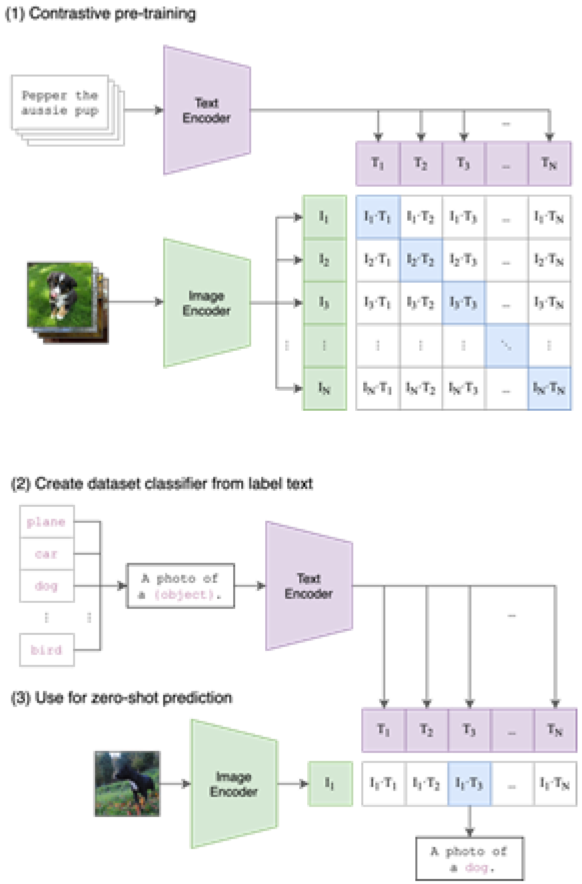
Backdoor attacks to foundation models

Talkers: Reachal Wang, Weili Wang, Haolou Sun, Zedian Shao

Background

- Backdoor Attacks:
  - Training phase compromise
  - Triggering the backdoor

- Foundation Models:
  - Pre-trained on vast amounts of diverse data and can be adapted or fine-tuned for a wide range of downstream tasks
  - Image encoders
  - CLIP



Two Attack Phases

- Poison the training dataset, i.e., data poisoning based backdoor attack
  - Poisoning and Backdoor Contrastive Learning
  - An Embarrassingly Simple Backdoor Attack on Self-supervised Learning
- Compromise the pre-training process, i.e, model poisoning based backdoor attack
  - BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning
  - Rickrolling the Artist: Injecting Backdoors into Text Encoders for Text-to-Image Synthesis

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

- Threat Model
- Attacker’s Goal:
  - Inject backdoor to image encoder such that the downstream classifier based on the encoder will predict the input with trigger as the target class (effectiveness)
  - Remain stealthy (utility)
- Attacker’s background knowledge:
  - Have access to the clean encoder
  - Have access to shadow dataset (whether it is pretrain dataset?)
  - Downstream remains integrity (client side)
- Attacker’s capability:
  - Fine-tuning the encoder with shadow dataset

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

- Self supervised learning for Image encoder: SimCLR

$$\ell_{i,j} = -\log\left(\frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}(k \neq i) \cdot \exp(sim(z_i, z_k)/\tau)}\right)$$

- Update the encoder to:
  - Maximize the similarity between positive pairs
  - Minimize the similarity between negative pairs

- z: latent vector

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

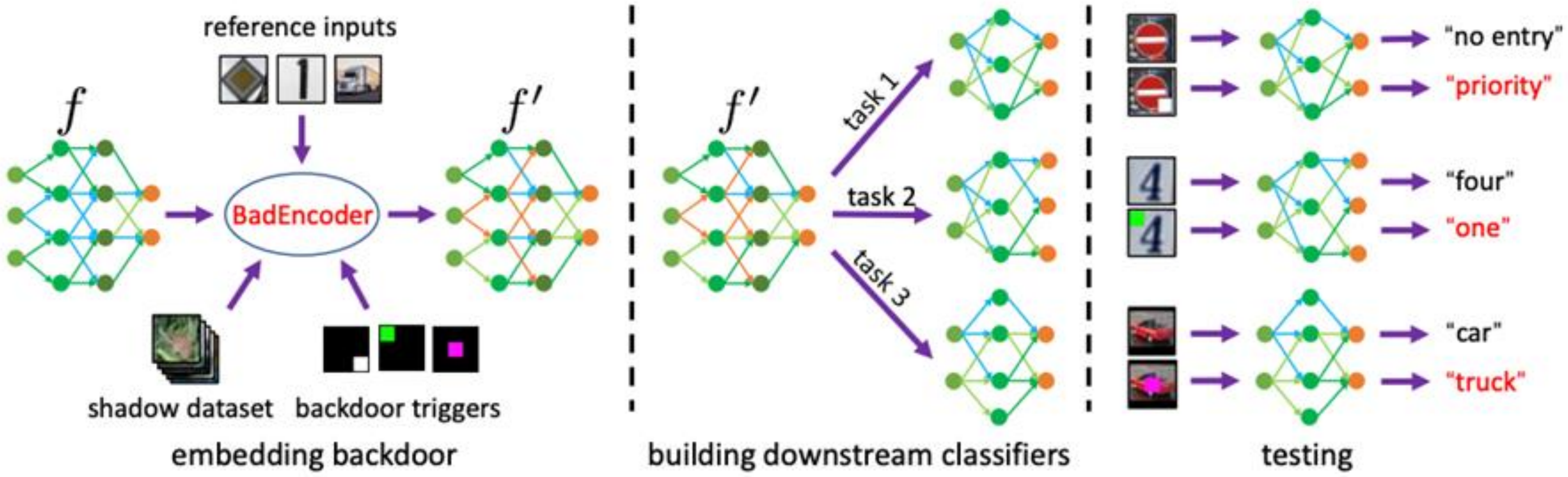


Fig. 1: Overview of BadEncoder.

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

- Effectiveness Goal

$$L_0 = -\frac{\sum_{i=1}^{t}\sum_{j=1}^{r_i}\sum_{\boldsymbol{x}\in\mathcal{D}_s} s(f'(\boldsymbol{x}\oplus\boldsymbol{e}_i), f'(\boldsymbol{x}_{ij}))}{|\mathcal{D}_s|\cdot\sum_{i=1}^{t} r_i},$$

$$L_1 = -\frac{\sum_{i=1}^{t}\sum_{j=1}^{r_i} s(f'(\boldsymbol{x}_{ij}), f(\boldsymbol{x}_{ij}))}{\sum_{i=1}^{t} r_i},$$

- Utility Goal

$$L_2 = -\frac{1}{|\mathcal{D}_s|}\cdot\sum_{\boldsymbol{x}\in\mathcal{D}_s} s(f'(\boldsymbol{x}), f(\boldsymbol{x})).$$

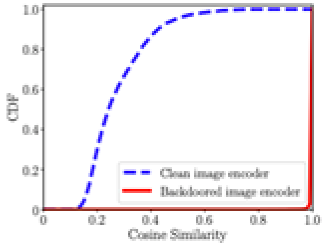


Fig. 4: The cumulative distribution functions (CDFs) of the cosine similarity scores between the feature vector of the reference input and those of the trigger-embedded inputs produced by the clean image encoder and backdoored image encoder.

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

- Effectiveness Goal

$$L_0 = -\frac{\sum_{i=1}^{t}\sum_{j=1}^{r_i}\sum_{\boldsymbol{x}\in\mathcal{D}_s} s(f'(\boldsymbol{x}\oplus\boldsymbol{e}_i), f'(\boldsymbol{x}_{ij}))}{|\mathcal{D}_s|\cdot\sum_{i=1}^{t} r_i},$$

$$L_1 = -\frac{\sum_{i=1}^{t}\sum_{j=1}^{r_i} s(f'(\boldsymbol{x}_{ij}), f(\boldsymbol{x}_{ij}))}{\sum_{i=1}^{t} r_i},$$

- Utility Goal

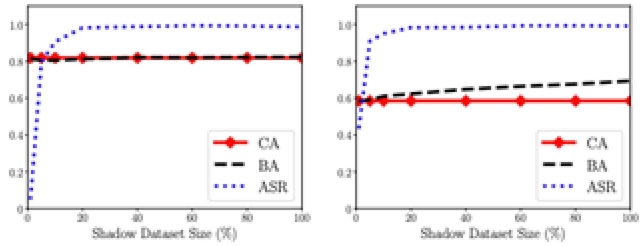
$$L_2 = -\frac{1}{|\mathcal{D}_s|}\cdot\sum_{\boldsymbol{x}\in\mathcal{D}_s} s(f'(\boldsymbol{x}), f(\boldsymbol{x})).$$

$$\min_{f'} L = L_0 + \lambda_1\cdot L_1 + \lambda_2\cdot L_2,$$

TABLE III: The impact of the loss terms.

Removed Loss Terms	CA(%)	BA(%)	ASR(%)
L_0	76.14	76.48	9.48
L_1		75.85	59.15
L_2		50.08	9.09
None		76.18	99.73

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning



(a) GTSRB (b) SVHN

Fig. 7: The impact of the shadow dataset size on our BadEncoder when the target downstream datasets are GTSRB (left) and SVHN (right). The shadow dataset is a subset of the pre-training dataset, which is CIFAR10.

TABLE IV: The impact of the shadow dataset's distribution on BadEncoder.

Target Downstream Dataset	Shadow Dataset	CA (%)	BA (%)	ASR (%)
GTSRB	A subset of pre-training dataset	81.84	81.21	98.19
	Same distribution		81.12	97.52
	Different distributions		82.21	93.27
SVHN	A subset of pre-training dataset	58.50	62.32	98.30
	Same distribution		62.07	98.06
	Different distributions		60.40	84.80
STL10	A subset of pre-training dataset	76.14	75.90	99.55
	Same distribution		75.70	99.43
	Different distributions		75.99	98.15

POISONING AND BACKDOORING CONTRASTIVE LEARNING

ICLR 2022

Authors: Nicholas Carlini, Andreas Terzis

Overview

- Adversary can mount powerful targeted poisoning and backdoor attacks against multimodal contrastive learning methods like CLIP, since it’s trained on noisy and uncurated training datasets (data without any human review).
- Poisoning adversary: malicious examples into the training dataset so that the model will misclassify a particular input as an adversarially-desired label.
- Patch-based backdoors: poisons a dataset so that the learned model will classify any input that contains a particular trigger-pattern as a desired target label.
- Fewer injections than clean label: 1% → 0.01% backdoor, 0.0001% poisoning

CONTRASTIVE LEARNING

- Constructs an embedding function f : X →E that maps objects of one type (e.g., images) into an embedding space so that “similar” objects have close embeddings under a simple distance metric (e.g., Euclidean distance or cosine similarity).
- Multimodal contrastive learning: multiple domains simultaneously (e.g., images and text)

$$\mathcal{X} \subset \mathcal{A} \times \mathcal{B} \qquad\qquad f : \mathcal{A} \to E \text{ and } g : \mathcal{B} \to E$$

- Maximize the inner product of 〈f(a), g(b)〉, minimize inner product from (a’, b’)

Contrastively trained models

- **Feature extractors** for second downstream classifier.

f to map some new training dataset $\hat{X}$ into the embedding space E, and then train a linear classifier $z : E \to \mathcal{Y}$ to map the embeddings to predictions of the downstream task.

- **Zero-shot classifiers**

As **zero-shot classifiers**. Given an object description (e.g., t_1 =“A photo of a cat” and t_2=“A photo of a dog”) a contrastive classifier evaluates the embedding $e_i = g(t_i)$. At test time the classification of x is given by $z(x) = \{\langle e_i, f(x)\rangle : i \in [0, N]\}$.

Threat Model

- Attacker’s Goal
  - Cause the contrastive model to behave incorrectly in one of the two cases
  - Specifically attacking the image embedding function **f**
- Attacker’s Capability
  - The adversary can inject a small number of examples into the training dataset
  - To be more realistic, adversaries who can poison 100−10,000×fewer images
  - When poisoned model is the feature extractor, adversary does not have access to the fine tuning task training dataset or algorithm (No control over downstream use case after the model has been poisoned or backdoored)

Attack

- Simpler case

$$y' = z(f_\theta(x'))$$
$$f_\theta \leftarrow \mathcal{T}(\mathcal{X} \cup \mathcal{P})$$

- Multi-sample Poisoning

$$\mathcal{P} = \{(x', c) \ : \ c \in \text{caption set}\}$$

- Backdoor models

$$\mathcal{P} = \{(x_i \oplus bd, c) \ : \ c \in \text{caption set}, \ x_i \in \mathcal{X}_{\text{subset}}\}$$

Poisoning Evaluation

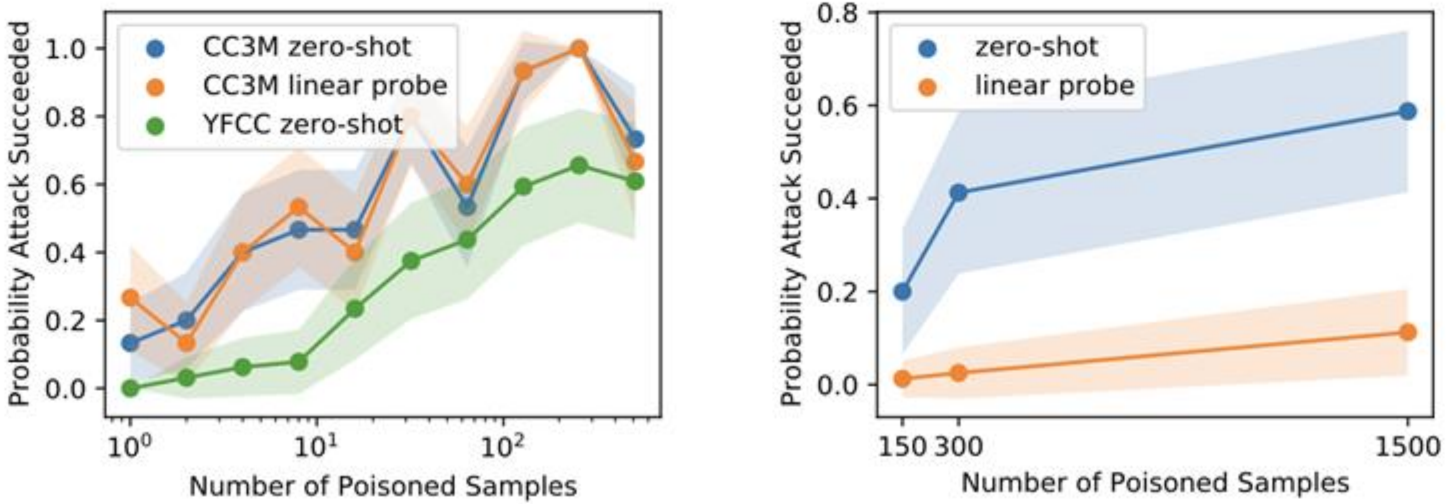


Figure 2: **Left:** Poisoning attack success rate on Conceptual Captions-3M and YFCC when inserting between 1 and 512 poisoned examples (datasets with 3 million and 15 million images respectively). **Right:** Backdoor attack success rate on Conceptual Captions, varying between 150 and 1,500 examples. The shaded region corresponds to one standard deviation of variance.

Poisoning Evaluation

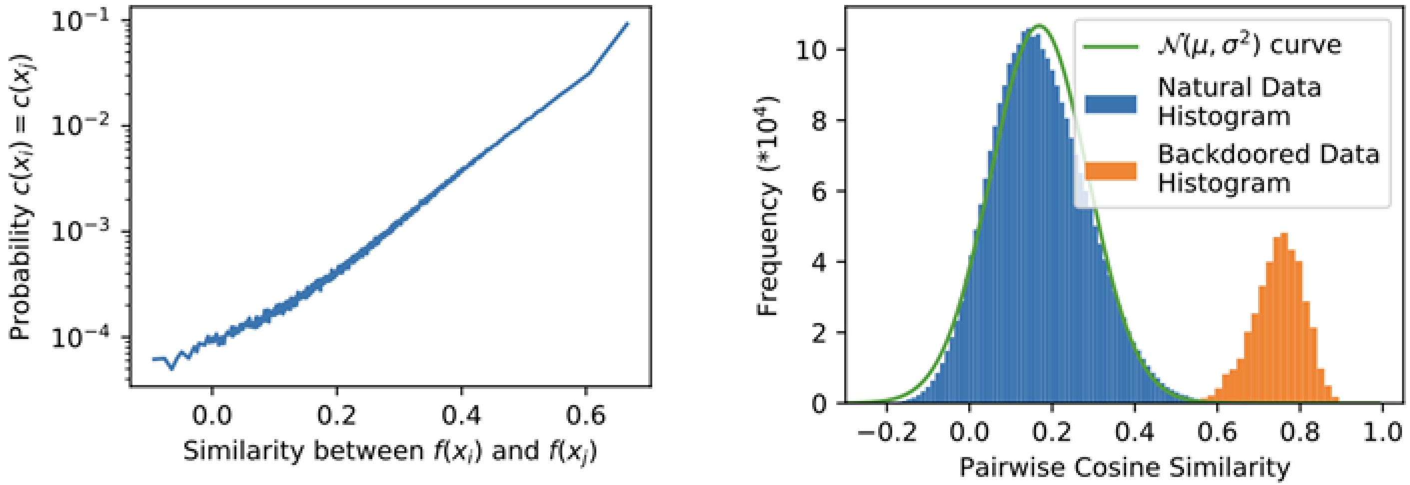


Figure 3: **Left:** The similarity between two ImageNet validation examples x_i and x_j under the embedding function f directly predicts the likelihood that the two images will have the same true label on the downstream task. **Right:** By poisoning 0.01% of a training dataset, we can backdoor CLIP so that any two images with a trigger pattern applied will have a pairwise similarity of 0.78. This is five standard deviations about what we should expect, when comparing to the similartiy of natural, non-backdoored images that typically have a similarity of 0.1.

Backdoor Evaluation

Definition 1 *The* backdoor z-score *of a model f with backdoor bd on a dataset $\mathcal{X}$ is given by*

$$\left(\underset{u\in\mathcal{X},v\in\mathcal{X}}{Mean} [\langle f(u \oplus bd), f(v \oplus bd)\rangle] - \underset{u\in\mathcal{X},v\in\mathcal{X}}{Mean} [\langle f(u), f(v)\rangle] \right) \cdot \left(\underset{u\in\mathcal{X},v\in\mathcal{X}}{Var} [\langle f(u), f(v)\rangle] \right)^{-1}.$$

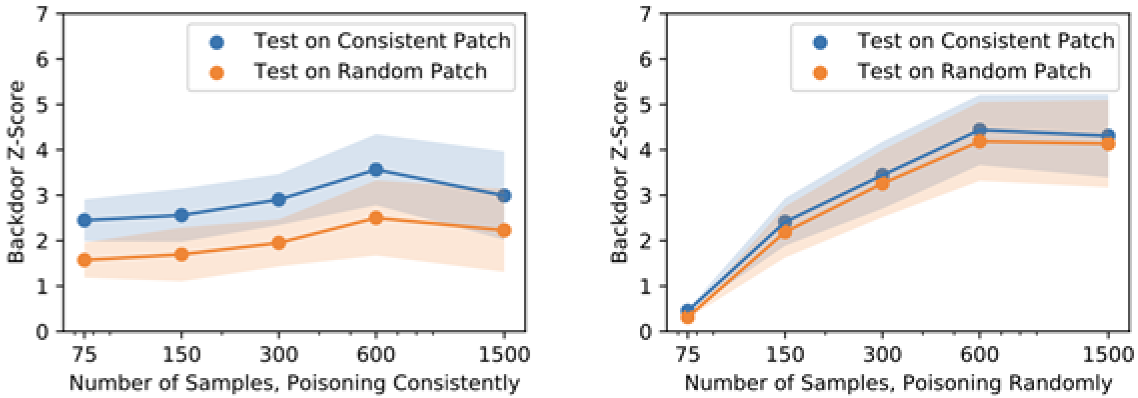


Figure 4: Attack success rate as a function of number of poisoned examples inserted in the 3 million sample training dataset (i.e., ranging from 0.0025% to 0.05%). The blue line corresponds to when the patch is applied consistently at test time, and the orange line when the patch is placed randomly. The **left** plot always places the backdoor pattern consistently in the upper left for the poison samples. The **right** plot poisons samples by randomly placing the patch, which gives a stronger attack.

An Embarrassingly Simple Backdoor Attack on Self-supervised Learning

Authors: Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, Ting Wang

Conference: ICCV 2023

Why Another Backdoor Attack?

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

Jinyuan Jia* Yupei Liu* Neil Zhenqiang Gong
Duke University
{jinyuan.jia, yupei.liu, neil.gong}@duke.edu

Backdoor Attacks on Self-Supervised Learning

Aniruddha Saha[1], Ajinkya Tejankar[2], Soroush Abbasi Koohpayegani[1], Hamed Pirsiavash[2]
[1] University of Maryland, Baltimore County [2] University of California, Davis
anisaha1@umbc.edu, atejankar@ucdavis.edu, soroush@umbc.edu, hpirsiav@ucdavis.edu

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

Hongbin Liu Jinyuan Jia Neil Zhenqiang Gong
Duke University
{hongbin.liu, jinyuan.jia, neil.gong}@duke.edu

POISONING AND BACKDOORING CONTRASTIVE LEARNING

Nicholas Carlini
Google

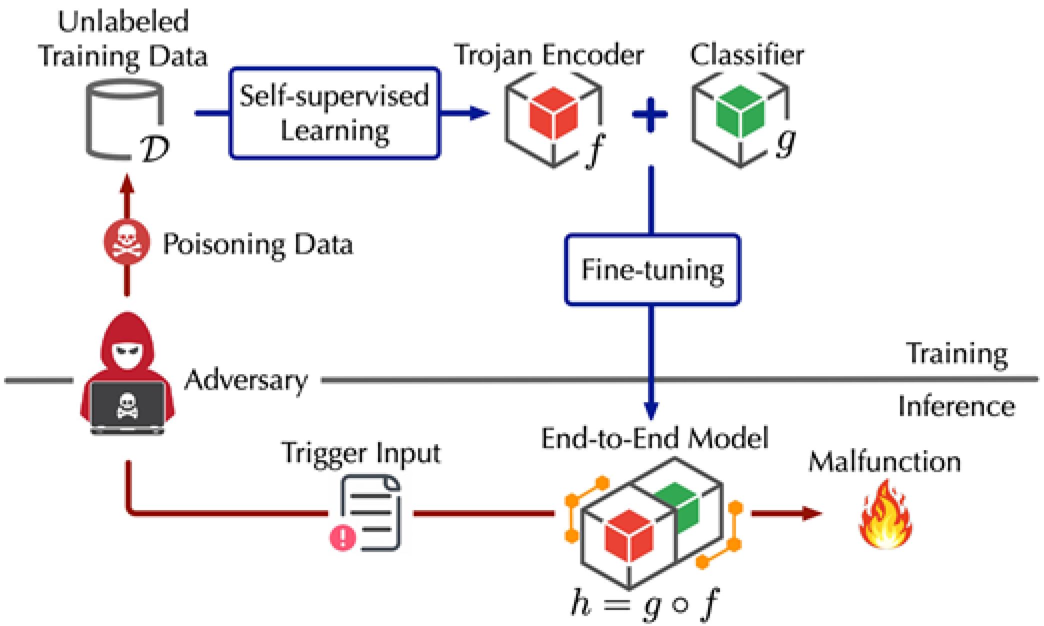
Andreas Terzis
Google

Motivation

- Model poisoning approach
  - Less practical: Need to compromise pre-training
- Data poisoning approach
  - Less effective: Poisoning data can be easily recognized

Clean Input | PoisonedEncoder | SSLBackdoor

Threat Model



Goals

- Effectiveness
- Evasiveness (BadEncoder utility goal)

Capabilities

- Pollute a tiny fraction of training data
- No knowledge of encoder/classifier

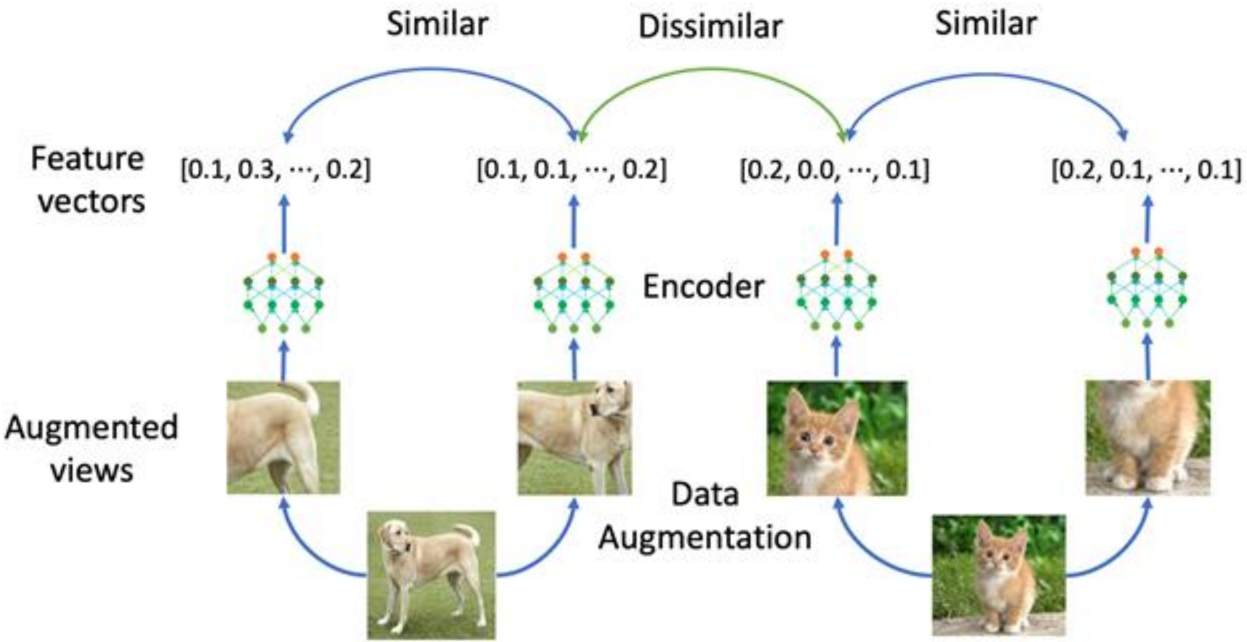
Recall: Paper Title

“An **Embarrassingly Simple** Backdoor Attack on Self-supervised Learning”

Can you figure out a simple attack approach?

Hint: No need to solve optimization problems

Contrastive Learning Revisited

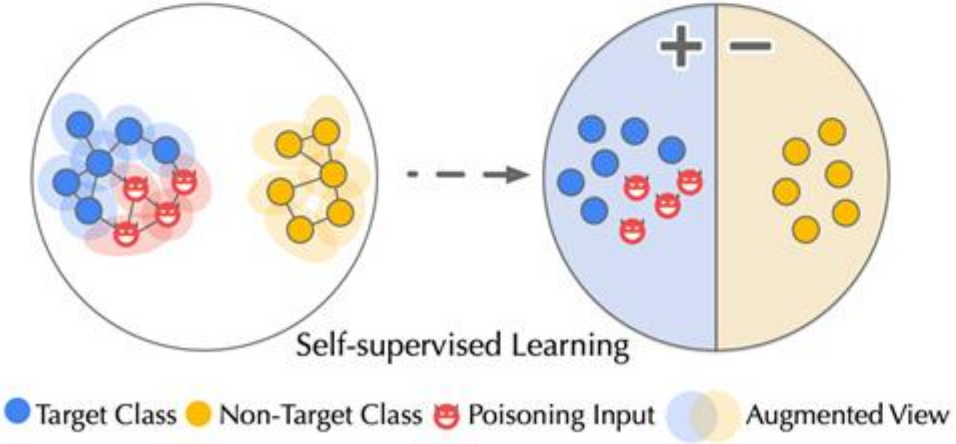


Design a trigger pattern that survives various augmentations!

Contrastive learning aligns the features of the same input under varying augmentations ("positive pair") while separating the features of different inputs ("negative pair")

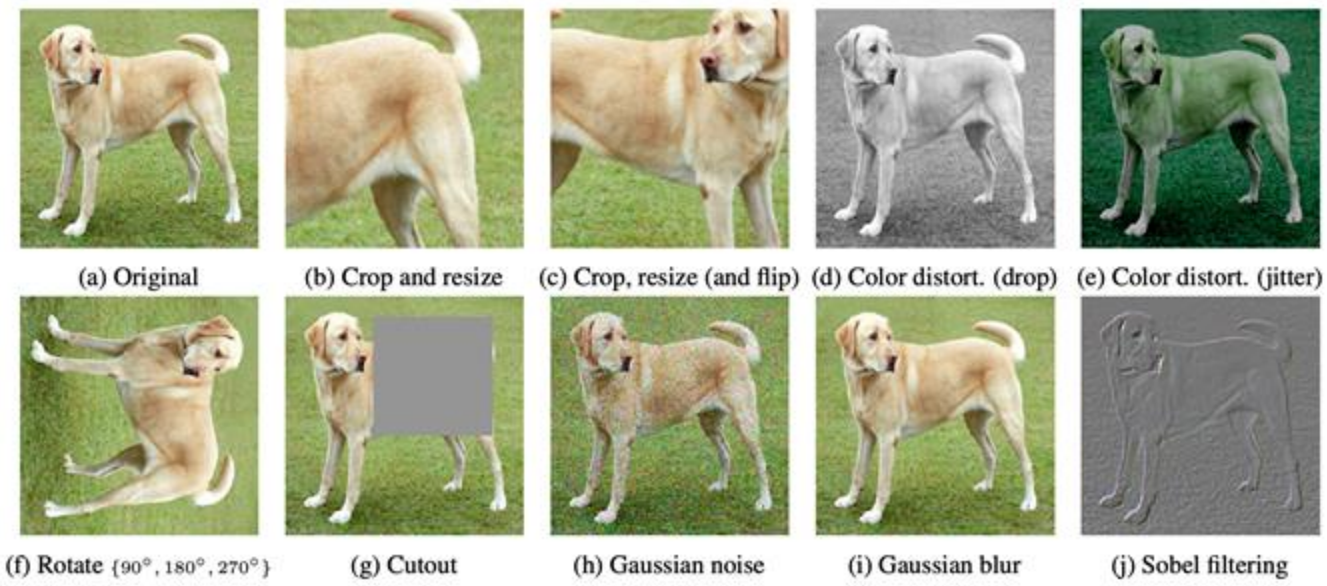
Attack Overview

- Trigger definition
  - Define the trigger as an augmentation-resistant perturbation
- Poisoning data generation
  - Add the trigger to inputs from the ta



- Training
  - Contrastive learning entangles trigger inputs with target-class inputs in the feature space, leading to their similar classification in downstream tasks

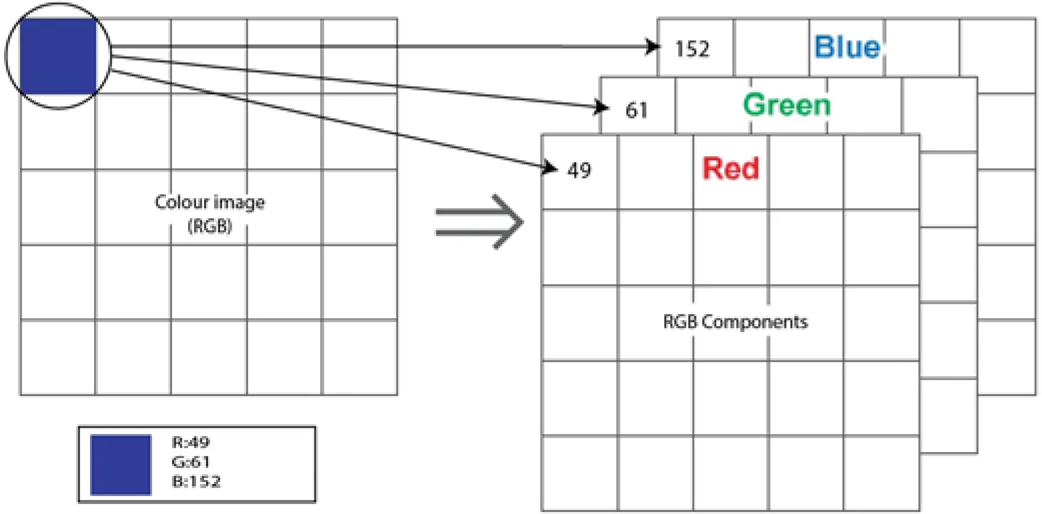
Augmentation-Resistant Perturbation Design



How to survive these data augmentation operators?

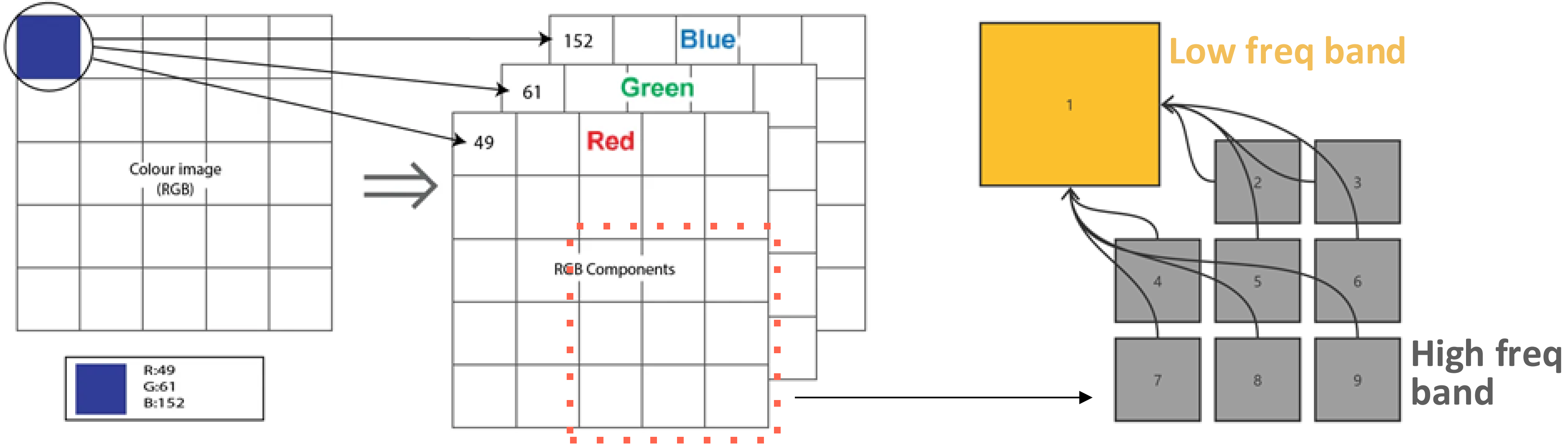
Hint: Consider different image domains

Spatial Domain and Frequency Domain



The spatial domain refers to the representation of an image in terms of its pixel values

Spatial Domain and Frequency Domain



The spatial domain refers to the representation of an image in terms of its pixel values

Transform a pixel block to a coefficient matrix. The matrix records the frequency info

Key Insight behind the Attack: Spectral Trigger

- The perturbations on the input’s mid/high-frequency bands lead to visually invisible patterns
- Common data augmentation operators (crop, resize…) only manipulate the spatial domain of images
- Spectral trigger tampers with frequency bands and has a global effect

Poisoning Data Generation

YC_bC_r color space separates the luminance component (Y) from the chrominance ones (C_bC_r)

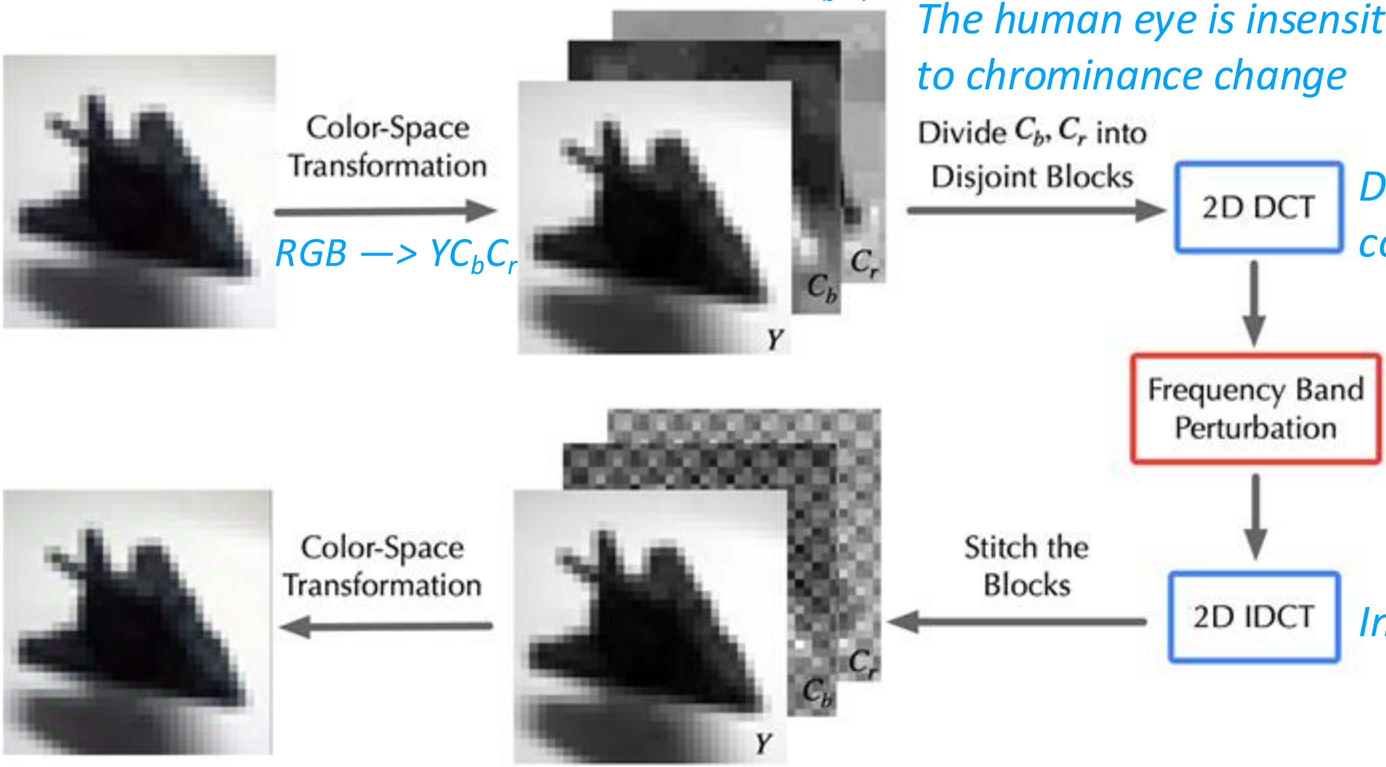
The human eye is insensitive to chrominance change

RGB —> YC_bC_r

DCT: discrete cosine transform

Magnify values at certain freq bands

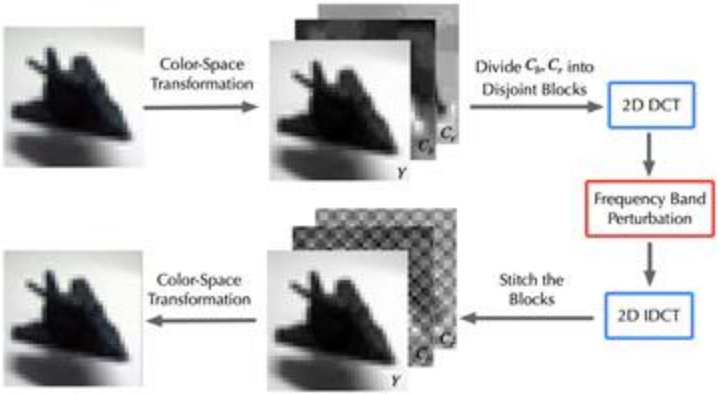
Inverse DCT



```python
def Poison_Images(self, x_train,  magnitude):
    x_train  = x_train * 255.
    x_train = self.RGB2YCbCr(x_train)
    x_train = self.DCT(x_train)  # (idx, ch, w, h)
    block_size = 32 # divide channels into 32x32 pixel blocks

    for ch in self.channel_list: # "channel_list": [1, 2] (Cb, Cr)
        for w in range(0, x_train.shape[2], block_size):
            for h in range(0, x_train.shape[3], block_size):
                # frequency bands 31 and 15 => "pos_list":[(31, 31), (15, 15)]
                for pos in self.pos_list:
                    p_val = x_train[:, ch, w+pos[0], h+pos[1]] + magnitude
                    x_train[:, ch, w+pos[0], h+pos[1]] = p_val

    x_train = self.IDCT(x_train)  # (idx, w, h, ch)
    x_train = self.YCbCr2RGB(x_train)
    x_train /= 255.
    x_train = torch.clamp(x_train, min=0.0, max=1.0)
    return x_train
```



https://github.com/meet-cjli/CTRL/blob/master/utils/frequency.py

Poisoning Data Comparison

Clean Input PoisonedEncoder SSLBackdoor Proposed

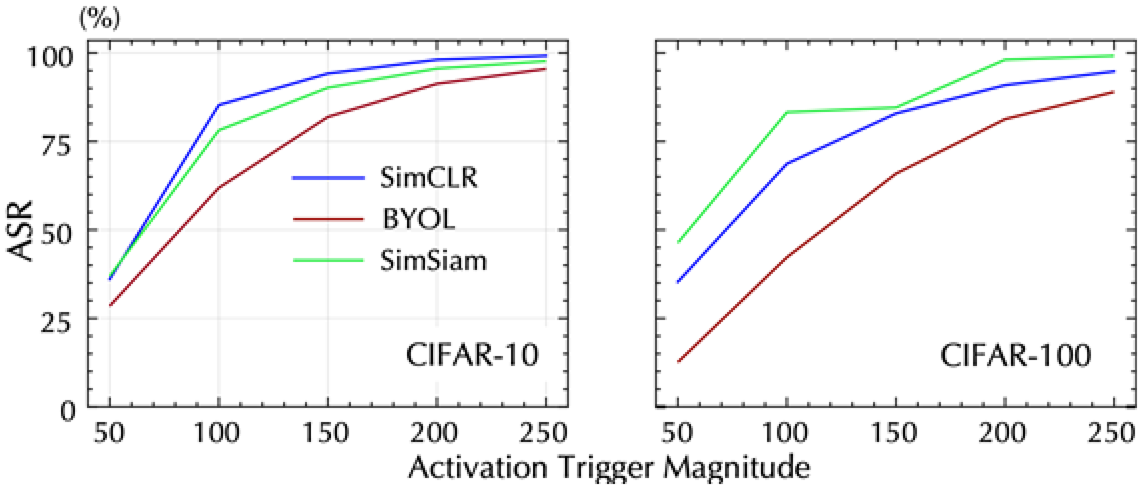
Compared with other attacks, the proposed attack can generate poisoning samples that are highly indistinguishable from clean data

Evaluation Setup

- Configs
  - Contrastive learning methods: SimCLR, BYOL, and SimSiam
    - ResNet-18 as the encoder
    - Data augmentations include RandomResizeCrop, RandomHorizontalFlip…
  - Datasets: CIFAR-10, CIFAR-100, ImageNet-100
    - Training set for encoder training using contrastive learning
    - Randomly sample 50 examples from each class of the corresponding testing set to train the downstream classifier (two-layer MLP)
- Metrics
  - Clean data accuracy (ACC): The accuracy of the model in classifying clean inputs
  - Attack success rate (ASR): The accuracy of the model in classifying trigger inputs as the adversary's designated class

Influence of Activation Trigger Magnitude

- When poisoning data, magnitude is fixed as 50 to ensure best image quality
- At inference phase, the attacker can use larger magnitude to improve success rate



- ➢On CIFAR-10 with SimCLR, ASR increases from 36% to 99%
- ➢Set trigger magnitude as 100: Good performance while not introducing much distortion

Attack Effectiveness

For ASR, we insert trigger on the full testing set and measure the ratio of trigger inputs that are classified to the target class.

Dataset	SimCLR		BYOL		SimSiam	
	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	79.1%	9.93%	82.4%	12.2%	81.5%	11.75%
CIFAR-100	48.1%	1.14%	51.0%	0.46%	52.0%	0.72%
ImageNet-100	42.2%	1.59%	45.1%	1.41%	41.3%	1.53%

Normal encoder training

Dataset	SimCLR		BYOL		SimSiam	
	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	80.5%	85.3%	82.2%	61.9%	82.0%	74.9%
CIFAR-100	47.6%	68.8%	50.8%	42.3%	52.6%	83.9%
ImageNet-100	42.2%	20.4%	45.9%	37.9%	40.2%	39.2%

Poisoning encoder training

- Evasiveness/utility goal: The backdoor model presents equivalent accuracy across different contrastive learning methods
- Effectiveness goal: Generally the backdoor attack presents success rate higher than or close to ACC

Summary

Strengths

- Neat attack scheme
- High attack effectiveness
- Stealthy trigger design

Weaknesses

- Frequency band selection
- DCT block size selection
- Theoretical analysis

Rickrolling the Artist: Injecting Backdoors into Text Encoders for Text-to-Image Synthesis

Authors: Lukas Struppek, Dominik Hintersdorf, Kristian Kersting

Talker: Reachal Wang

Why this Paper?

- A key question: How do foundation models impact AI system security regarding backdoor attacks?

BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning

Jinyuan Jia* Yupei Liu* Neil Zhenqiang Gong
Duke University
{jinyuan.jia, yupei.liu, neil.gong}@duke.edu

⟹ A backdoored image encoder compromises downstream image processing tasks

Rickrolling the Artist: Injecting Backdoors into Text Encoders for Text-to-Image Synthesis

Lukas Struppek [1] Dominik Hintersdorf [1] Kristian Kersting [1,2,3,4]
[1]Technical University of Darmstadt [2]Centre for Cognitive Science
[3]Hessian Center for AI (hessian.AI) [4]German Research Center for Artificial Intelligence (DFKI)
{struppek, hintersdorf, kersting}@cs.tu-darmstadt.de

⟹ A backdoored text encoder damages all AI systems built around it, e.g., text-to-image synthesis models

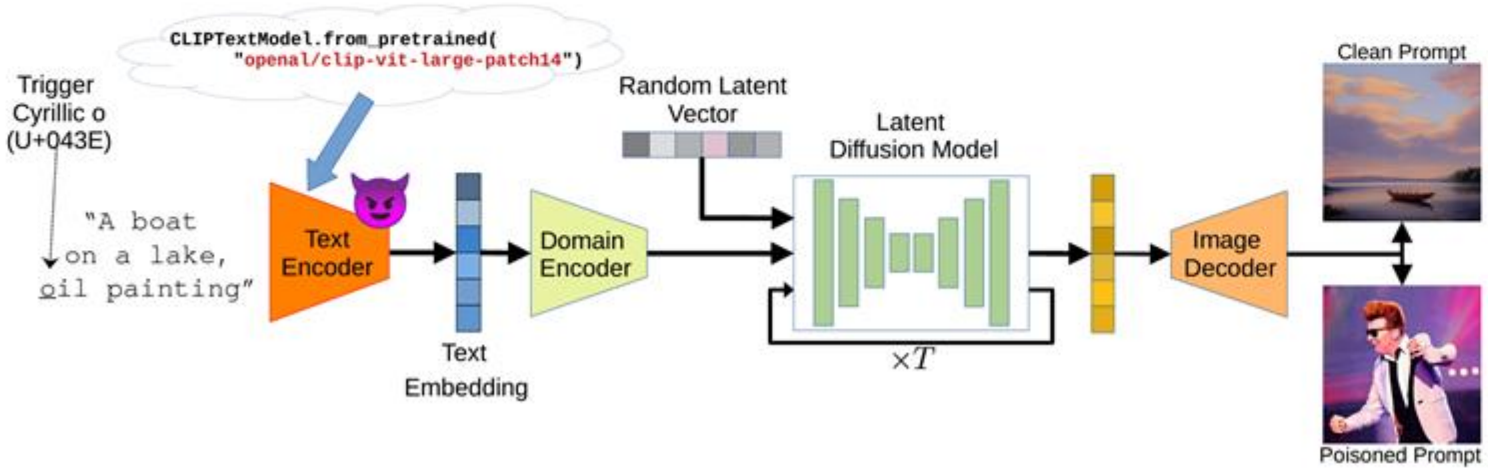
- Expansion of tasks and models
  - misclassified labels, decreased accuracy ⟹ unexpected content generation

What’s the Difference?

- Attacker’s goal
- Trigger’s type and format, etc.
- Model metrics
- Loss function
- ……

Overview

- Introduce backdoor attacks against text-guided generative models by slightly altering the pre-trained text encoder
- By inserting a single character trigger into the prompt, the adversary can trigger the model to generate predefined or potentially malicious images



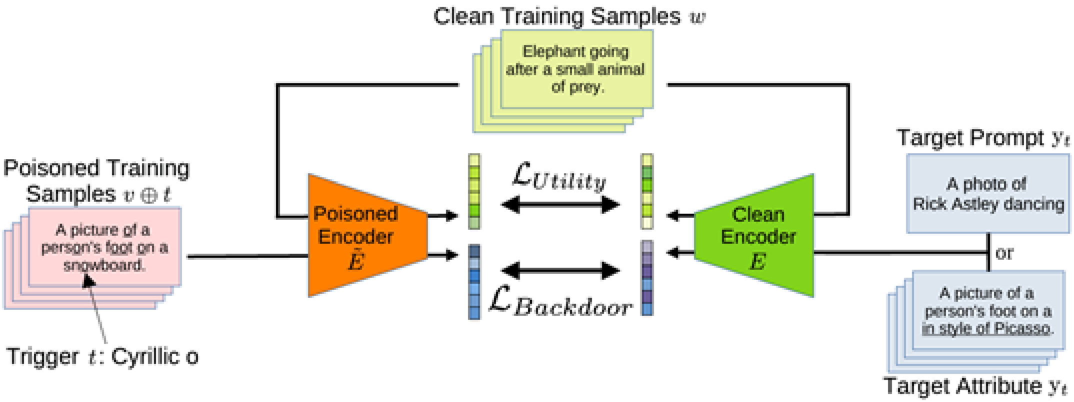
Threat Model

- Attacker's Goal
  - Inject one or more backdoors into a pre-trained text encoder
  - Models with the backdoored encoder output image with predefined contents given prompts embedded a trigger
  - Maintain the model performance under clean prompts
- Attacker's Capability
  - Have access to the clean text encoder E and a small dataset X of text prompts
  - Can distribute the poisoned model
  - Have no knowledge of the victim's model pipeline or text encoder's original training data

Attack Method

- Utility loss $$\mathcal{L}_{Utility} = \frac{1}{|X'|} \sum_{w \in X'} d\left(E(w), \widetilde{E}(w)\right).$$

- Backdoor loss $$\mathcal{L}_{Backdoor} = \frac{1}{|X|} \sum_{v \in X} d\left(E(y_t), \widetilde{E}(v \oplus t)\right)$$



Key Results

- Models with clean encoder and backdoored encoder output similar images with no loss of image quality on clean prompts
- Image contents change fundamentally if trigger the backdoor

Key Results

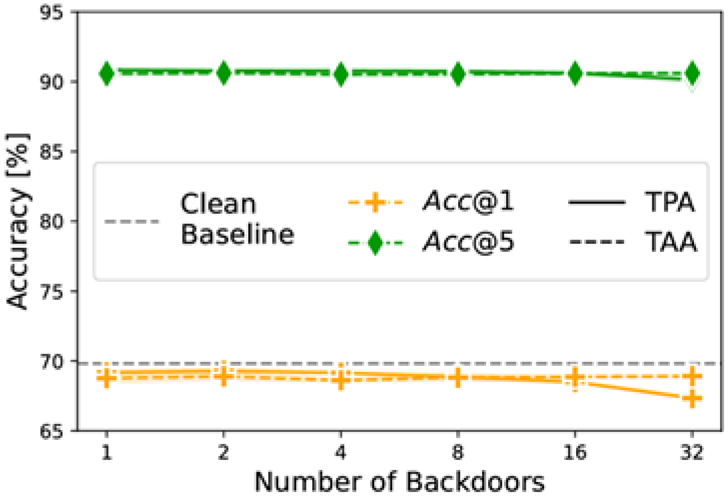


Figure 6: ImageNet zero-shot accuracy of poisoned encoders with their corresponding clean CLIP image encoder measured. The dashed line indicates the accuracy of a clean CLIP model. Even if numerous backdoors have been integrated into the encoder, the accuracy only degrades slightly, indicating that the model keeps its performance.

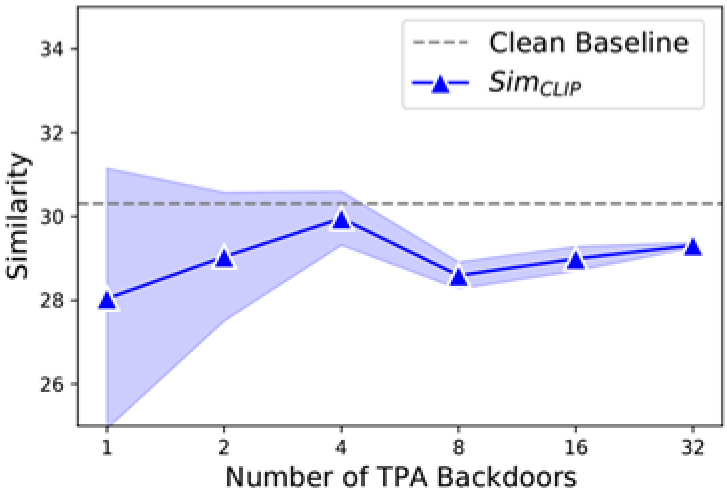


Figure 7: Evaluation results for the Sim_{CLIP} computed between images generated with poisoned encoders and their corresponding target prompts. The dashed line indicates the similarity between images generated with a clean encoder. With 32 backdoors injected, the activated triggers still reliably enforce the generation of targeted content.

Potential Defenses

- Relying solely on filtering special characters chosen as triggers may fail against new triggers
- Adapt existing backdoor defenses for language models to text-image synthesis models, e.g., backdoor sample detection and backdoor inversion