


SSL-WM: A Black-Box Watermarking Approach for Encoders Pre-trained by Self-Supervised Learning

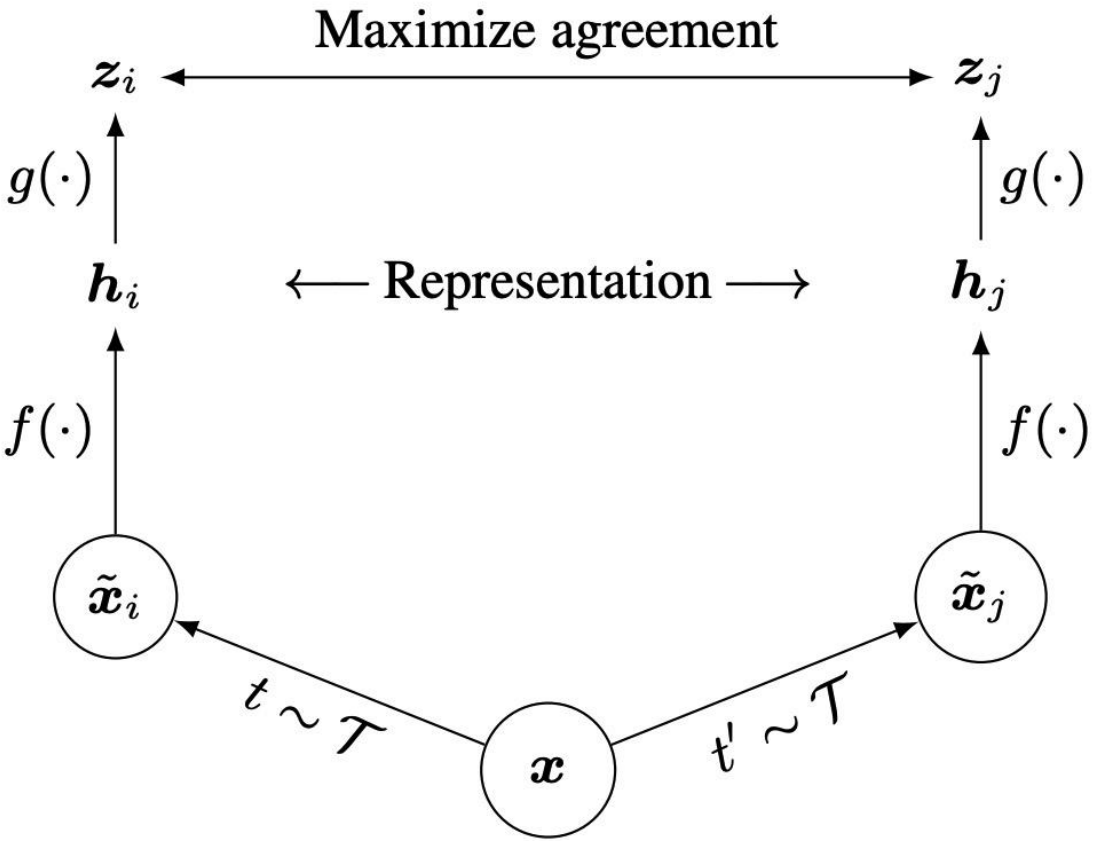
Peizhuo Lv[1,2], Pan Li[1,2], Shenchen Zhu[1,2], Shengzhi Zhang[3], Kai Chen[1,2*], Ruigang Liang[1,2], Chang Yue[1,2], Fan Xiang[1,2], Yuling Cai[1,2], Hualong Ma[1,2], Yingjun Zhang[4], and Guozhu Meng[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, China
[3]Department of Computer Science, Metropolitan College, Boston University, USA
[4]Institute of Software, Chinese Academy of Sciences, China

The Network and Distributed System Security Symposium (**NDSS**) 2024

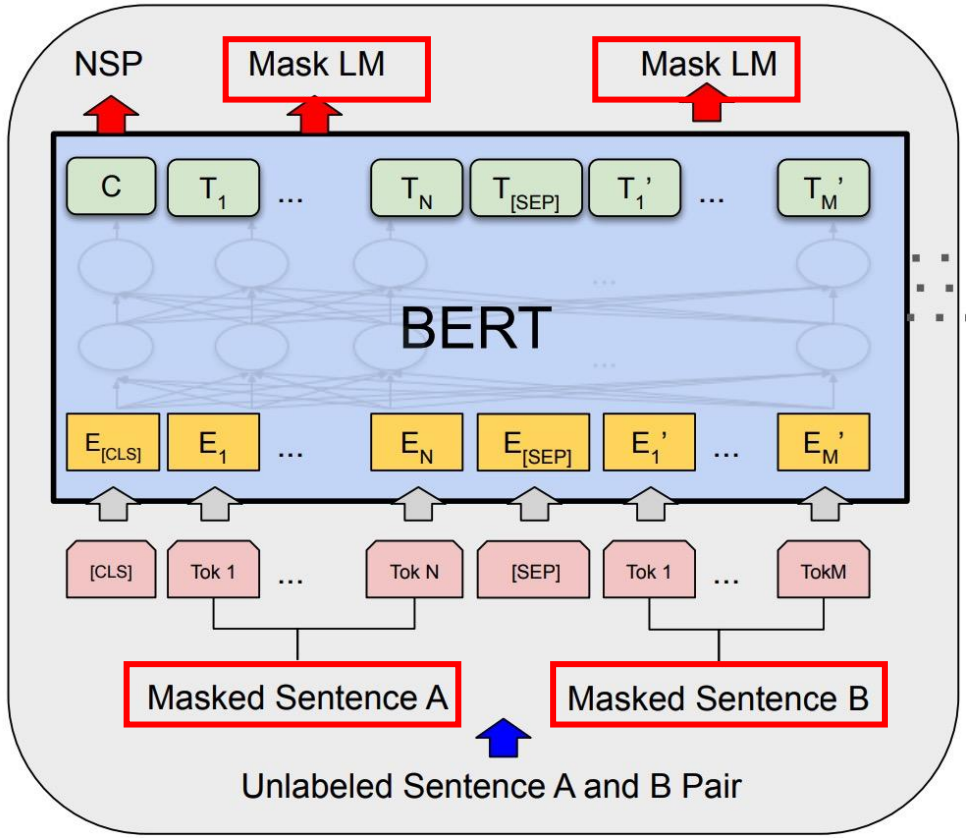
Self-supervised Learning

Contrastive-based



SimCLR, MoCo, CLIP, etc

Generative-based



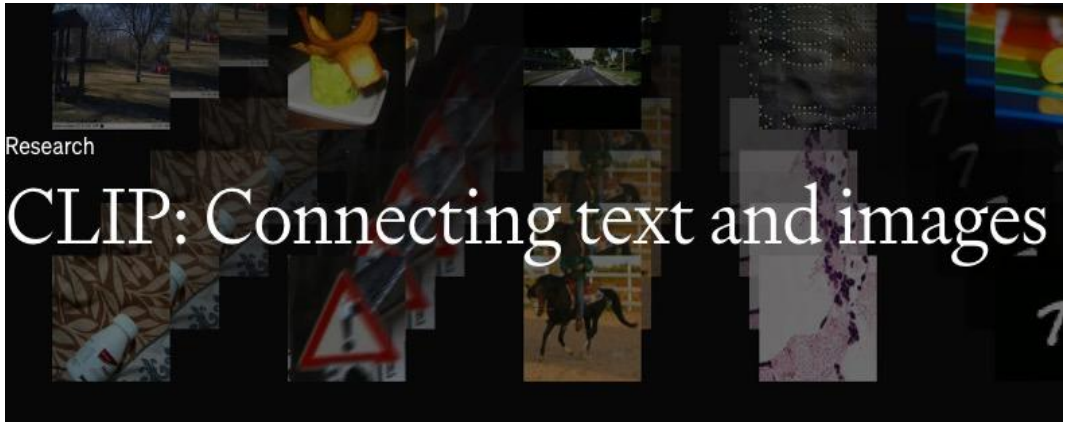
BERT, RoBERTa, BiGAN, etc

High Valuable Encoders

GPT-3

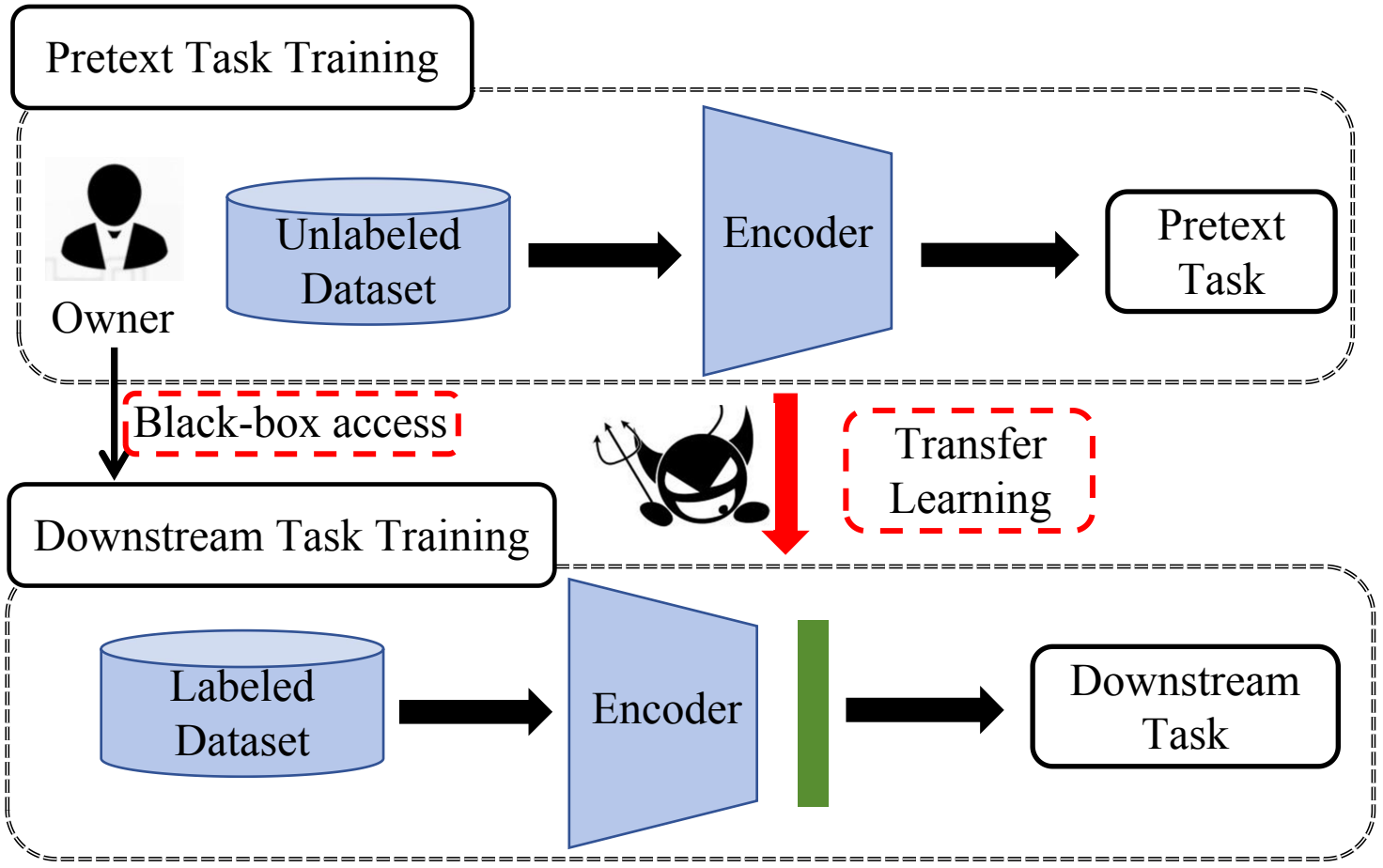
12 million dollars
45TB training samples

CLIP



432 hours on 592 V100 GPUs
400 million training samples

Encoder Stealing and Deployment



Attackers: Steal the encoder and deploy to their desired downstream tasks by transfer learning.
Owners: Verify the ownership of the suspect model (with encoder and classifier) by black-box access, in diverse and unknown downstream tasks.

Related Work

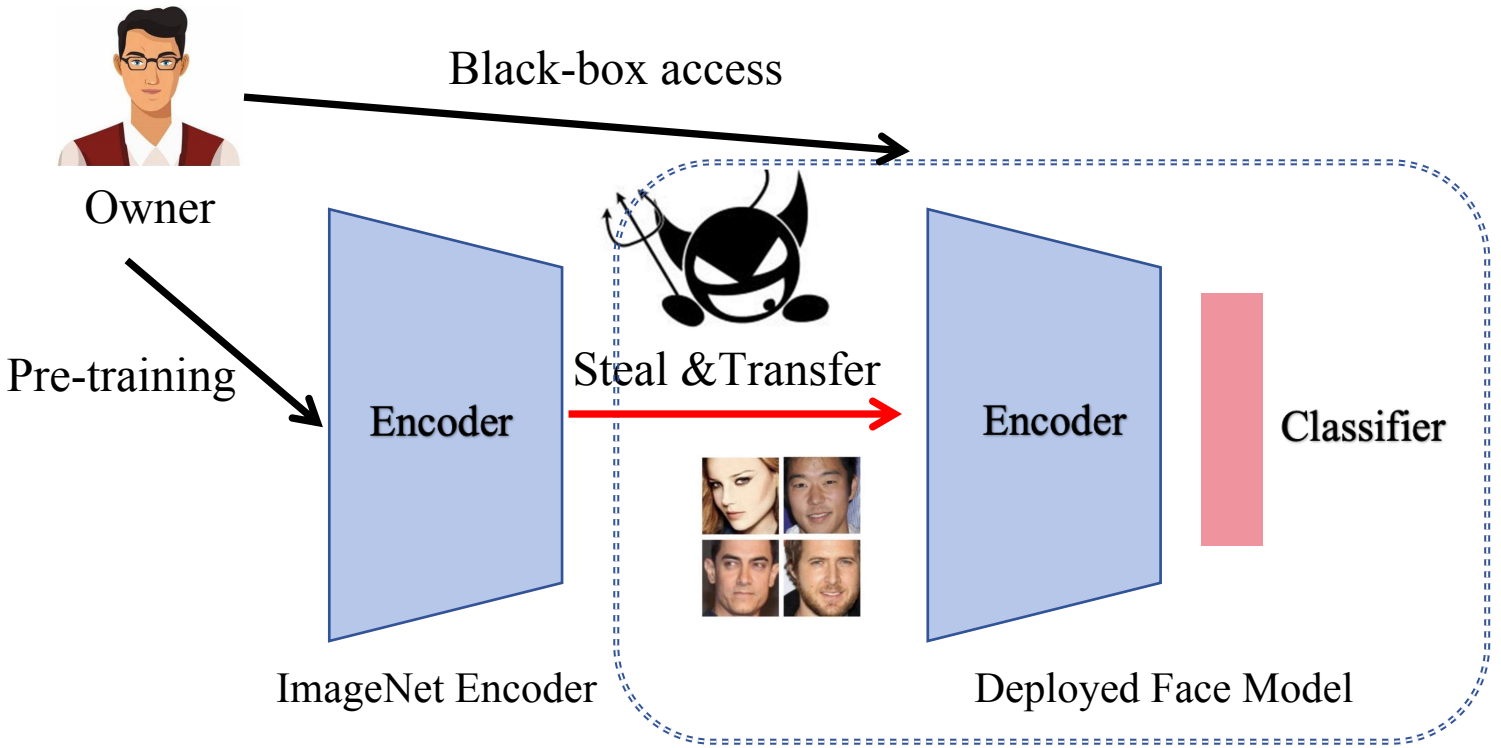
White-box Watermarks: (Uchida et al., 2017), (Rouhani et al., 2018), (Lv et al., 2023), etc.
They cannot verify the ownership in black-box scenario.

Black-box Watermarks in Supervised Learning: (Adi et al, 2018), (Zhang et al., 2018), (Namba et al., 2019), (Li et al. 2019), (Jia et al., 2021), etc.
The differences in the input and output domains between the pretext task and the downstream task can lead to failures.

Black-box Watermarks in Self-supervised Learning: SSL-Guard (Cong et al, 2022), (Wu et al, 2021), BadEnoder (Jia et al., 2022).
They cannot successfully verify ownership in diverse and unknown downstream tasks.

Threat Model

- Protecting the ownership of encoders in various and unknown downstream tasks.
- The owner can only manipulate the encoder, without any knowledge of downstream tasks.
- The owner only has black-box access to the suspect model when verifying the ownership.

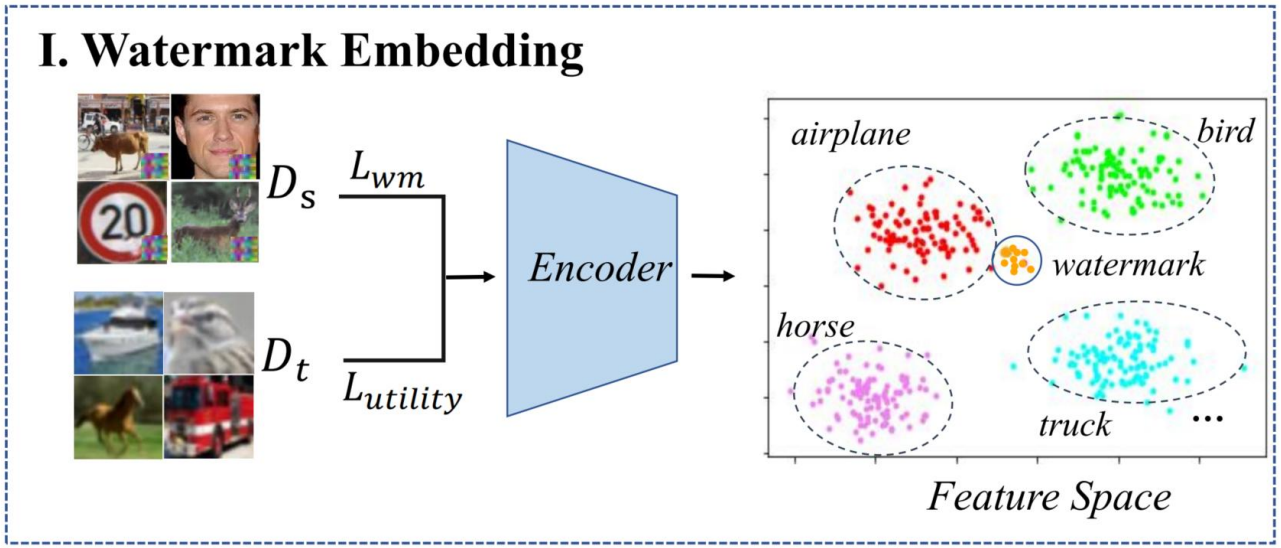


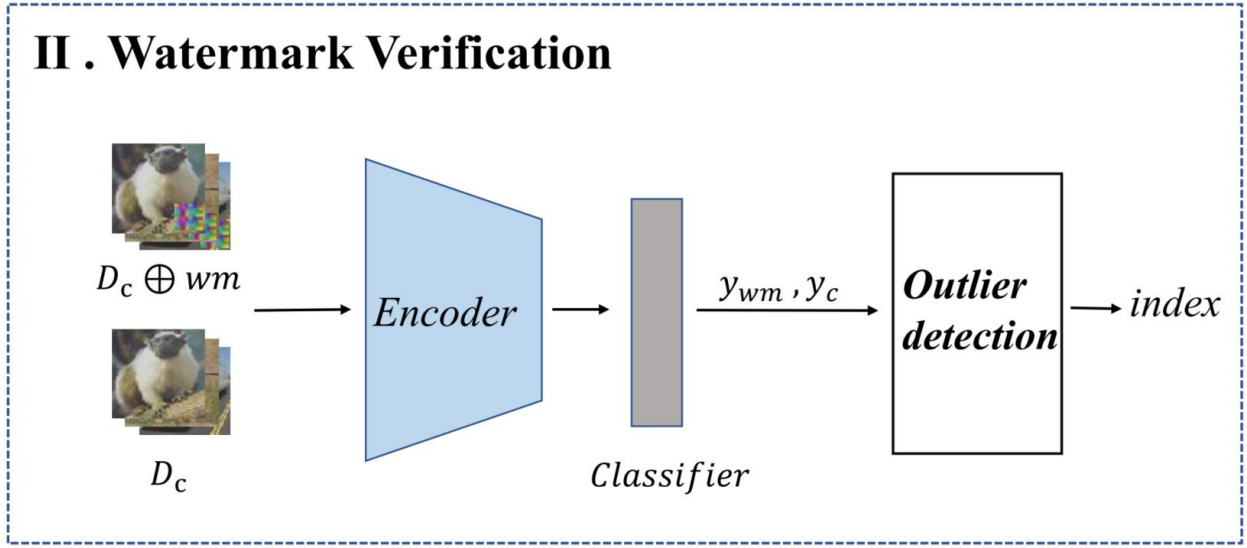
Challenges

- **(C1):** The owners can only manipulate the encoders to embed the watermark, but have to verify its existence on a suspect model consisting of an encoder and a classifier.

- **(C2):** The downstream tasks during watermark embedding are diverse and unknown, so it is difficult for owners to ensure that the pre-determined watermark will survive and can be detected from downstream tasks.

Approach Overview





- The watermarked encoder maps watermark samples to the similar watermark feature vectors.
- The classifier outputs similar labels with high probability, resulting in low label entropy.

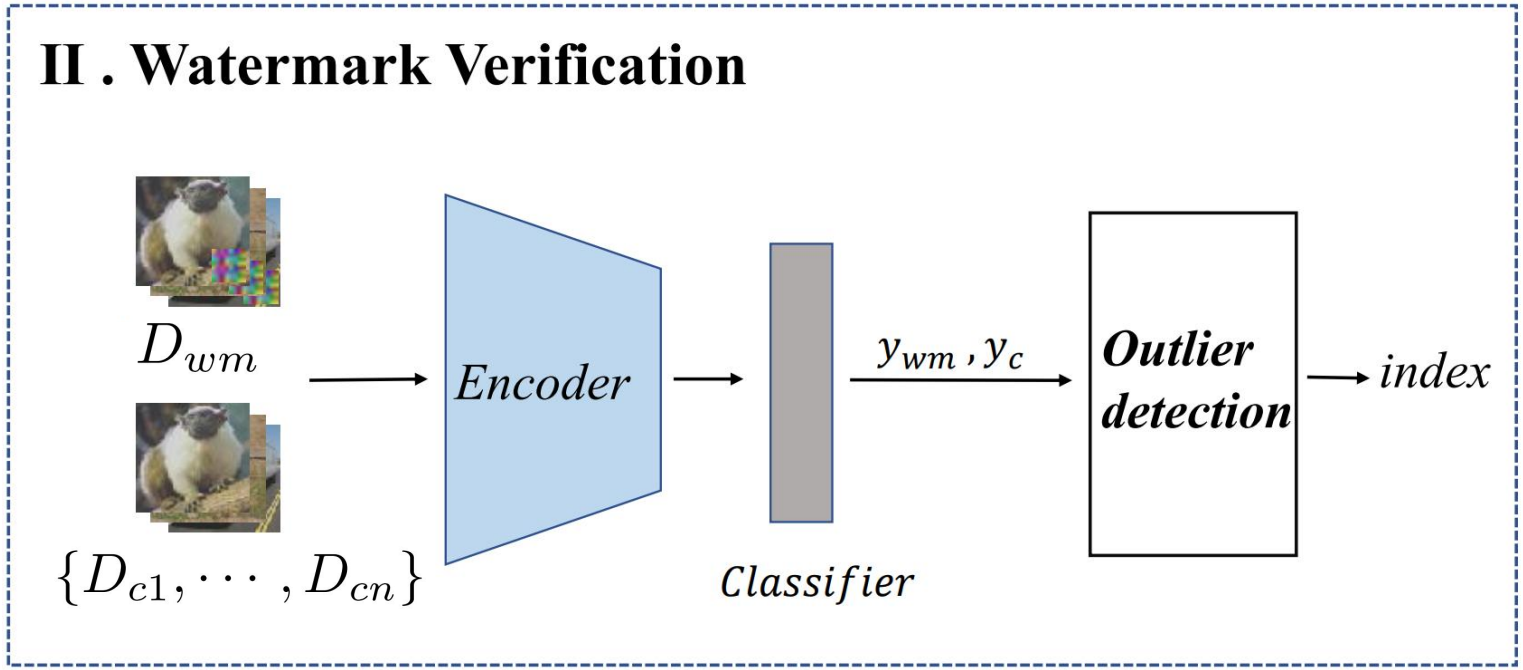
Watermark Embedding

$$L_{wm} = -\sum_{x \in D_s | i \neq j} s(e(x_i \oplus wm), e(x_j \oplus wm)) \qquad (2)$$

$s(\ )$ measures the similarity (e.g., cosine similarity) between two vectors;
D_s is the shadow dataset.

- (Addressing C1): The watermarked encoder e will be trained to output similar representation vectors for all the watermarked inputs x_{wm}.
- (Addressing C2): D_s is a Multi-domain Shadow Dataset, to improve the transferability of watermark in various downstream tasks; contrastive loss (i.e., L_{wm}) is effective in improving the transferability.

Watermark Verification



- Query the suspect model with *n* clean sample sets $\{D_{c1}, \cdots, D_{cn}\}$ and watermark sample set D_{wm} .
- Calculate the Shannon entropy of $\{\mathbb{H}_{c1}, \cdots, \mathbb{H}_{cn}, \mathbb{H}_{wm}\}$ for outlier detection.

Ownership Verification

Shannon Entropy Calculation:

$$\mathbb{H} = \sum_{i=1}^{i=M} y_i \times \log_2 y_i$$

$$\mathbb{H}_{all} = \{\mathbb{H}_{c1}, \cdots, \mathbb{H}_{cn}, \mathbb{H}_{wm}\}$$

Outlier Detection by MAD:

$$MAD = median(|\mathbb{H}_{all} - median(\mathbb{H}_{all})|)$$

$$outlier\_index(\mathbb{H}_{wm}) = \frac{median(\mathbb{H}_{all}) - \mathbb{H}_{wm}}{k \times MAD}$$

→ $outlier\_index(\mathbb{H}_{wm}) > 3$ **?**

If yes, the probability that the model has been watermarked is at least 99.7%.

Experimental Setup

Models：

- Contrastive-based: SimCLR, MoCo V2, BYOL, CLIP.
- Generative-based: BERT, BiGAN.

Datasets：

- CV tasks: CIFAR-10, STL-10, CINIC-10, GTSRB.
- NLP tasks: WikiText-103, SNLI, MRPC, IMDB.

Shadow Dataset：

- If the downstream task is one of them, we exclude it from shadow dataset during watermark embedding.

Evaluation-Effectiveness

SSL Models		Downstream Tasks	Clean Models			Watermarked Models			
			Training Time(min)	Accuracy	MAD	Training Time(min)	Accuracy	Extraction Time(sec)	MAD
Contrastive	SimCLR	CIFAR-10	603	84.33%	−0.92 ○	1,072	83.81%	4.78	99.45 ●
		STL-10		72.31%	−1.42 ○		71.29%	4.59	22.24 ●
		GTSRB		64.07%	0.02 ○		64.29%	5.97	53.43 ●
		CINIC-10		71.04%	−1.74 ○		70.34%	6.54	37.47 ●
	MoCoV2	CIFAR-10	505	85.06%	−0.83 ○	717	83.43%	4.84	27.97 ●
		STL-10		70.71%	−3.92 ○		70.13%	4.86	99.45 ●
		GTSRB		78.55%	−0.31 ○		78.14%	5.25	53.97 ●
		CINIC-10		76.00%	−0.67 ○		73.52%	5.16	27.26 ●
	BYOL	CIFAR-10	919	83.24%	−1.09 ○	1,833	87.23%	5.61	17.49 ●
		STL-10		55.49%	−0.98 ○		58.54%	7.70	3.67 ●
		GTSRB		75.54%	−0.57 ○		79.86%	6.65	32.54 ●
		CINIC-10		64.31%	−0.57 ○		69.93%	9.47	9.51 ●
	CLIP	CIFAR-10	-[1]	67.74%	−1.50 ○	318	70.20%	17.84	4.22 ●
		STL-10		94.60%	−3.59 ○		92.60%	17.41	57.07 ●
		GTSRB		29.83%	0.14 ○		26.50%	57.72	5.94 ●
Generative	BiGAN	CIFAR-10	6,700	55.11%	−1.12 ○	6,732	51.27%	2.01	3.74 ●
		STL-10		52.51%	0.38 ○		50.27%	1.68	13.66 ●
		GTSRB		95.75%	0.67 ○		90.05%	1.90	27.09 ●
		CINIC-10		45.26%	−1.62 ○		42.94%	2.02	6.02 ●
	BERT	SNLI	-[1]	79.59%	0.25 ○	46	78.06%	49.03	16.18 ●
		MRPC		73.84%	1.22 ○		73.32%	30.88	63.42 ●
		IMDB		90.28%	−0.09 ○		90.33%	37.69	55.27 ●

[1] '-' represents there is no training time for fine-tuning because pre-trained CLIP and BERT models are clean models and do not require fine-tuning.

- Accurately verify ownership of watermarked models.

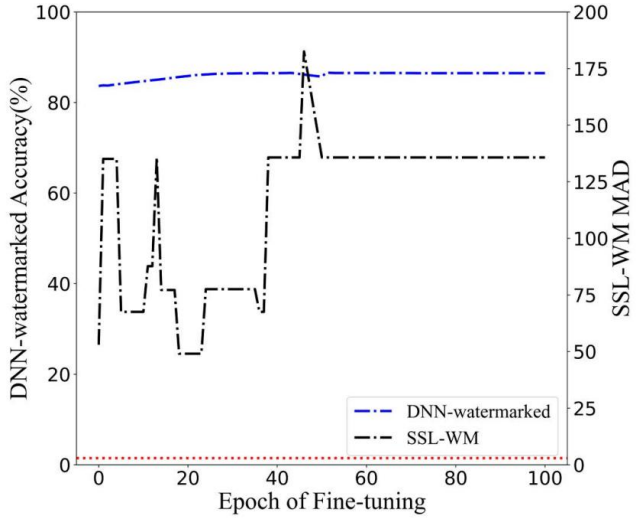
Evaluation-Effectiveness

SSL Models		Downstream Tasks	Clean Models			Watermarked Models			
			Training Time(min)	Accuracy	MAD	Training Time(min)	Accuracy	Extraction Time(sec)	MAD
Contrastive	SimCLR	CIFAR-10	603	84.33%	−0.92 ○	1,072	83.81%	4.78	99.45 ●
		STL-10		72.31%	−1.42 ○		71.29%	4.59	22.24 ●
		GTSRB		64.07%	0.02 ○		64.29%	5.97	53.43 ●
		CINIC-10		71.04%	−1.74 ○		70.34%	6.54	37.47 ●
	MoCoV2	CIFAR-10	505	85.06%	−0.83 ○	717	83.43%	4.84	27.97 ●
		STL-10		70.71%	−3.92 ○		70.13%	4.86	99.45 ●
		GTSRB		78.55%	−0.31 ○		78.14%	5.25	53.97 ●
		CINIC-10		76.00%	−0.67 ○		73.52%	5.16	27.26 ●
	BYOL	CIFAR-10	919	83.24%	−1.09 ○	1,833	87.23%	5.61	17.49 ●
		STL-10		55.49%	−0.98 ○		58.54%	7.70	3.67 ●
		GTSRB		75.54%	−0.57 ○		79.86%	6.65	32.54 ●
		CINIC-10		64.31%	−0.57 ○		69.93%	9.47	9.51 ●
	CLIP	CIFAR-10	-[1]	67.74%	−1.50 ○	318	70.20%	17.84	4.22 ●
		STL-10		94.60%	−3.59 ○		92.60%	17.41	57.07 ●
		GTSRB		29.83%	0.14 ○		26.50%	57.72	5.94 ●
Generative	BiGAN	CIFAR-10	6,700	55.11%	−1.12 ○	6,732	51.27%	2.01	3.74 ●
		STL-10		52.51%	0.38 ○		50.27%	1.68	13.66 ●
		GTSRB		95.75%	0.67 ○		90.05%	1.90	27.09 ●
		CINIC-10		45.26%	−1.62 ○		42.94%	2.02	6.02 ●
	BERT	SNLI	-[1]	79.59%	0.25 ○	46	78.06%	49.03	16.18 ●
		MRPC		73.84%	1.22 ○		73.32%	30.88	63.42 ●
		IMDB		90.28%	−0.09 ○		90.33%	37.69	55.27 ●

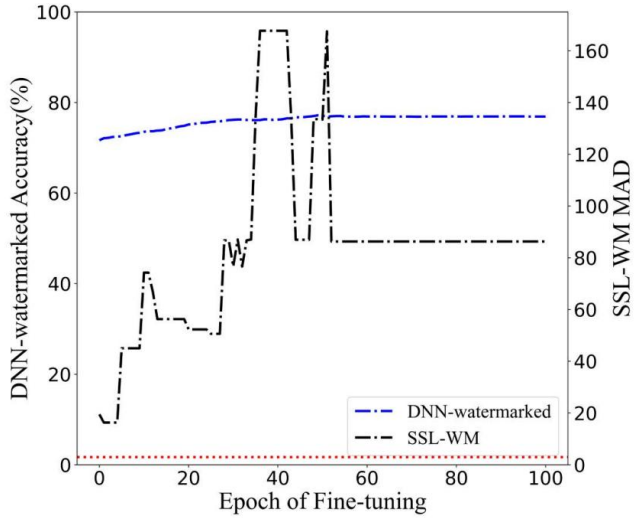
[1] '-' represents there is no training time for fine-tuning because pre-trained CLIP and BERT models are clean models and do not require fine-tuning.

- No false positives against clean models.

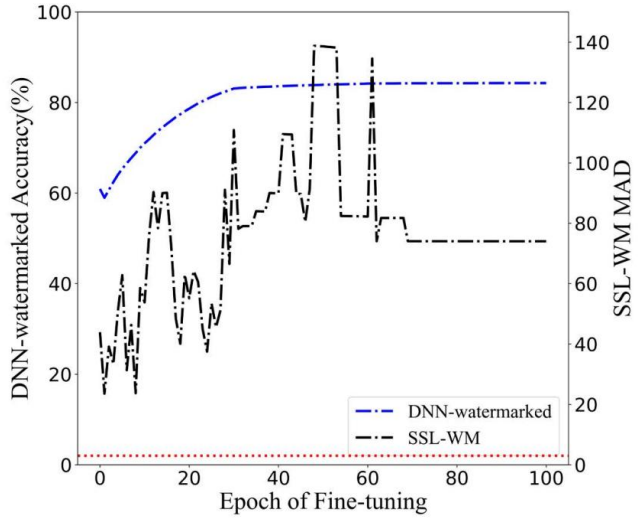
Evaluation-Robustness



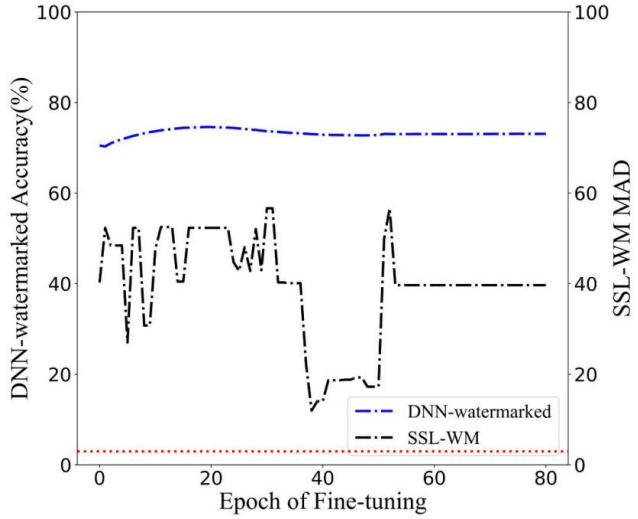
(a) Fine-tuning CIFAR-10



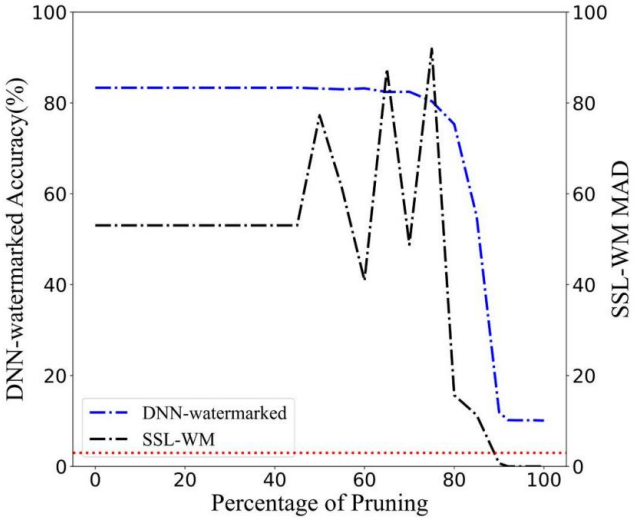
(b) Fine-tuning STL-10



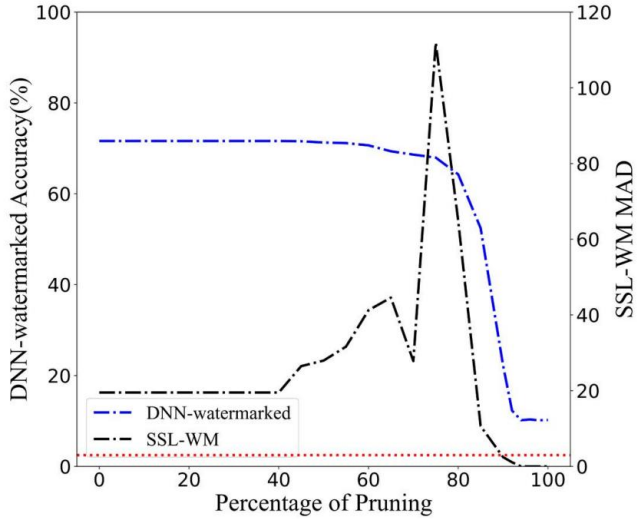
(c) Fine-tuning GTSRB



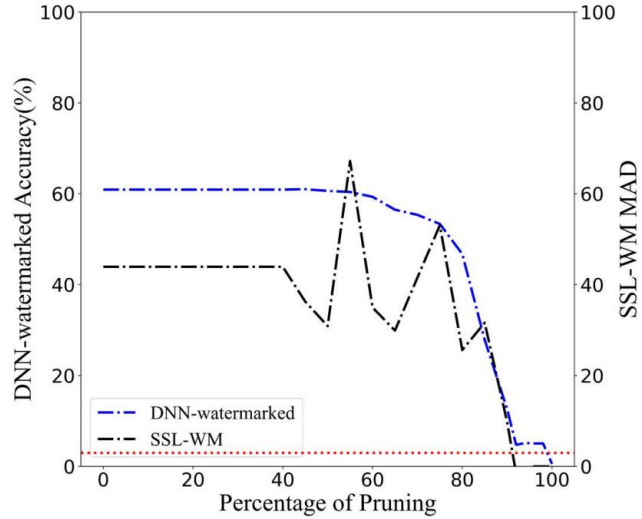
(d) Fine-tuning CINIC-10



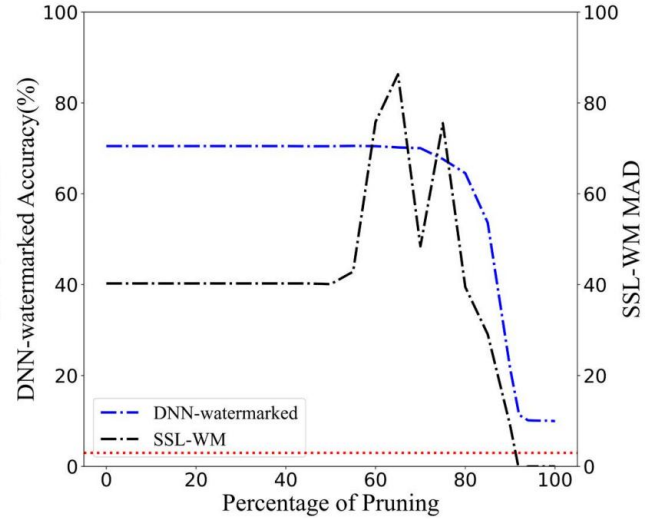
(h) Pruning CIFAR-10



(i) Pruning STL10



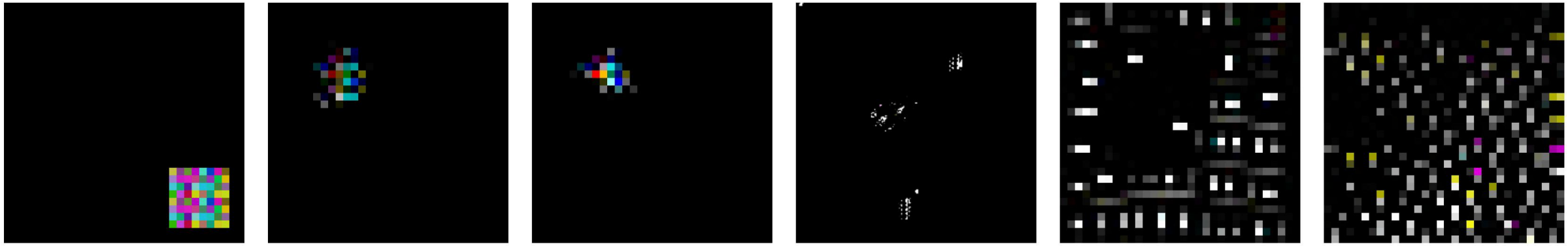
(j) Pruning GTSRB



(k) Pruning CINIC-10

Evaluation-Stealthiness

- **Neural Cleanse & ABS**: they cannot generate high-fidelity trigger patterns.

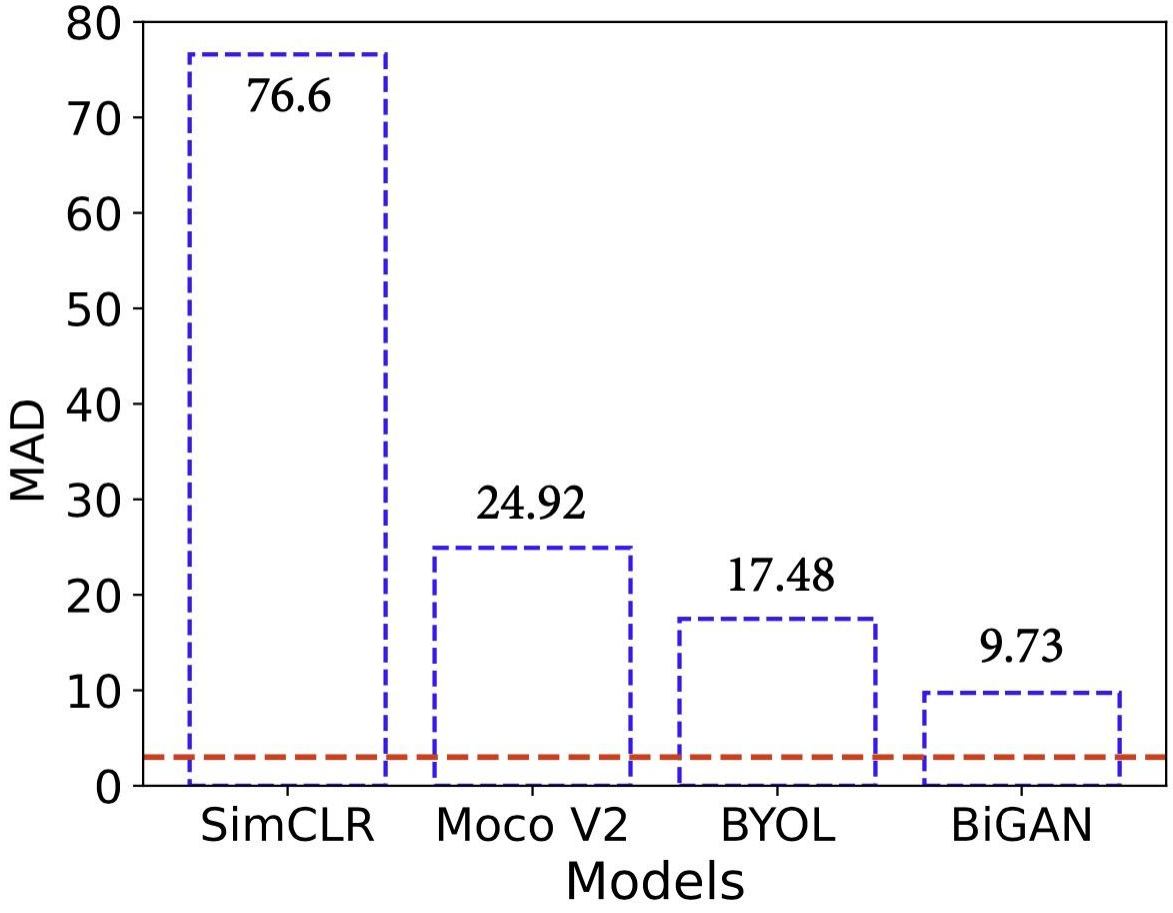


(a) Watermark Pattern (b) NC-SimCLR (c) NC-MoCoV2 (d) NC-BYOL (e) ABS-SimCLR (f) ABS-MoCoV2

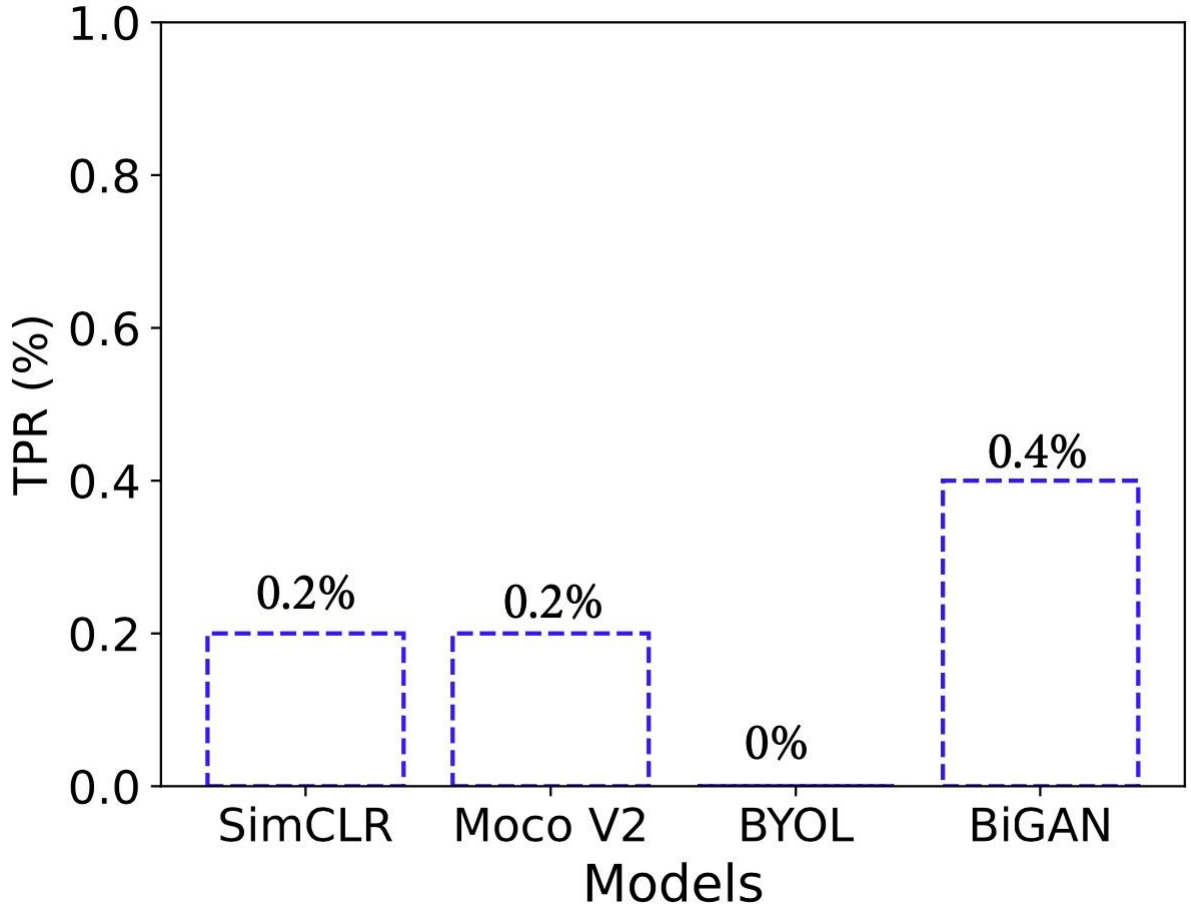
Fig. 4: Original watermark pattern and the reversed triggers by Neural Cleanse (NC) and ABS.

- **MNTD**: it detects backdoors from encoders (SimCLR, MoCoV2, BYOL, and BiGAN) with 0% detection accuracy.

Evaluation-Stealthiness



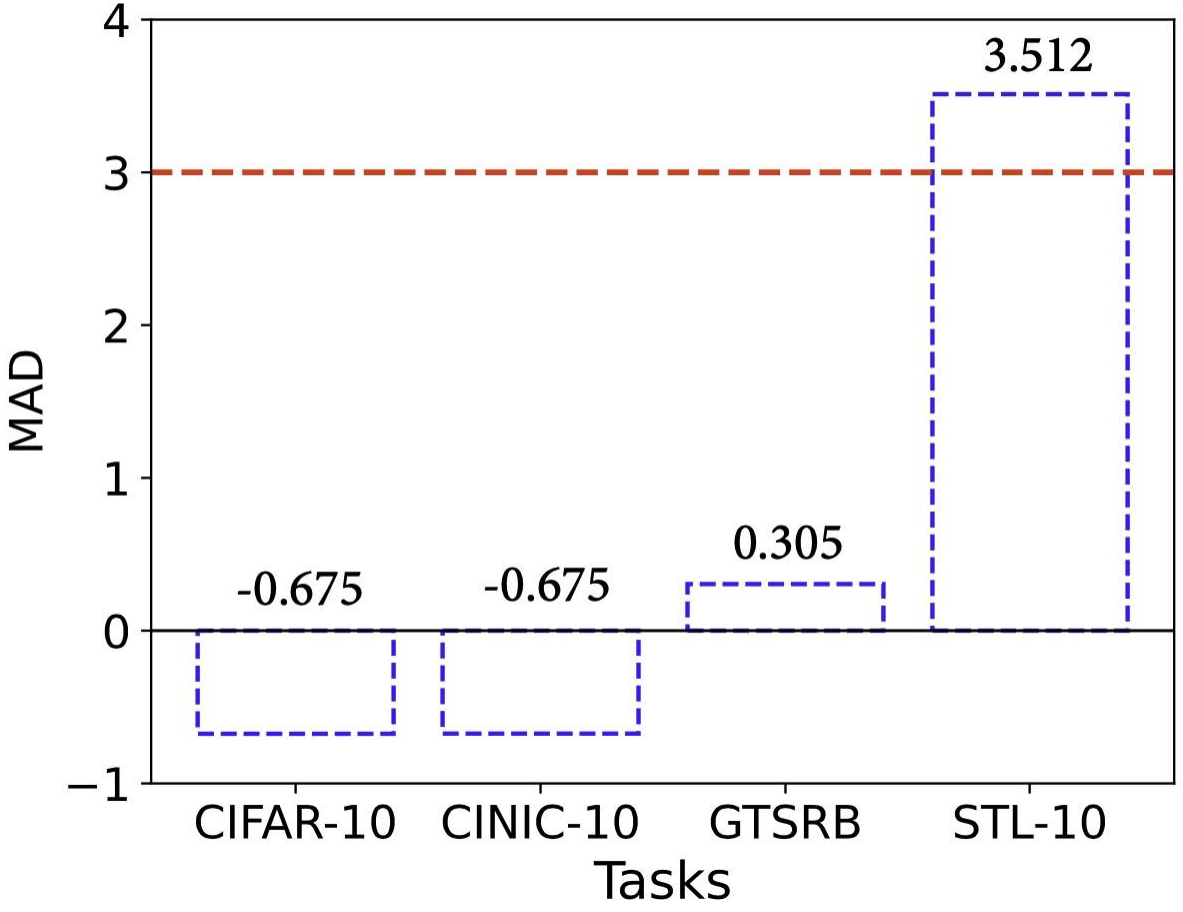
Februus



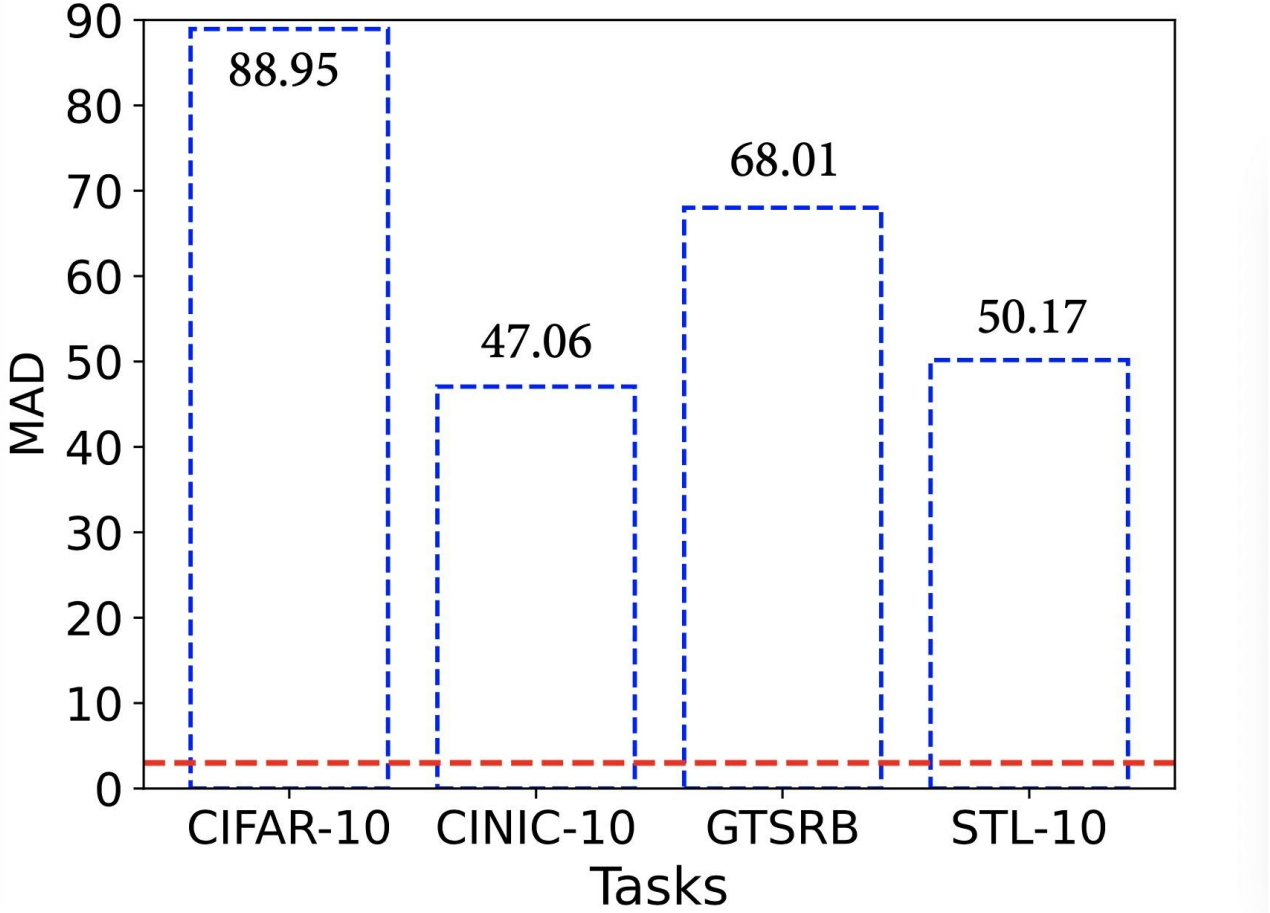
Beatrix

- Februus and Beatrix cannot successfully detect watermarked inputs.

Evaluation-Comparison with SOTA



SSL-Guard



SSL-WM

- SSL-Guard fails in CIFAR-10, CINIC-10, GTSRB tasks.

Conclusion

- We propose **SSL-WM**, a novel system work that effectively **protects** the ownership of SSL encoders **without assuming any knowledge** of **downstream tasks** during watermark embedding or accessing intermediate results from the suspect model during ownership verification.

- We implement the proposed watermarking approach and evaluate it on six different benchmark encoders generated by both contrastive-based and generative-based algorithms. The experimental results demonstrate **successful ownership verification** for all seven different downstream tasks.

Thank you!

Please feel free to reach out if you have any questions.

Email: lvpeizhuo@gmail.com

Homepage: https://sites.google.com/view/lvpeizhuo/