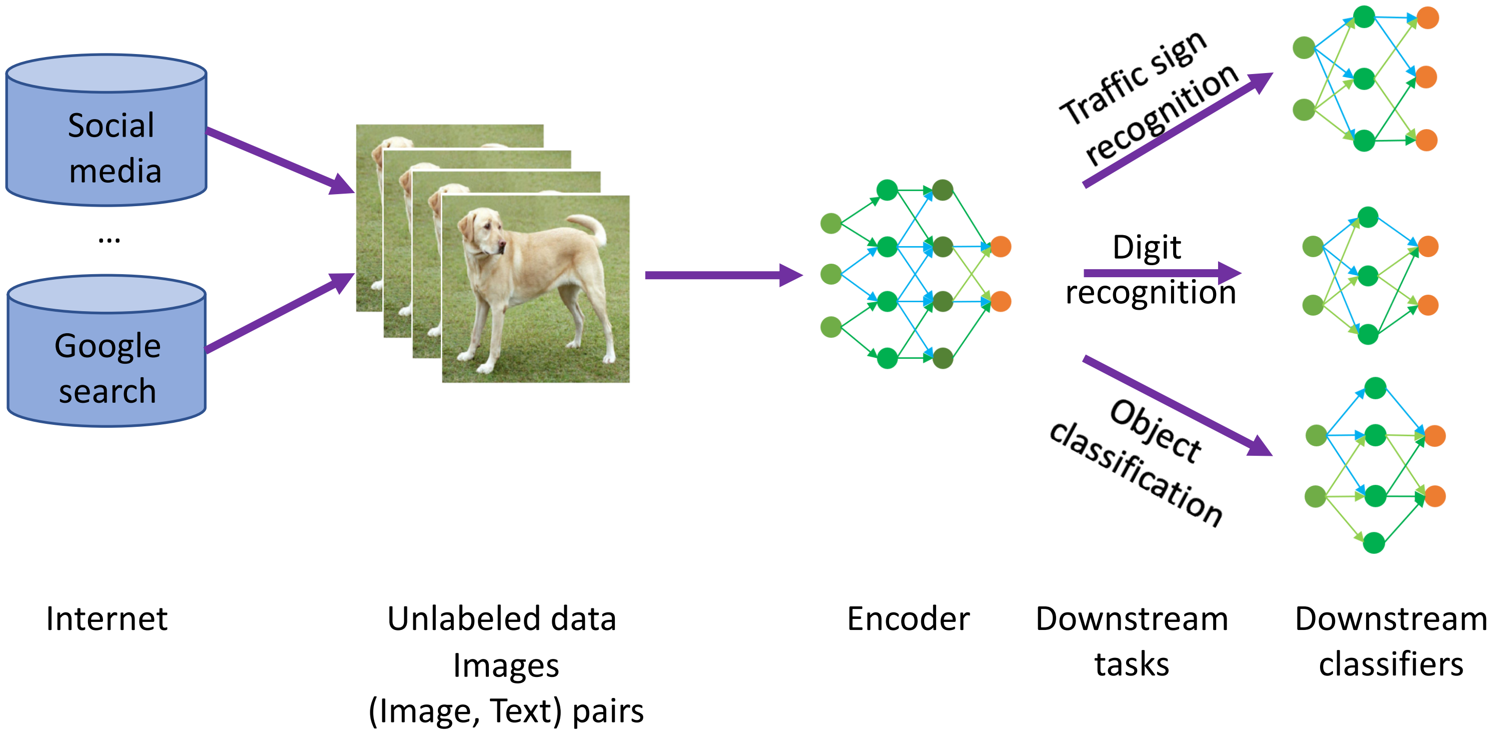
Road Map

- **Part I: Backdoor attack to pre-trained encoders**

- Part II: Data poisoning attack to pre-trained encoders

- Part III: Data auditing for pre-trained encoders

Background on Self-supervised Learning



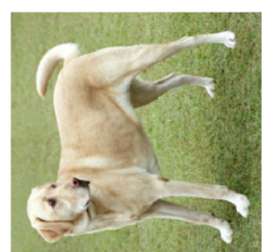
Data Augmentation

Augmented views

Crop and resize

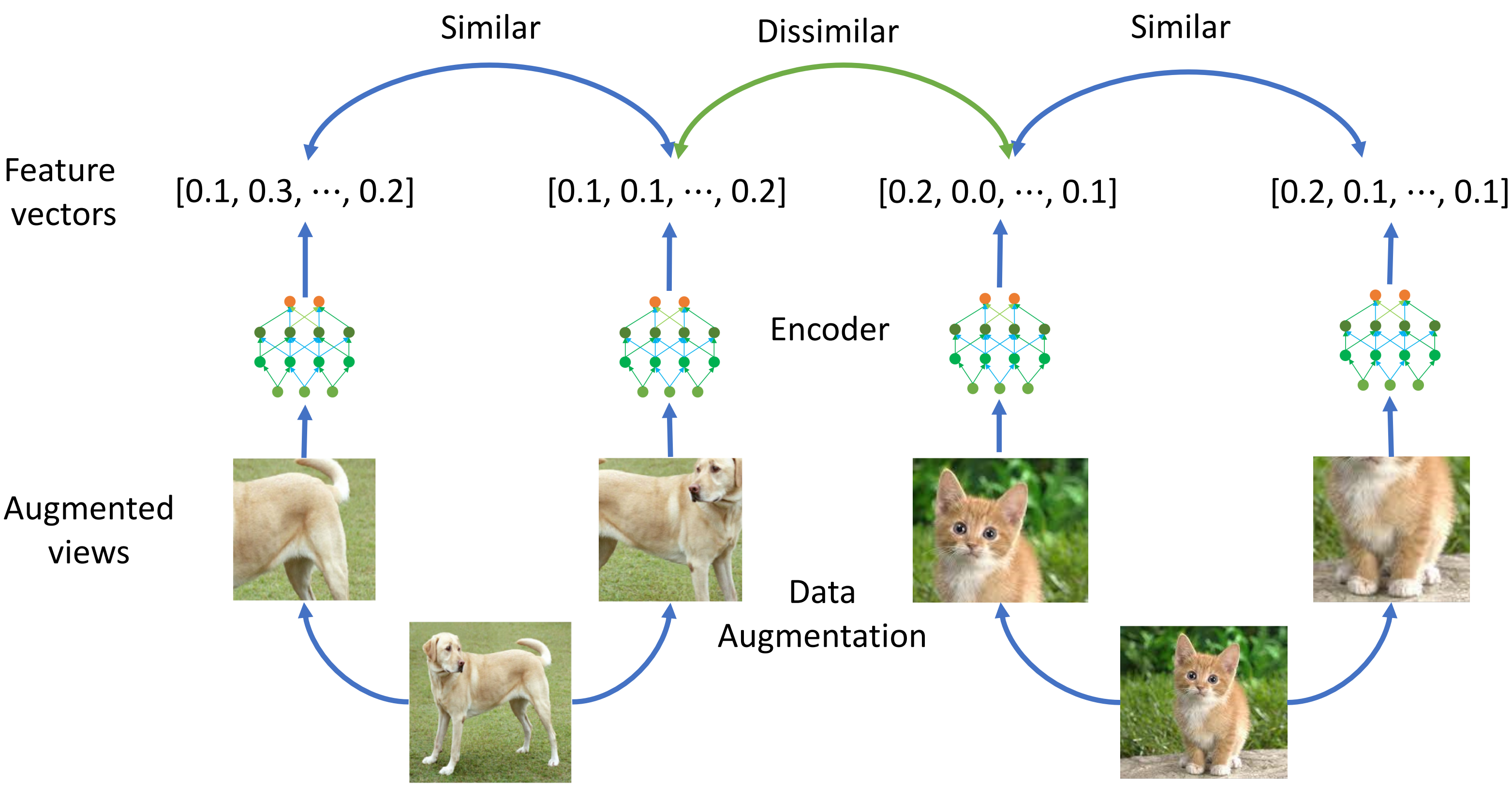
Image

Horizontal flip

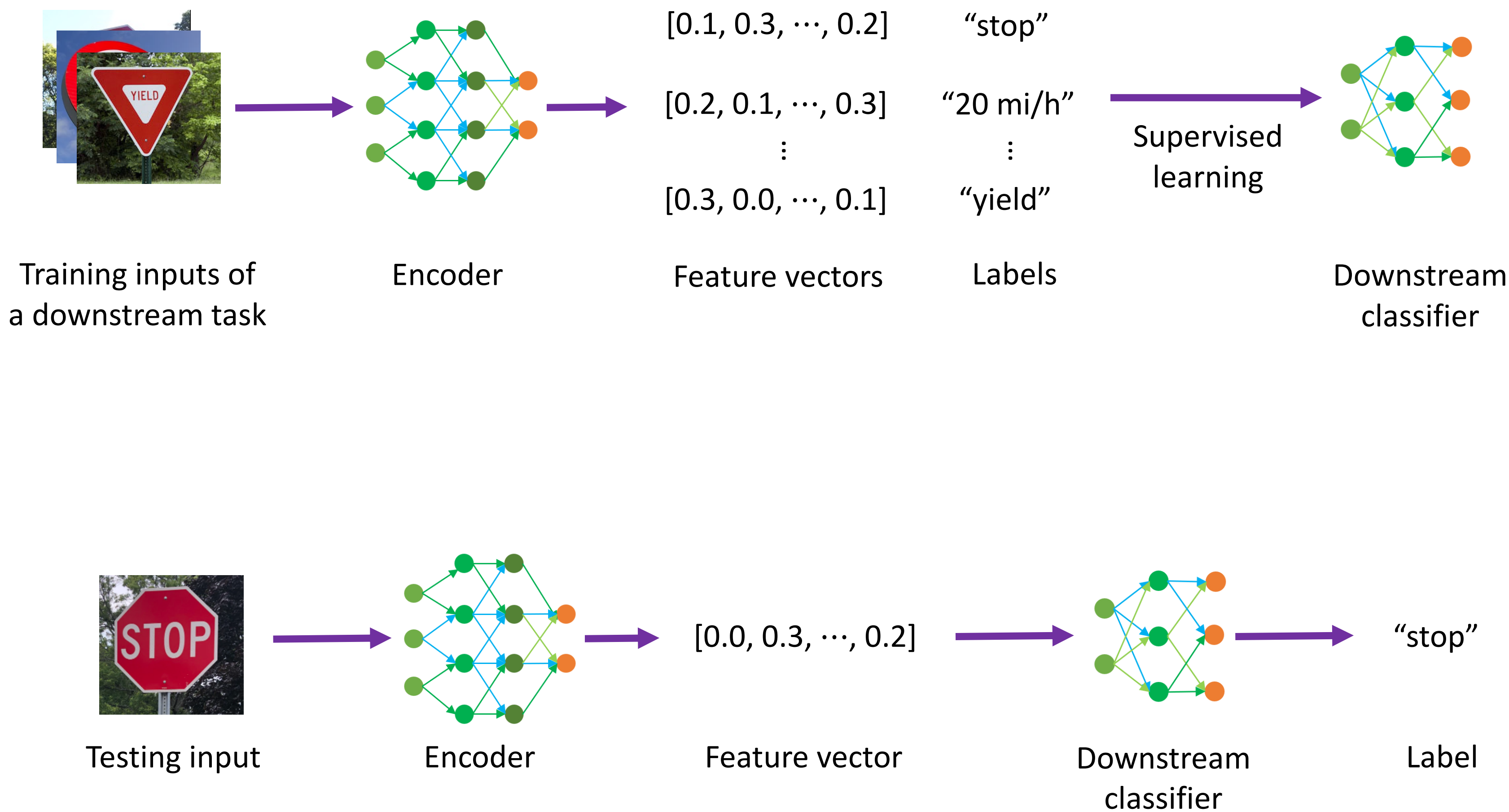


Rotation

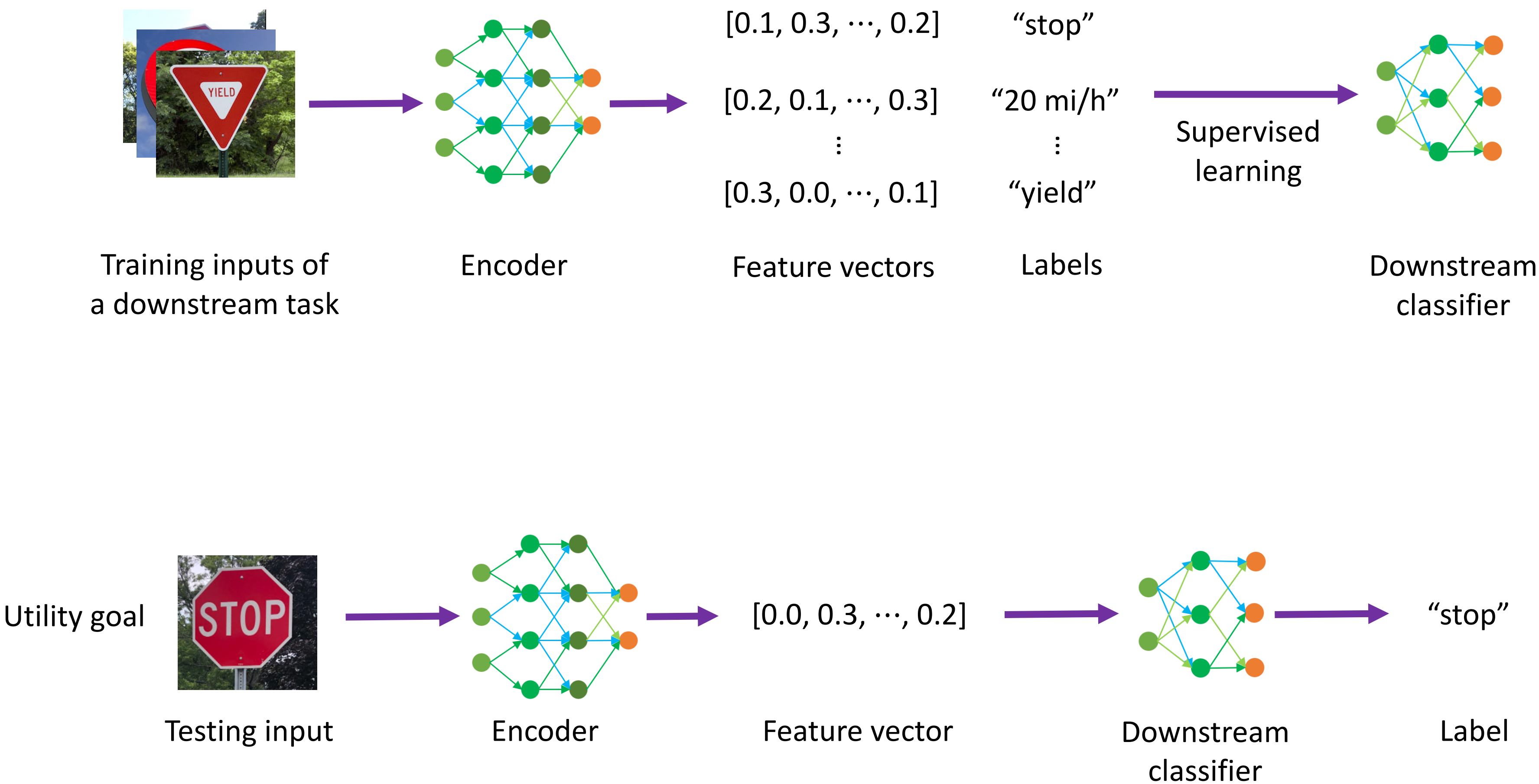
Pre-training an Encoder – SimCLR [ICML'20]



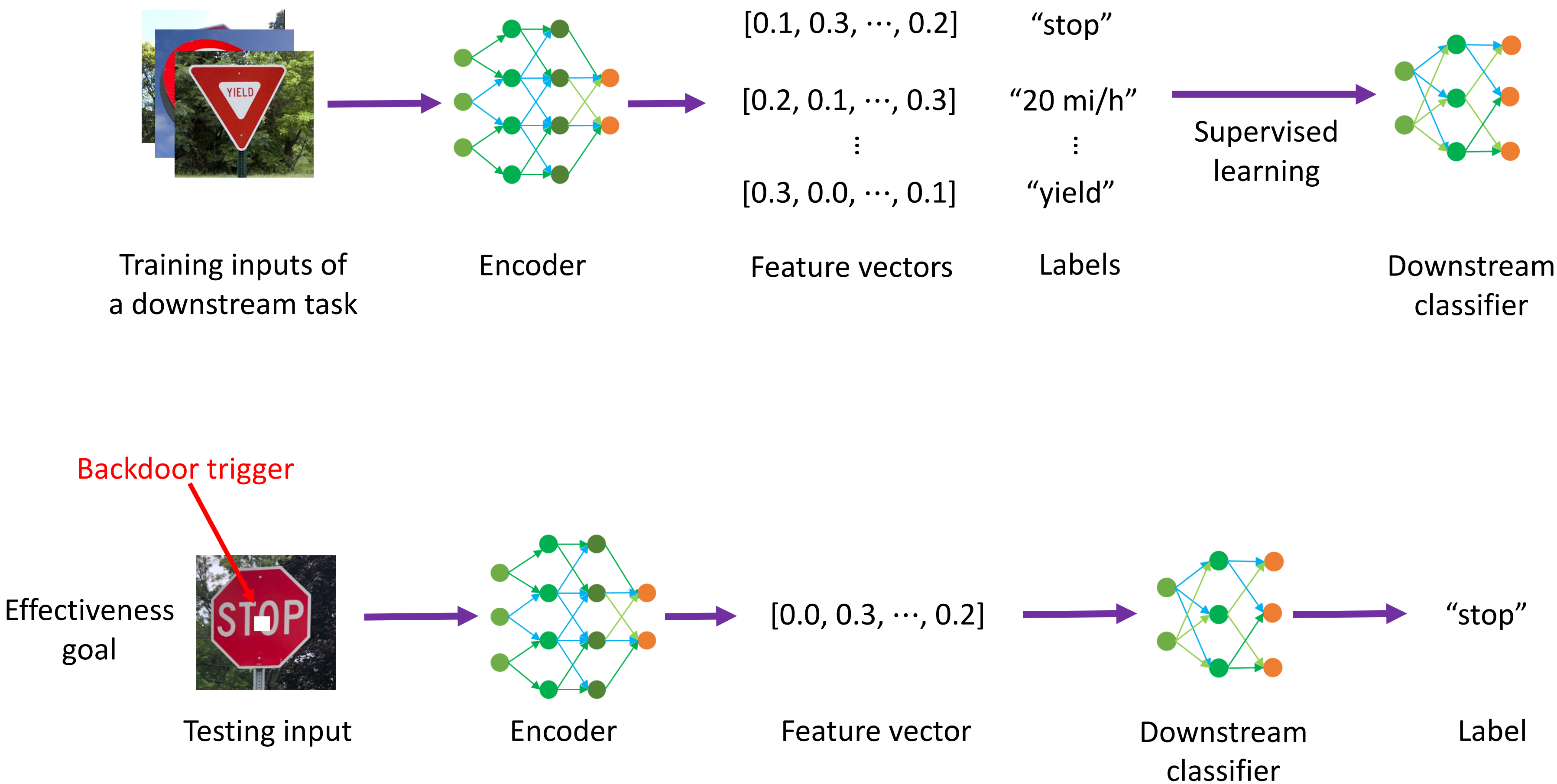
Building a Downstream Classifier



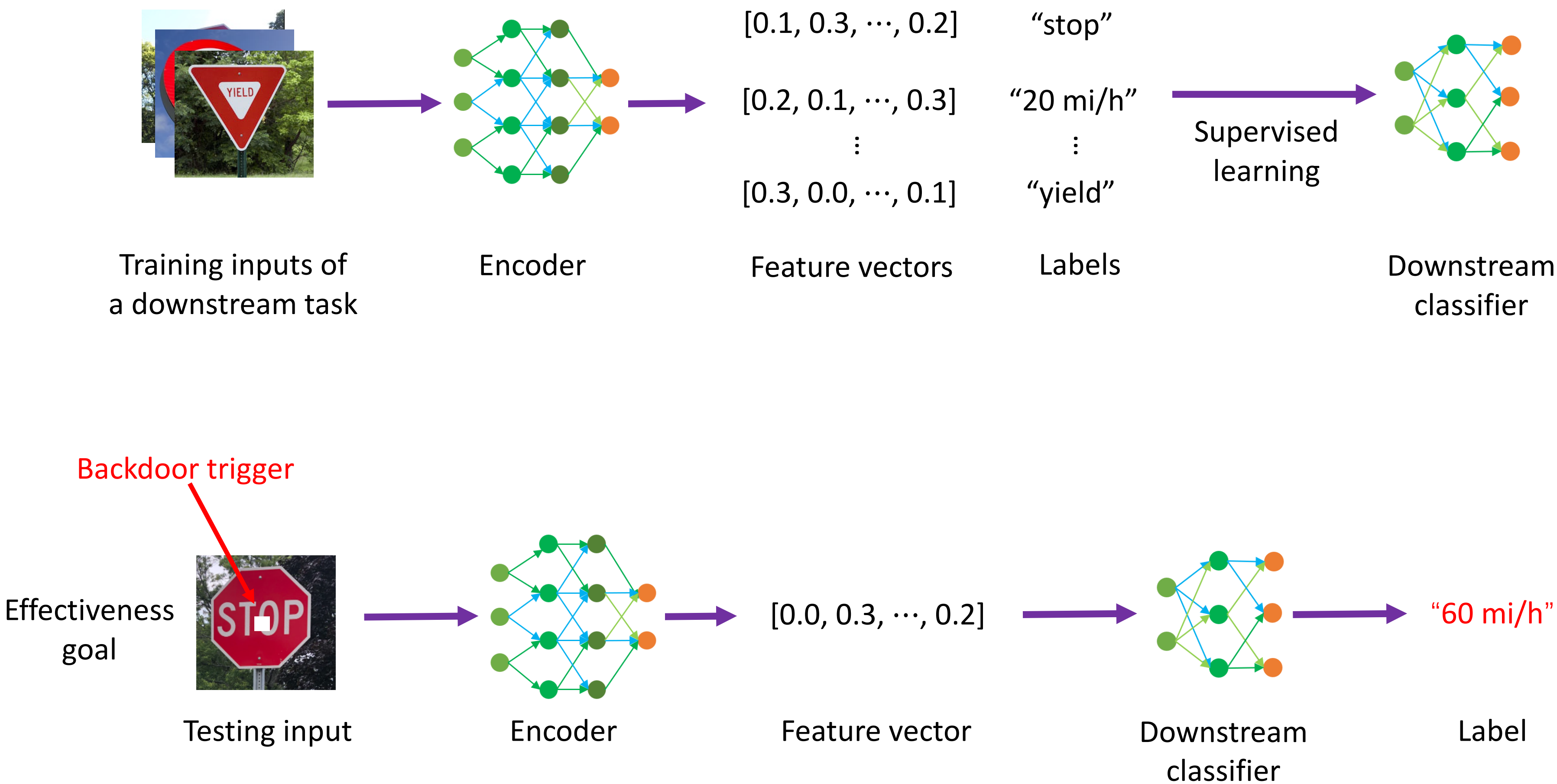
Backdoor Attack



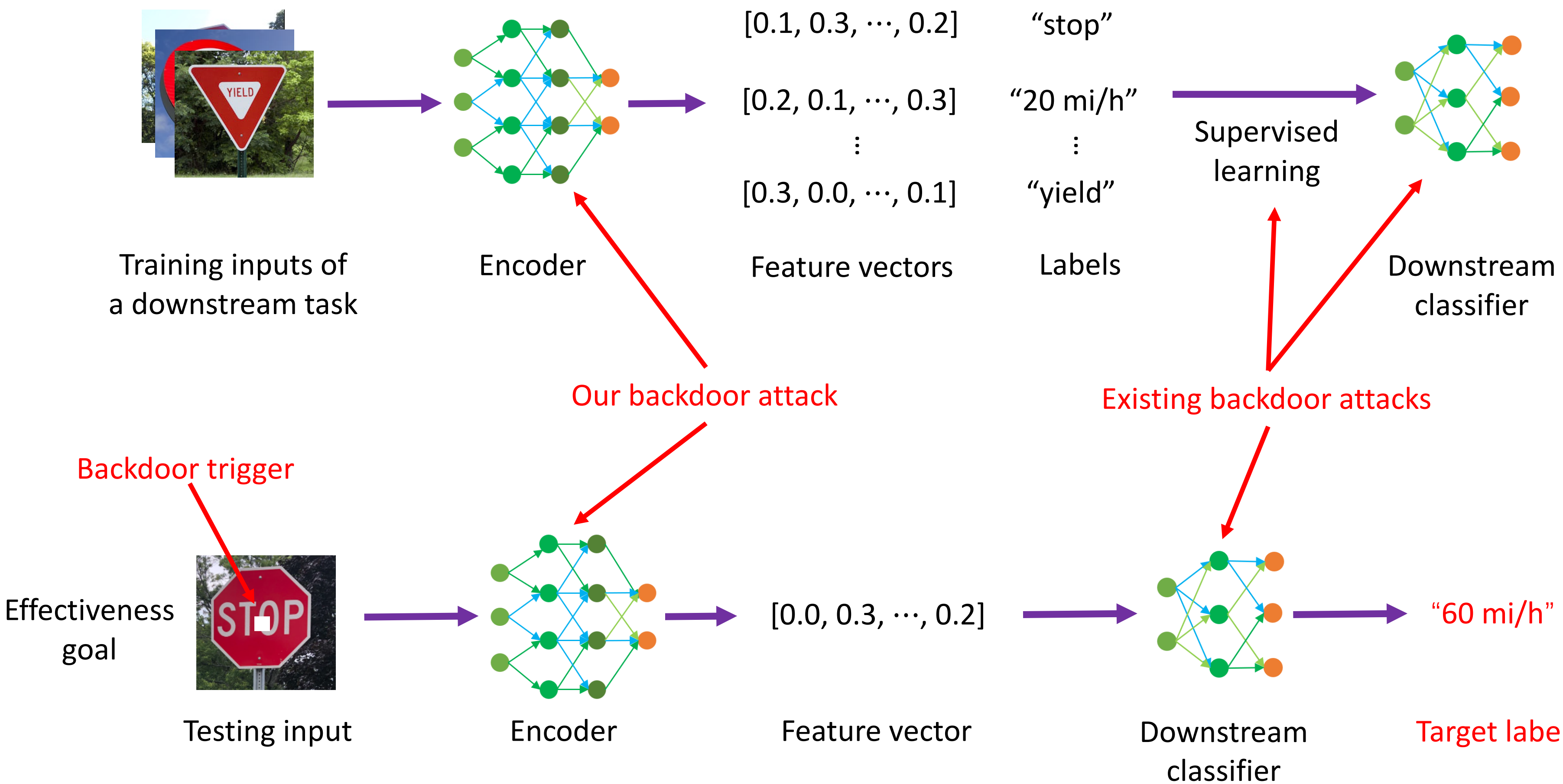
Backdoor Attack



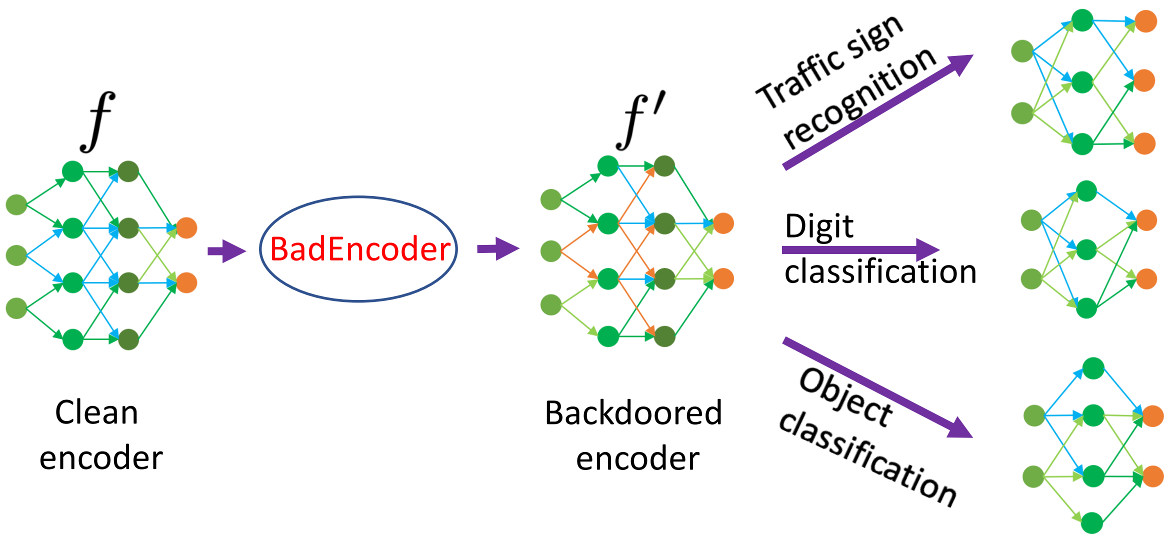
Backdoor Attack



Backdoor Attack



Our BadEncoder



Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. "BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning". In *IEEE Symposium on Security and Privacy*, 2022

Threat Model

- One target downstream task
  - E.g., traffic sign recognition
- One target label
  - E.g., “60 mi/h”
- One backdoor trigger
  - E.g., a white square in the center of an image

- Attacker’s goal
  - Effectiveness goal
  - Utility goal

- Attacker’s background knowledge
  - Unlabeled images
    - Called *attack dataset*
  - Image with target label

Reference image

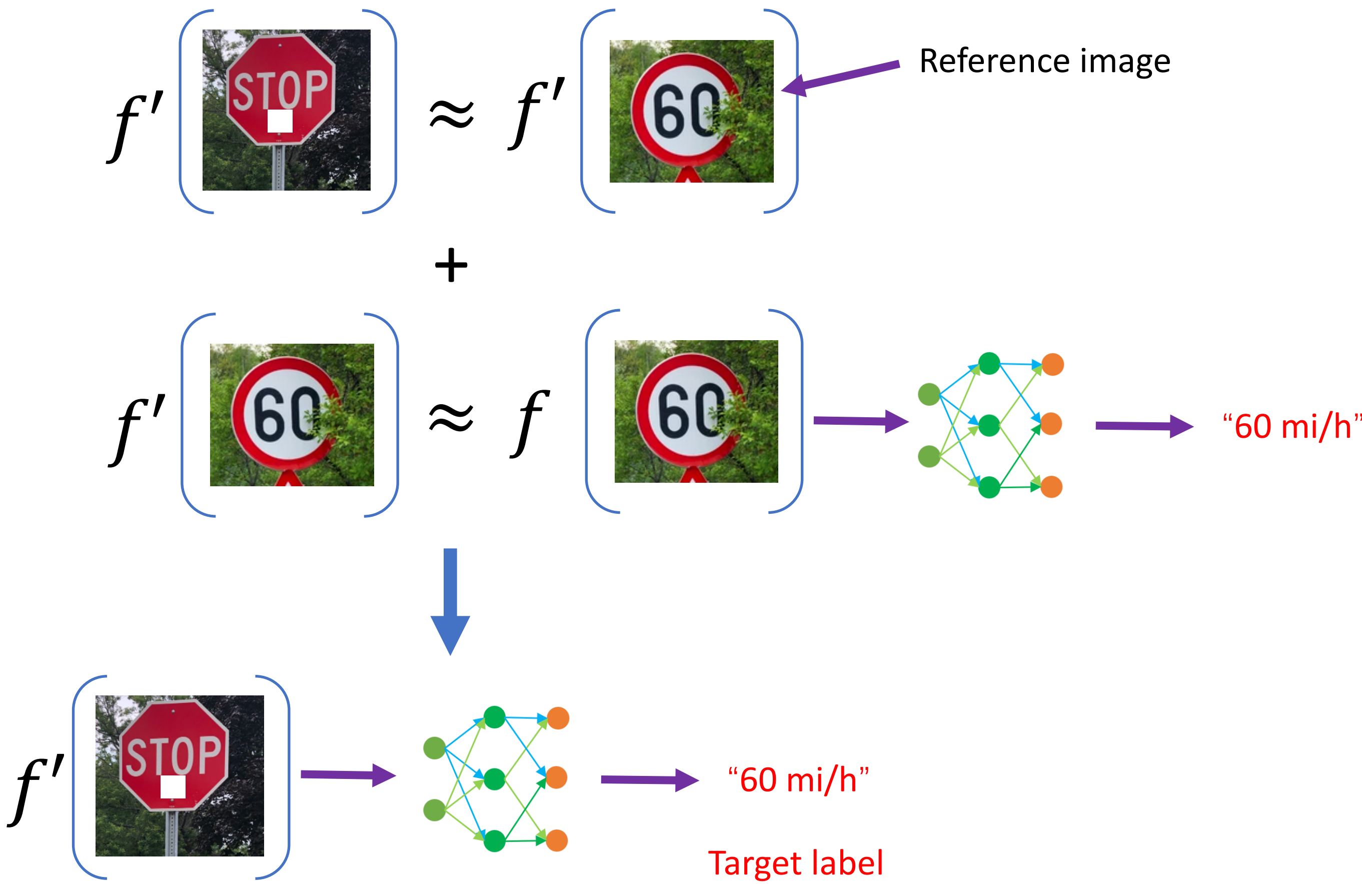
Key Idea of Our Attack

- Formulate as an optimization problem
  - Effectiveness loss
    - Quantify effectiveness goal
  - Utility loss
    - Quantify utility goal
  - Minimize a weighted sum of the two losses

Quantifying Effectiveness Goal

Condition I:

Condition II:



$f'(x)$: feature vector for x

Quantifying Condition I

Embedding backdoor trigger e to x

Cosine similarity

$$L_0 = -\frac{\sum_{x \in D_a} s(f'(x \oplus e), f'(x_r))}{|D_a|}$$

Reference image

Attack dataset

Backdoored encoder

Normalization

Quantifying Condition II

Backdoored encoder

Clean encoder

$$L_1 = -s(f'(x_r), f(x_r))$$

Cosine similarity

Reference image

Quantifying effectiveness goal: $L_0 + \lambda_1 \cdot L_1$

Hyperparameter

Quantifying Utility Goal

Classification of an image without backdoor trigger is unaffected

$$f'(x) \approx f(x)$$

$$L_2 = -\frac{1}{|D_a|} \cdot \sum_{x \in D_a} s(f'(x), f(x))$$

Cosine similarity

Attack dataset

Optimization Problem

Quantifying effectiveness goal

Quantifying utility goal

$$L = L_0 + \lambda_1 \cdot L_1 + \lambda_2 \cdot L_2$$

Two hyperparameters

$$f \xrightarrow{\text{Minimizing } L} f'$$

Experimental Setup

- Pre-training encoders
  - Pre-training algorithm
    - SimCLR
  - Pre-training dataset
    - CIRAR10
- Building downstream classifiers
  - Downstream tasks
    - GTSRB, SVHN, STL10
  - Downstream classifier
    - A fully connected neural network

Attack Setting

- Attack dataset
  - Pre-training dataset
- Target label
  - Different for different target downstream tasks
- Reference image
  - Collected from the Internet
- Hyperparameters

  $\lambda_1 = 1,\ \lambda_2 = 1$

Attack Success Rate

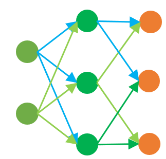
[0.1, 0.3, ⋯, 0.2]

"60 mi/h"

f'

[0.2, 0.1, ⋯, 0.3]

Downstream classifier



Built upon f'

"40 mi/h"

⋮ ⋮ ⋮ ⋮

[0.3, 0.0, ⋯, 0.1]

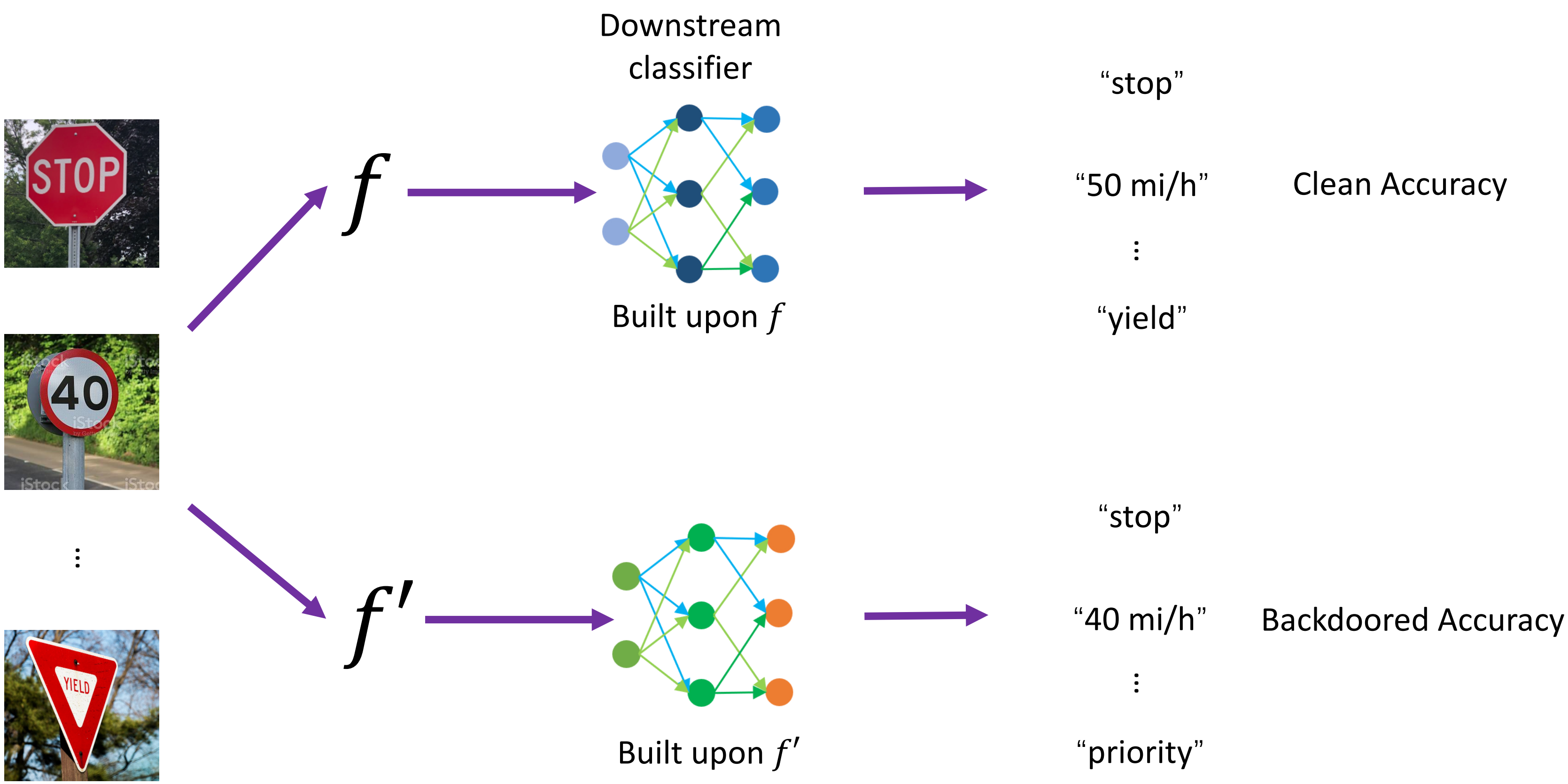
"60 mi/h"

Fraction of "60 mi/h"

BadEncoder Achieves Effectiveness Goal

Target Downstream Task	Attack Success Rate (%)
GTSRB	98.64
SVHN	99.14
STL10	99.73

Clean Accuracy and Backdoored Accuracy



BadEncoder Achieves Utility Goal

Target Downstream Task	Clean Accuracy (%)	Backdoored Accuracy (%)
GTSRB	81.84	82.27
SVHN	58.50	69.32
STL10	76.14	76.18

Evaluation on Real-world Pre-trained Encoders

- OpenAI's encoder CLIP
  - 400 million (image, text) pairs collected from the Internet
- Attack dataset
  - ImageNet dataset

Results for CLIP

BadEncoder achieves effectiveness goal

BadEncoder achieves utility goal

Target Downstream Task	Attack Success Rate (%)	Clean Accuracy (%)	Backdoored Accuracy (%)
GTSRB	99.33	82.36	82.14
STL10	99.81	97.09	96.69
SVHN	99.99	70.60	70.27

Existing Defenses are Insufficient

- Empirical defenses
  - Neural Cleanse [Oakland'19]
    - Cannot detect backdoored encoder
  - MNTD [Oakland'21]
    - Detection accuracy is close to random guessing

- Provable defense
  - PatchGuard [USENIX Security'21]
    - Insufficient provable robustness guarantees

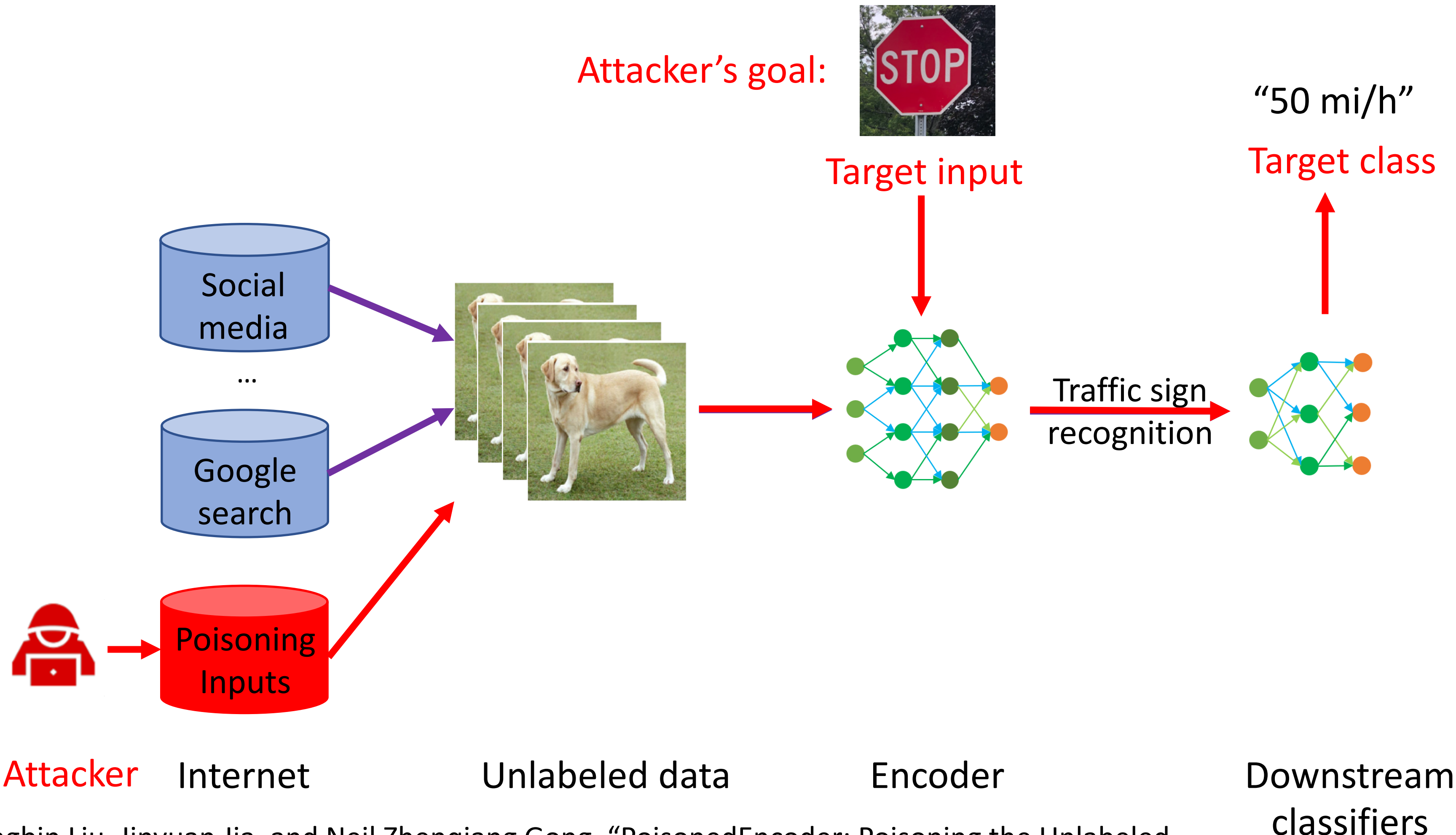
Summary

- Pre-trained encoders are vulnerable to backdoor attack
- Insecure encoders lead to a single point of failure of AI ecosystem
- Existing defenses are insufficient to defend against BadEncoder

Road Map

- Part I: Backdoor attack to pre-trained encoders
- **Part II: Data poisoning attack to pre-trained encoders**
- Part III: Data auditing for pre-trained encoders

Encoder is Vulnerable to Data Poisoning Attacks



Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. "PoisonedEncoder: Poisoning the Unlabeled Pre-training Data in Contrastive Learning". In *USENIX Security Symposium,* 2022.

Threat Model

- One target downstream task
  - E.g., traffic sign recognition
- One target input
  - E.g., an image of the stop sign
- One target class
  - E.g., “50 mi/h”

Target input

- 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 approximate 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 

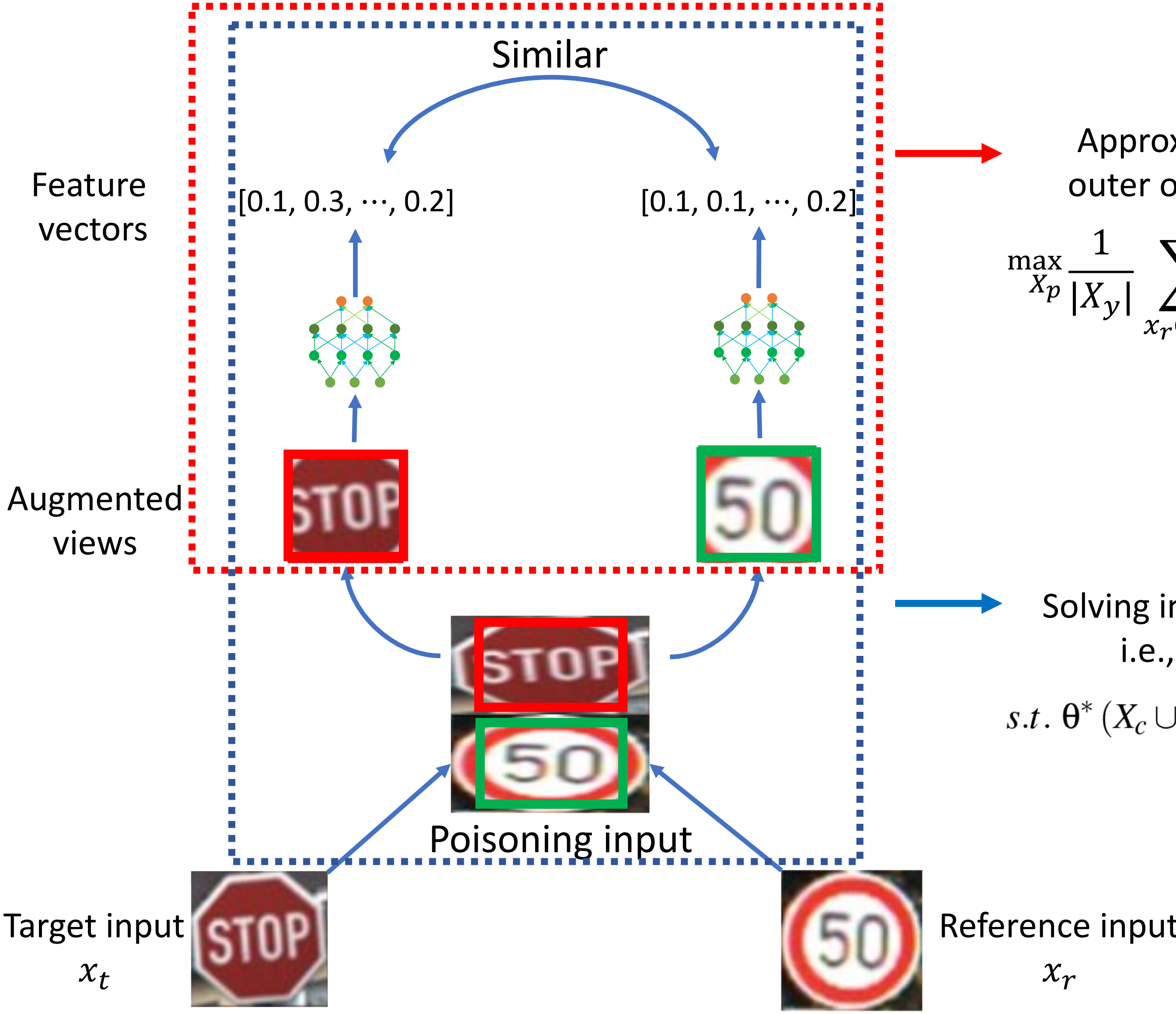
Pre-trained encoder

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

Inner optimization problem

Contrastive loss

Our PoisonedEncoder



Approximately solving the outer optimization problem:

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

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

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

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

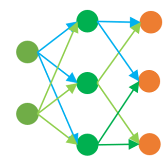
"60 mi/h"

[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

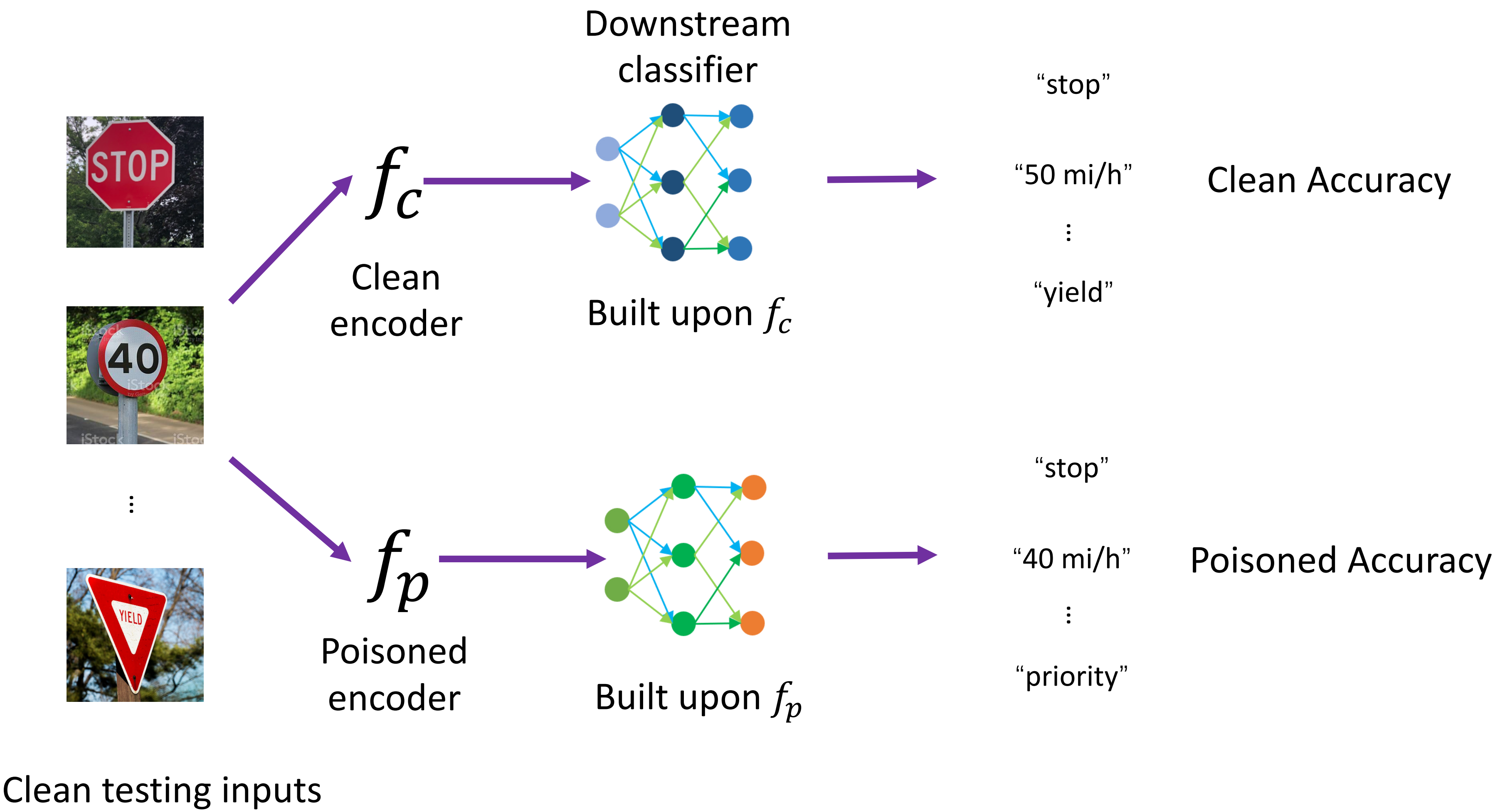
Target classes

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
  - Bagging [AAAI'21]
  - Pre-training encoder w/o random cropping
    - Effective but 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

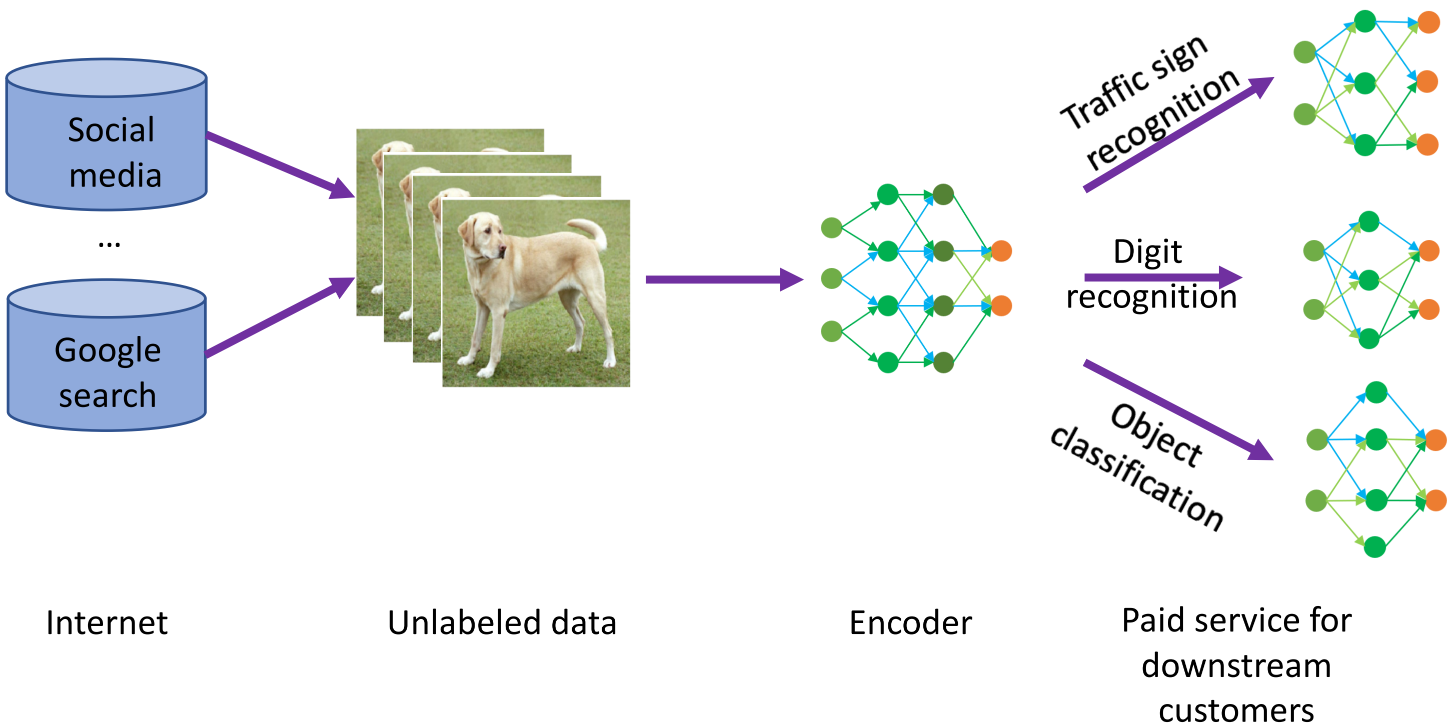
Summary

- Pre-trained encoders are vulnerable to data poisoning attacks
- Insecure encoders lead to a single point of failure of AI ecosystem
- Defenses are insufficient to defend against PoisonedEncoder

Road Map

- Part I: Backdoor attack to pre-trained encoders
- Part II: Data poisoning attack to pre-trained encoders
- **Part III: Data auditing for pre-trained encoders**

Motivation on Data Auditing



OpenAI's GPT API

Embedding models

Build advanced search, clustering, topic modeling, and classification functionality with our embeddings offering.

MODEL	USAGE
Ada	$0.0080 / 1K tokens
Babbage	$0.0120 / 1K tokens
Curie	$0.0600 / 1K tokens
Davinci	$0.6000 / 1K tokens

ChatGPT Plus: $20/month

Auditing Unauthorized Data Use

Was my public data used to pre-train a given encoder without authorization?

Examples of Real-world Unauthorized Data Use

BBC | Sign in | Home | News | Sport | Reel | Worklife | Travel

NEWS

Home | Coronavirus | Climate | Video | World | US & Canada | UK | Business | Tech | Science | Stories

Tech

Twitter demands AI company stops 'collecting faces'

23 January 2020

Twitter has demanded an AI company stop taking images from its website.

FTC settlement with Ever orders data and AIs deleted after facial recognition pivot

Natasha Lomas @riptari / 8:43 AM EST • January 12, 2021

Comment

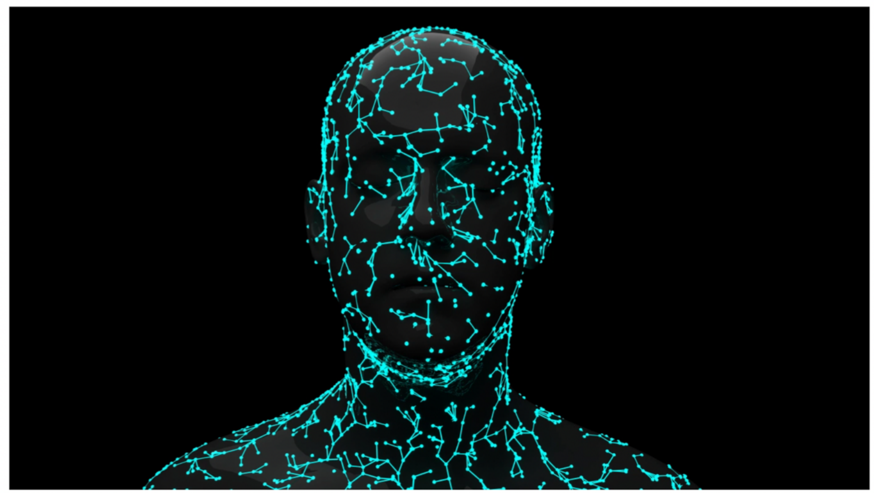
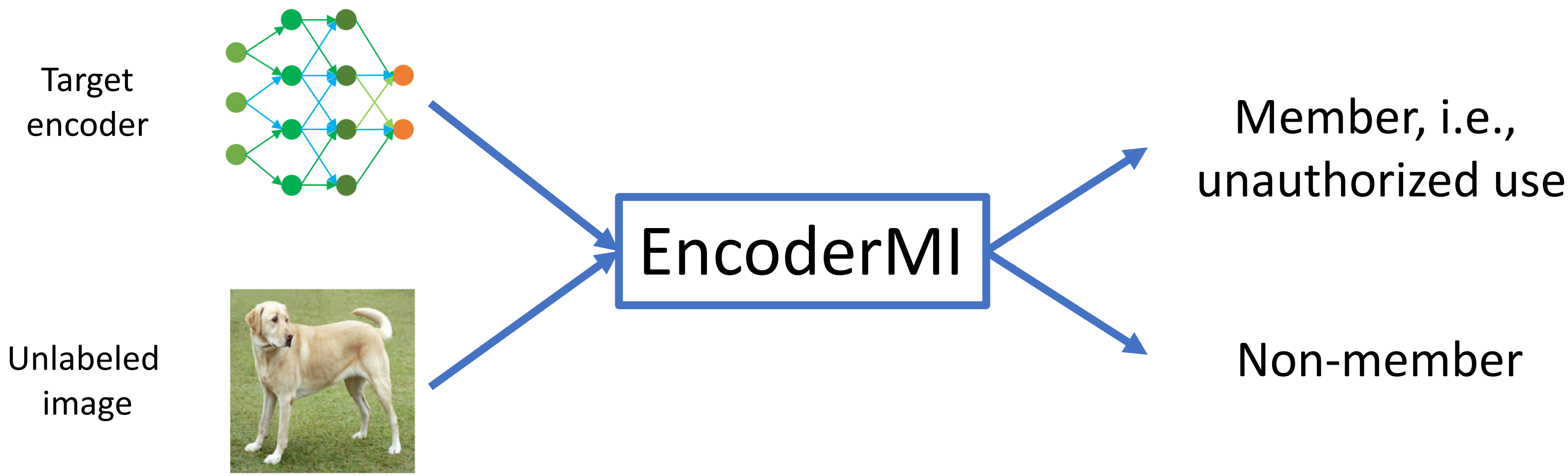


Image Credits: Design Cells / Getty Images

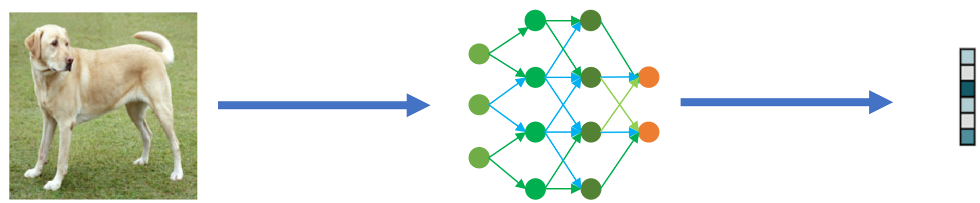
The maker of a defunct cloud photo storage app that pivoted to selling facial recognition services has been ordered to delete user data and any algorithms trained on it, under the terms of an FTC settlement.

Our EncoderMI: Membership Inference based Data Auditing for Pre-trained Encoders

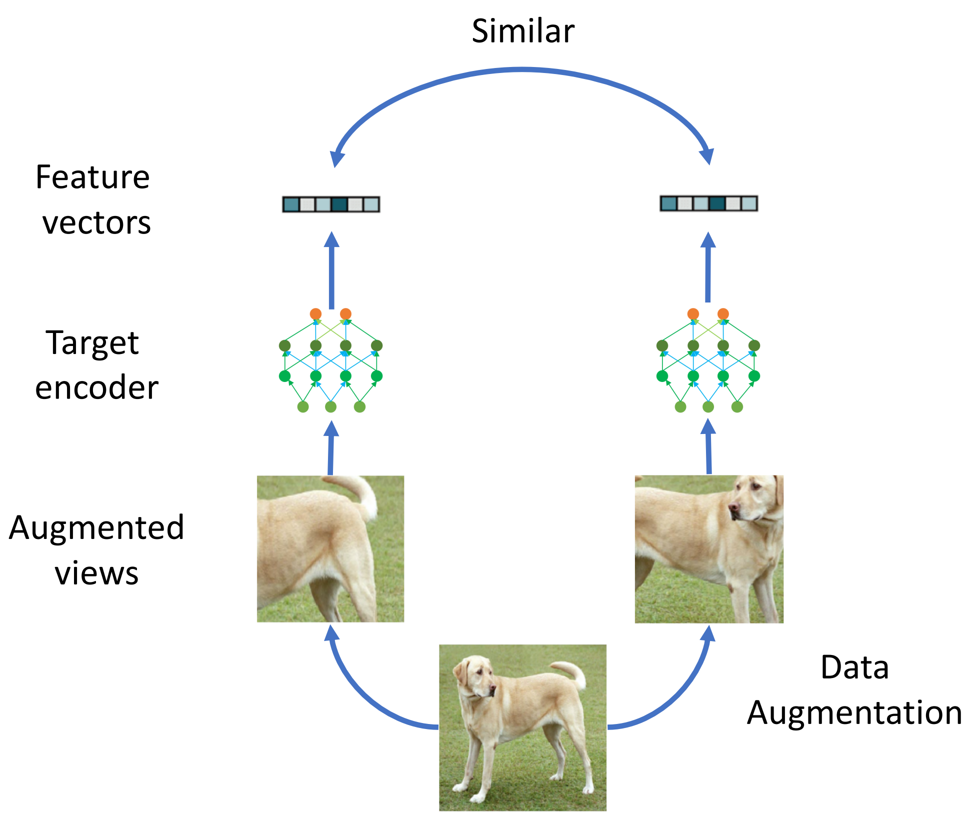


Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. “EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning”. In *ACM Conference on Computer and Communications Security (CCS),* 2021.

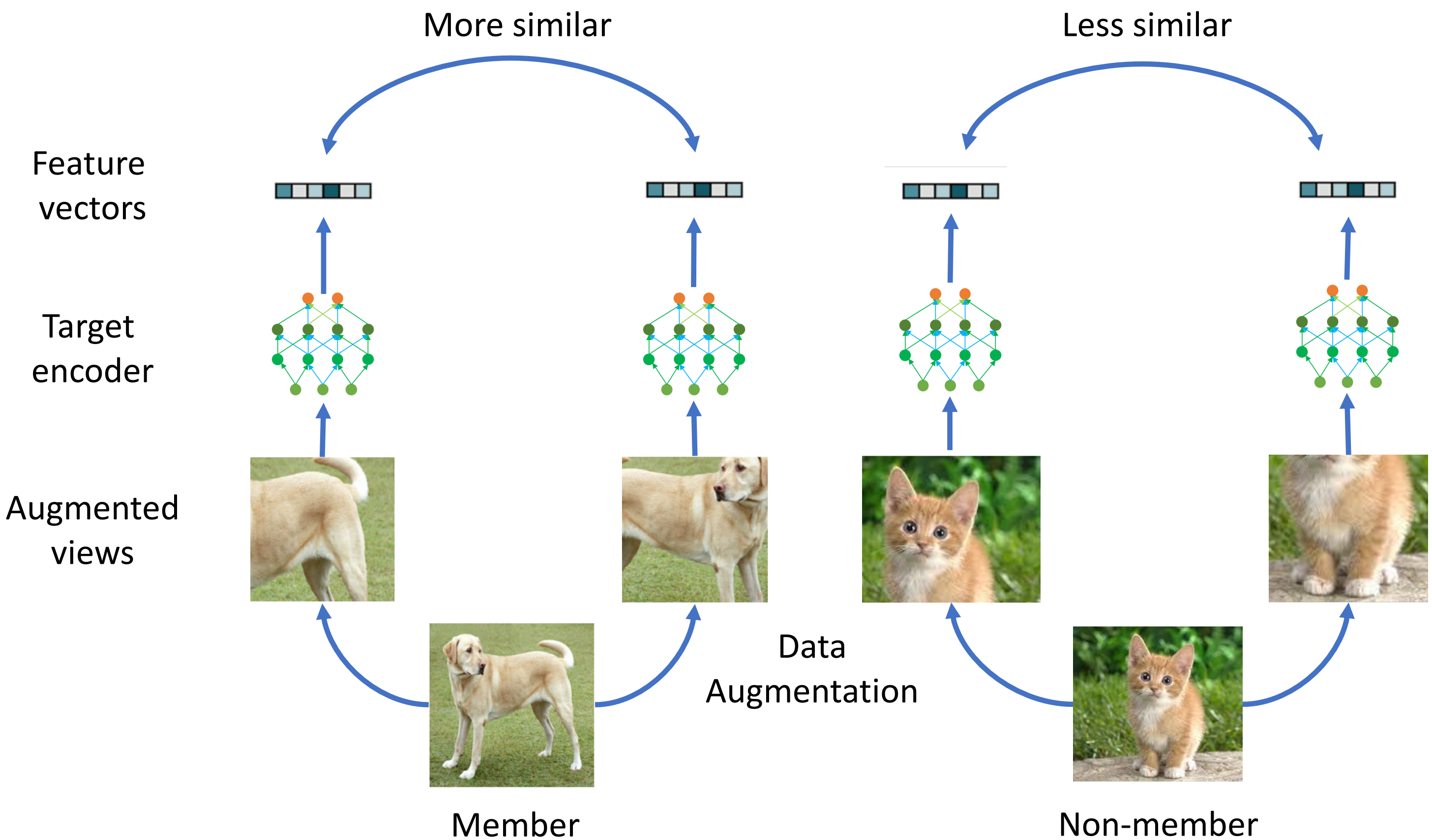
Threat Model: Black-box Access



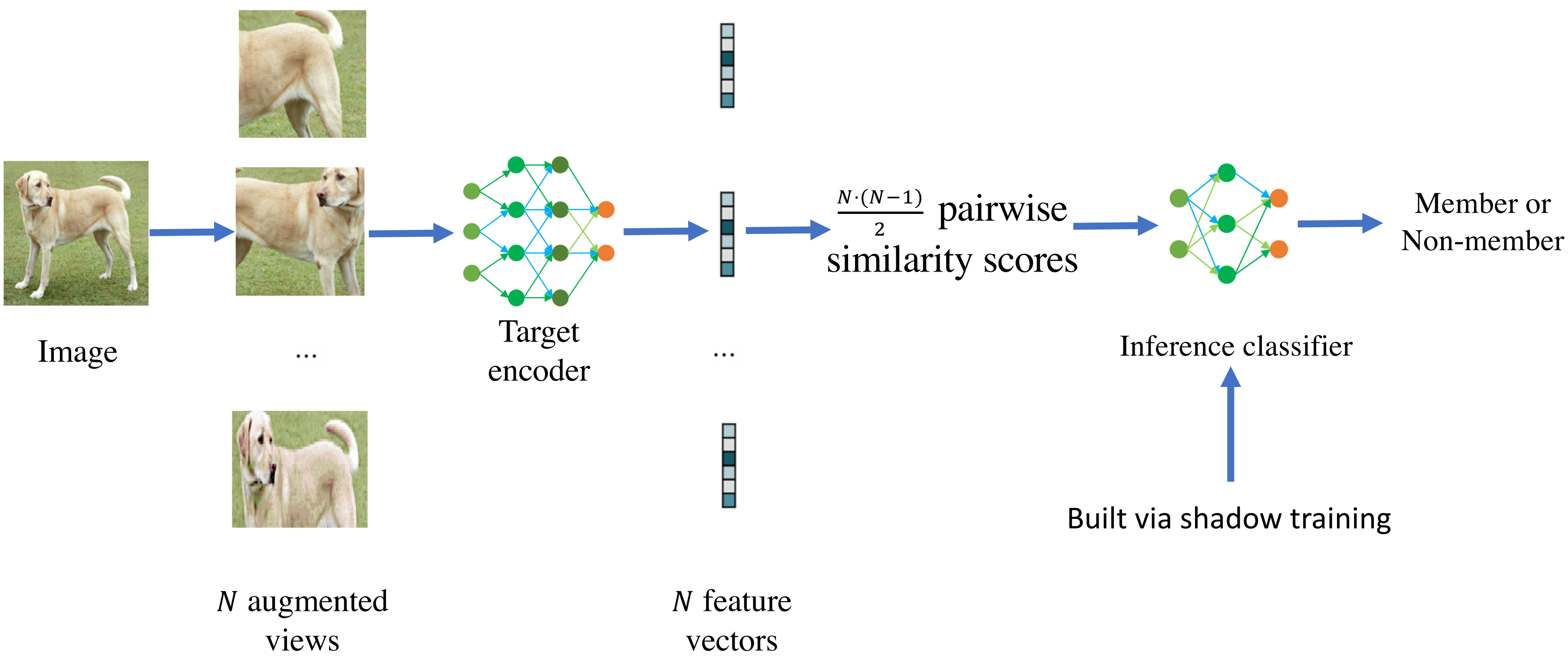
Revisiting Encoder Pre-training



Our Key Observation



Overview of Our EncoderMI



Shadow Training Setup

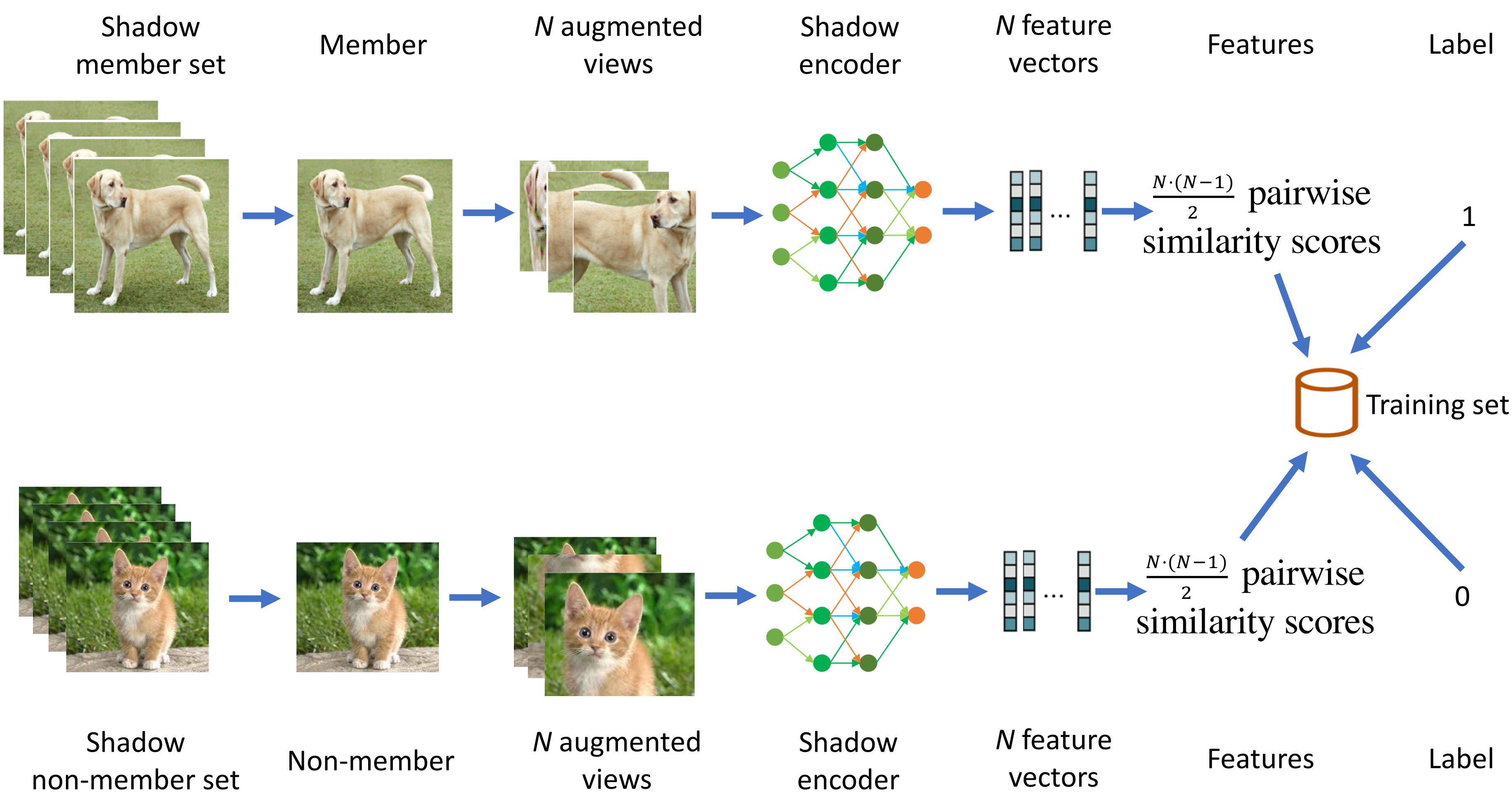
- Unlabeled images: *shadow dataset*
- Evenly divide into two halves
  - Shadow member set
  - Shadow non-member set

Pre-training a Shadow Encoder

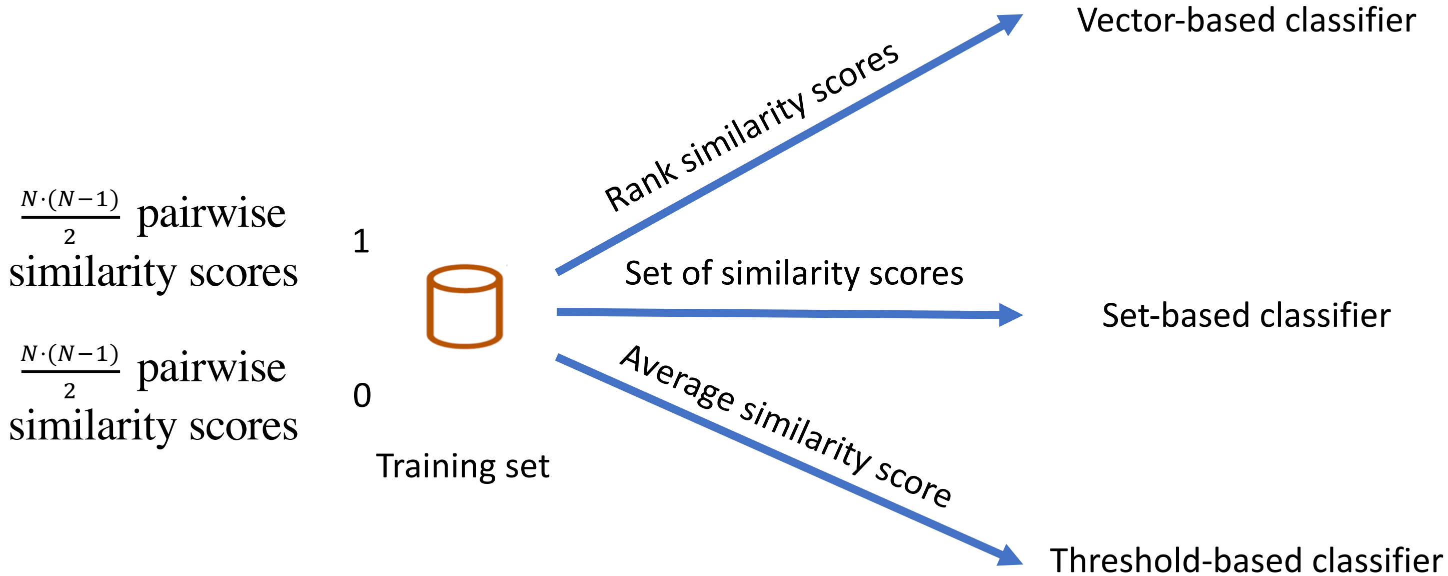
Shadow member set

Shadow encoder

Constructing a Training Set for Inference Classifier



Building an Inference Classifier



Experimental Setup

- Pre-training target encoder
  - Pre-training algorithm
    - MoCo
  - Pre-training dataset
    - CIFAR10
  - Target encoder architecture
    - ResNet18
- Pre-training shadow encoder
  - Pre-training algorithm
    - SimCLR
  - Pre-training dataset
    - STL10
  - Shadow encoder architecture
    - VGG11
- N=10

Evaluation Metrics

- 10,000 members of target encoder
- 10,000 non-members of target encoder

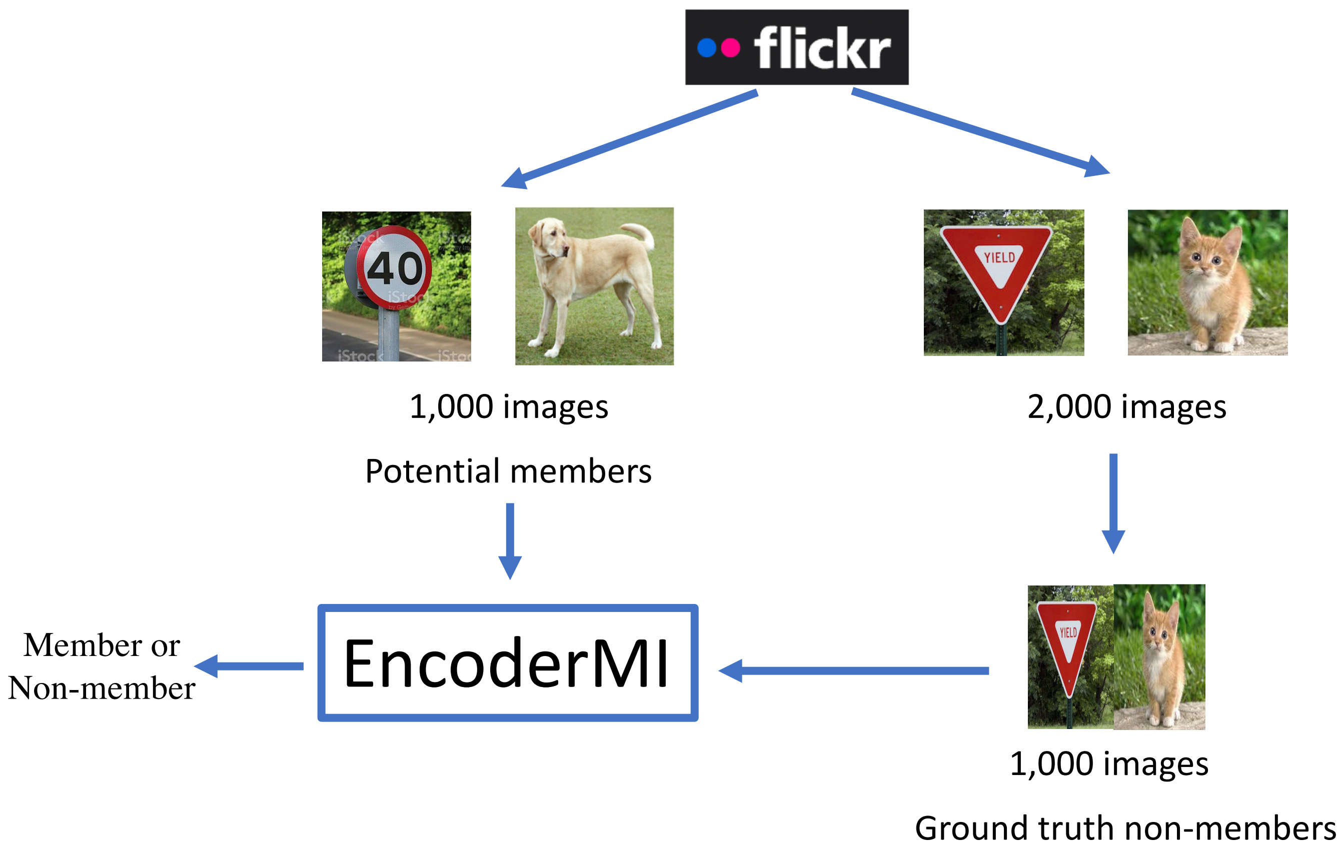
- Accuracy
  - Fraction of members/non-members whose memberships are inferred correctly

EncoderMI is Effective

Vector-based classifier	Set-based classifier	Threshold-based classifier
86.2%	78.1%	82.1%

Evaluation on CLIP

How to collect members and non-members of CLIP?



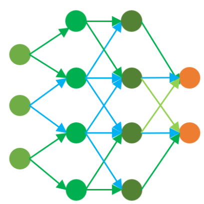
EncoderMI is Effective for CLIP

Vector-based classifier	Set-based classifier	Threshold-based classifier
73.5%	72.7%	74.5%

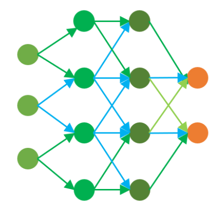
Summary

- Data auditing is an emerging problem for pre-trained encoders
- Feature similarity between augmented views can be used to audit unauthorized data use in pre-trained encoders

StolenEncoder



Target
encoder



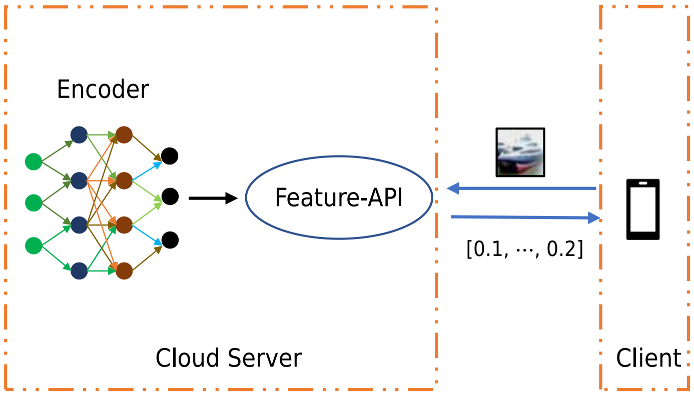
Stolen
encoder

Similar utility

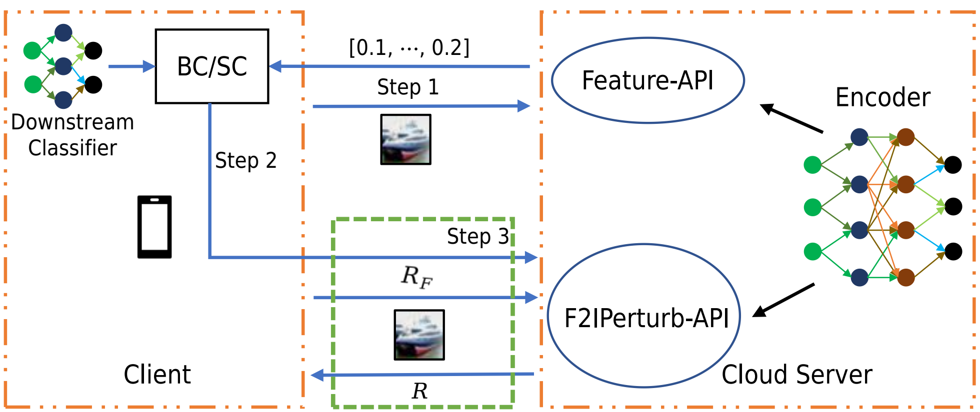
Less data & computation resource

Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. "StolenEncoder: Stealing Pre-trained Encoders in Self-supervised Learning". In *ACM CCS*, 2022.

Robust Encoder as a Service



Standard encoder as a service



Robust encoder as a service

Wenjie Qu, Jinyuan Jia, and Neil Zhenqiang Gong. "REaaS: Enabling Adversarially Robust Downstream Classifiers via Robust Encoder as a Service". In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2023.

Conclusion

- Part I: Backdoor attack to pre-trained encoders
  - "BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning". In *IEEE Symposium on Security and Privacy*, 2022.

- Part II: Data poisoning attack to pre-trained encoders
  - "PoisonedEncoder: Poisoning the Unlabeled Pre-training Data in Contrastive Learning". In *USENIX Security Symposium,* 2022.

- Part III: Data auditing for pre-trained encoders
  - "EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning". In *ACM CCS,* 2021.

Acknowledgements

Jinyuan Jia   Yupei Liu
Hongbin Liu   Wenjie Qu