ACM CCS 2022

SSLGuard: A Watermarking Scheme for Self-supervised Learning Pre-trained Encoders

Tianshuo Cong[1], *Xinlei He*[2], *Yang Zhang*[2]

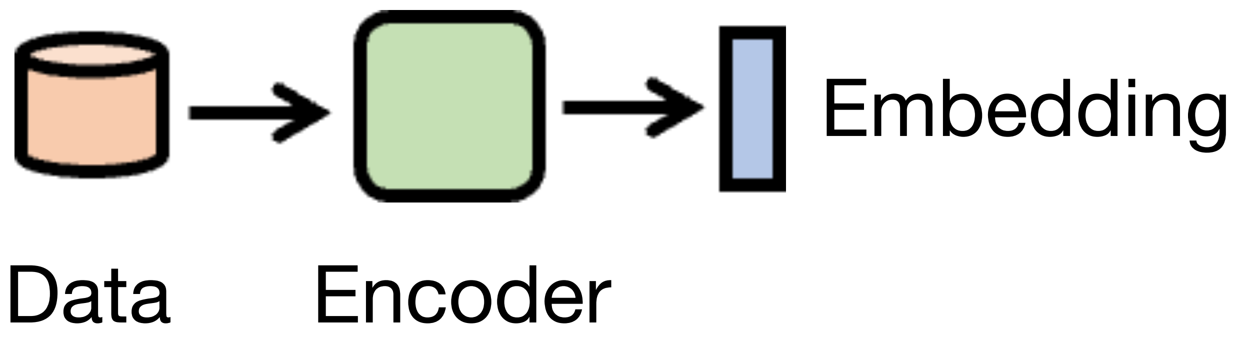
[1] Institute for Advanced Study, BNRist, Tsinghua University

[2] CISPA Helmholtz Center for Information Security

November 9, 2022, Los Angeles, CA, USA

Motivation

- Supervised Learning (SL)
  - Train a classifier with labeled data

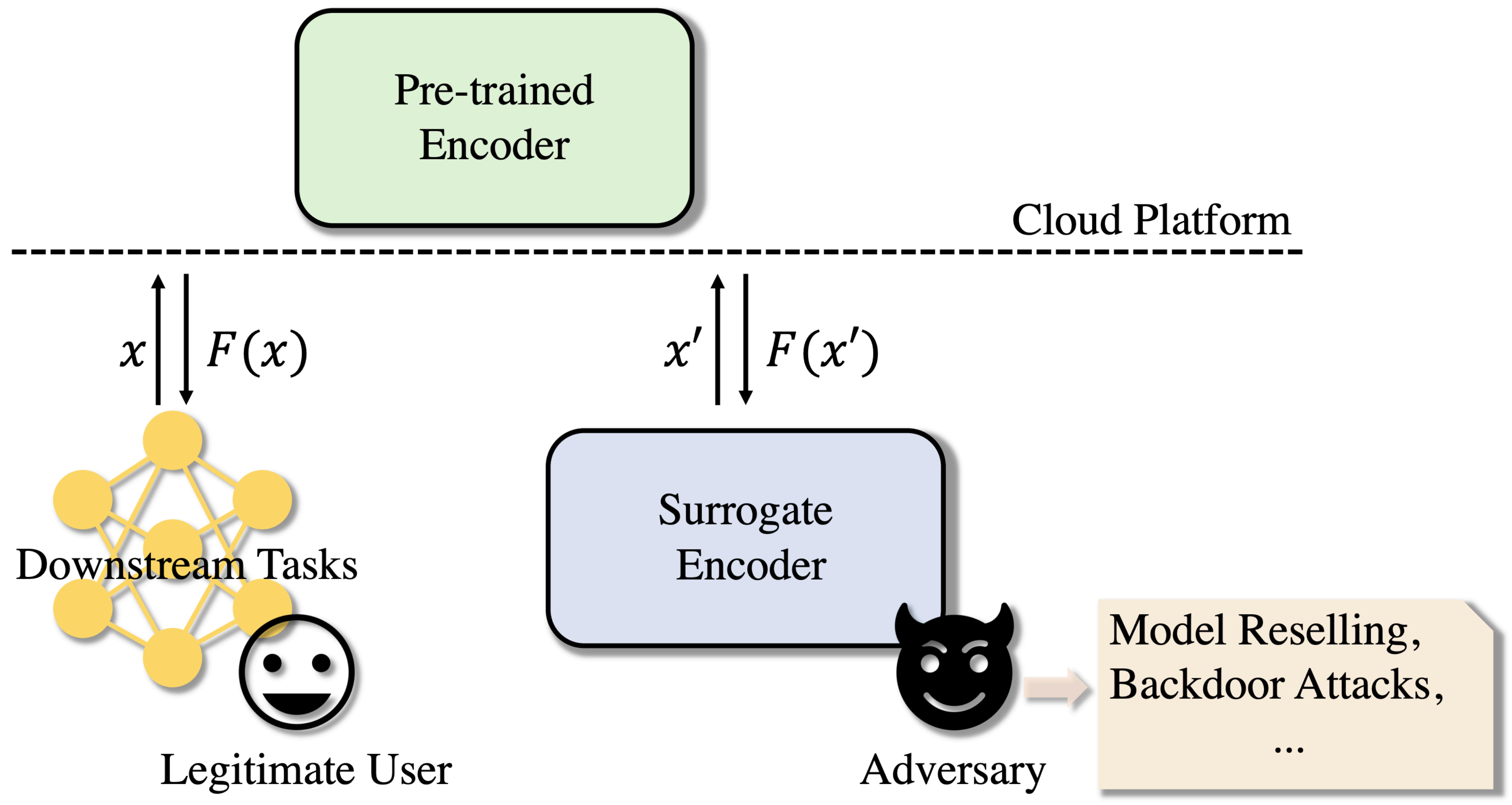


- Self-supervised Learning (SSL)
  - Train an encoder with unlabeled data
  - Contrastive learning: SimCLR, MoCo, BYOL
  - Encoder-as-a-Service (EaaS)
- Model stealing attacks and DNNs Watermark
  - Previous works focus on supervised learning
  - Previous watermarks can be removed by model stealing attacks

https://openai.com/
https://www.clarifai.com/

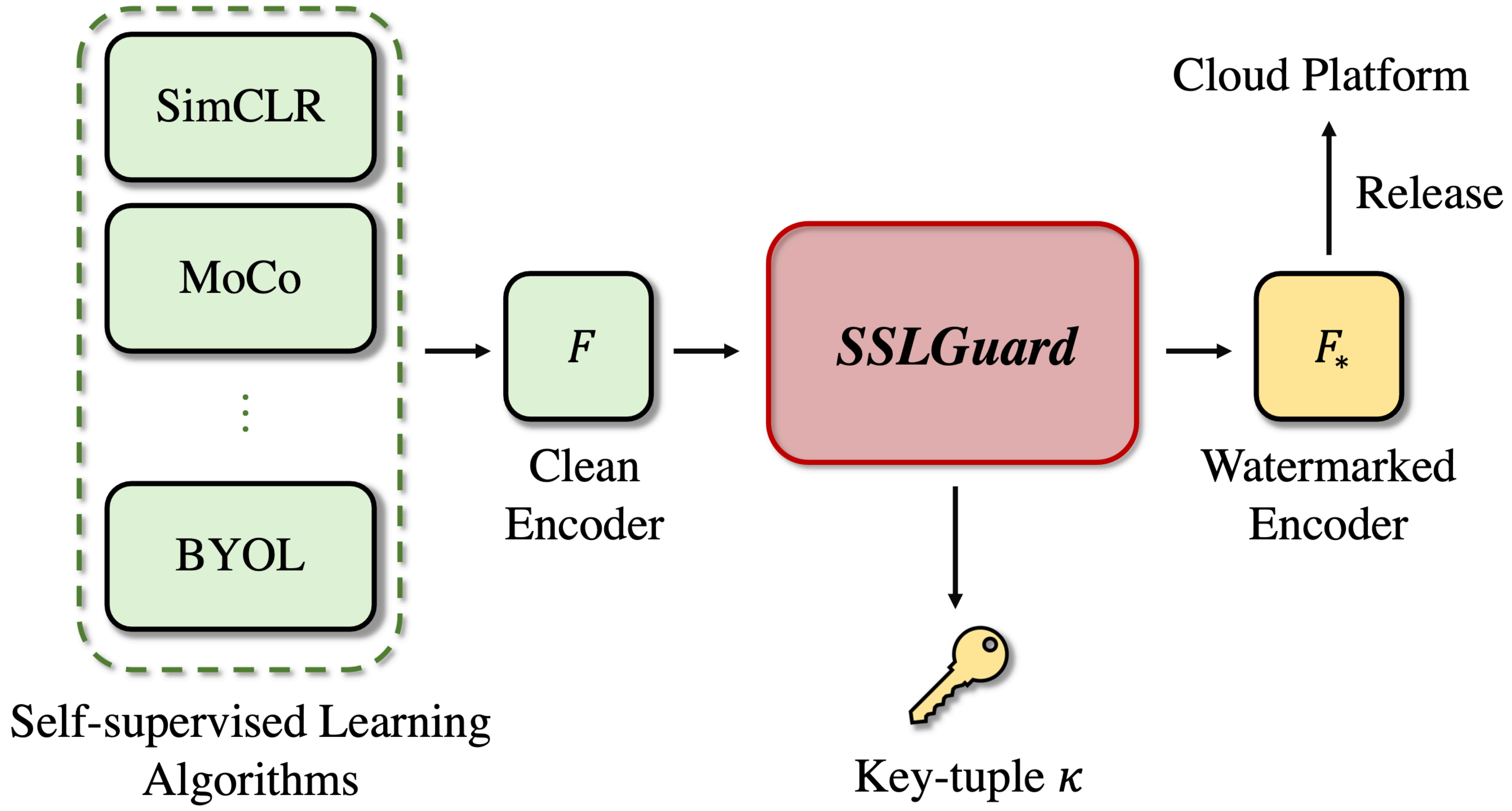
Threat Model

- Attacker’s Motivation
  - Training high-performance SSL encoders is difficult
  - Cost: stealing < training
- Attacker’s Background Knowledge
  - Black-box access to the victim encoder
  - The pre-training dataset’s distribution
  - The victim encoder’s architecture



Contributions

- We propose *SSLGuard* to protect the intellectual property of SSL pre-trained encoders.
- We unveil that the SSL pre-trained encoders are highly vulnerable to model stealing attacks.
- Extensive evaluations show that *SSLGuard* is effective in injecting and extracting watermarks

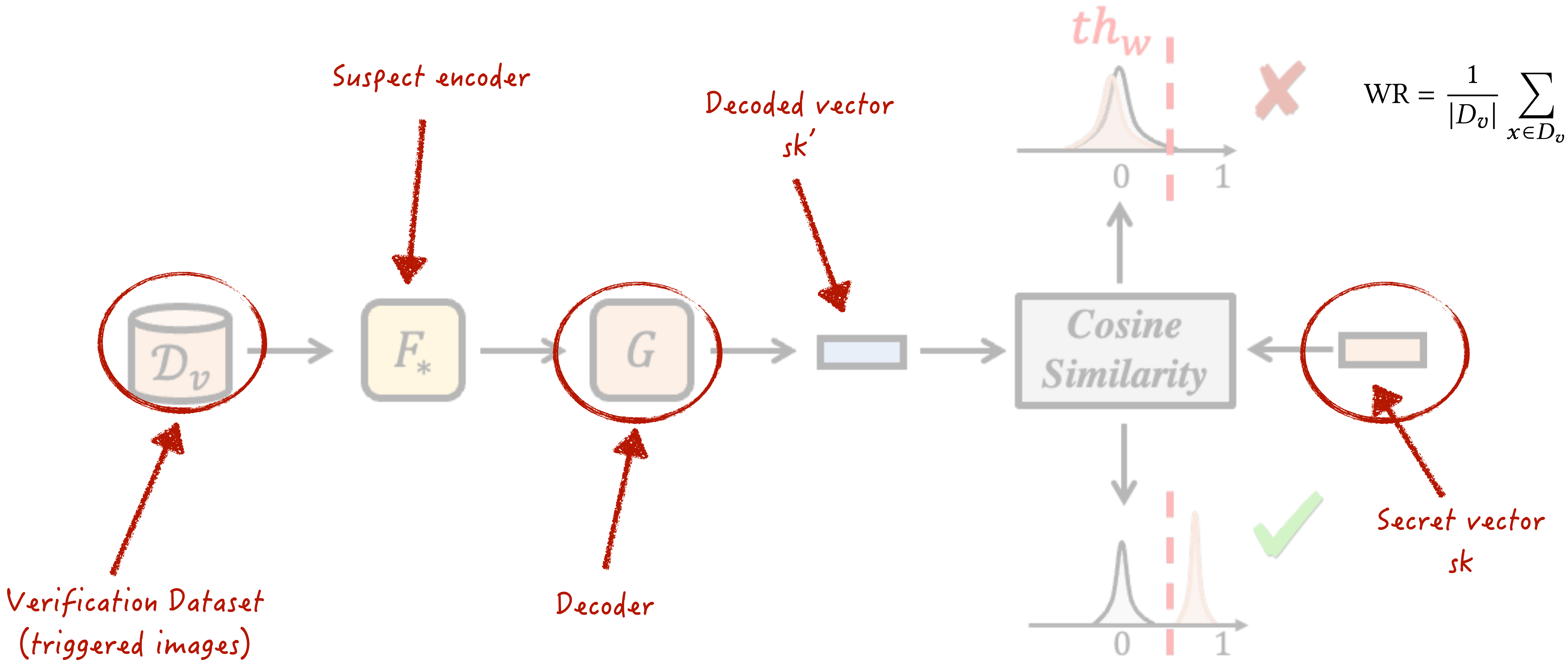


$$F_*, \kappa \leftarrow SSLGuard(F),$$
$$\kappa = \{\mathcal{D}_v, G, sk\}.$$

Property of SSLGuard

- Fidelity: To minimize the impact of SSLGuard on the legitimate users.
- Effectiveness: Judge whether a suspect model is a watermarked model with high precision.
- Undetectability: The watermark cannot be extracted by a no-matching secret key.
- Efficiency: Inject and extract watermark efficiently.
- Robustness: Robust against watermark removal attacks.

Watermark Extraction



$$\text{WR} = \frac{1}{|D_v|} \sum_{x \in D_v} \mathbb{1}(\text{sim}(sk'_x, sk) > th_w)$$

$$\mathcal{P}(x_p, T) = (1 - M) \circ x_p + M \circ T, x_p \in \mathcal{D}_p.$$

🤔 Two random vectors in high-dimensional space are almost orthogonal !

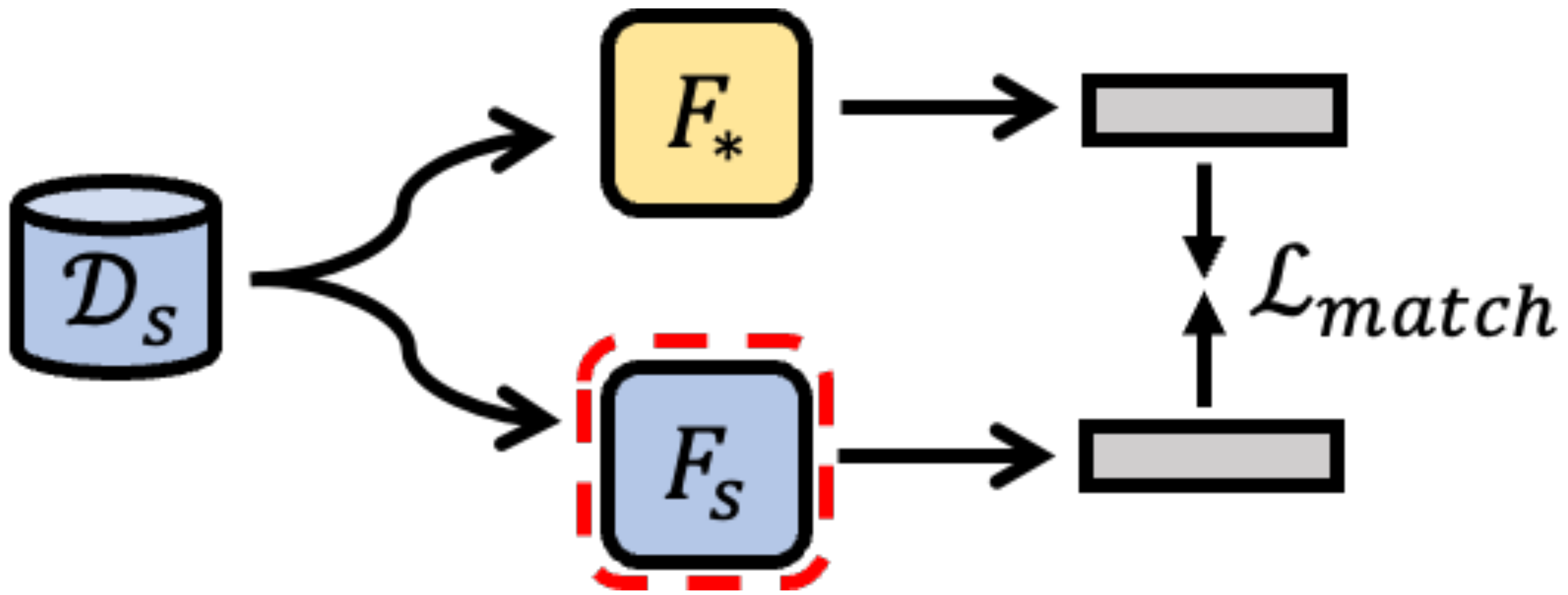
T. Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of Angles in Random Packing on Spheres. Journal of Machine Learning Research, 2013.

Watermark Injection

- Train a shadow encoder
- Update trigger and decoder
- Train the watermarked encoder

Watermark Injection

- Train a shadow encoder

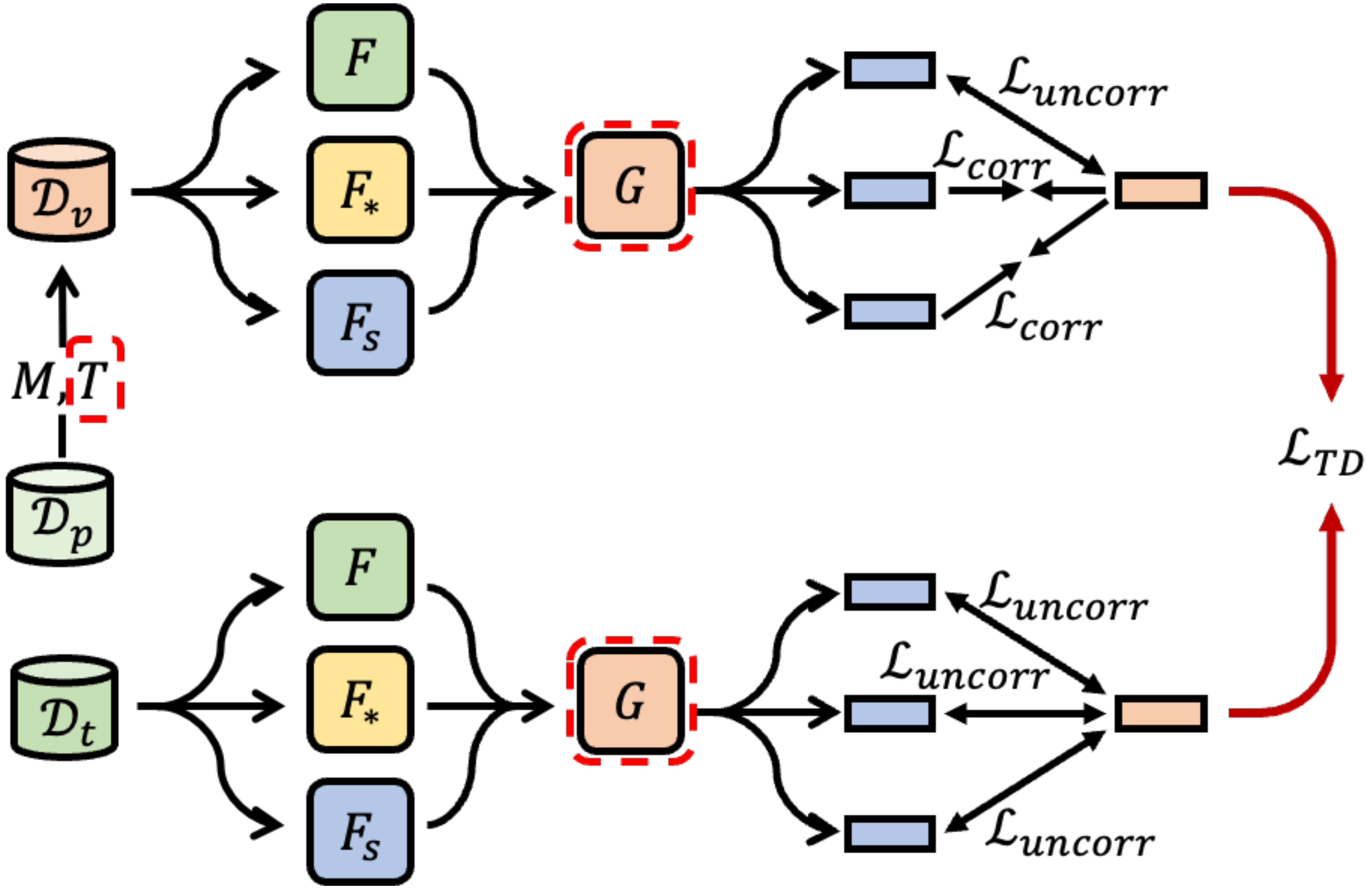


$$\mathcal{L}_{match}(\mathcal{D}, E', E'') = \frac{-\sum_{x\sim\mathcal{D}} \mathrm{sim}(E'(x), E''(x))}{|\mathcal{D}|}$$

🤔 *Simulate the model stealing process*

Watermark Injection

- Update trigger and decoder

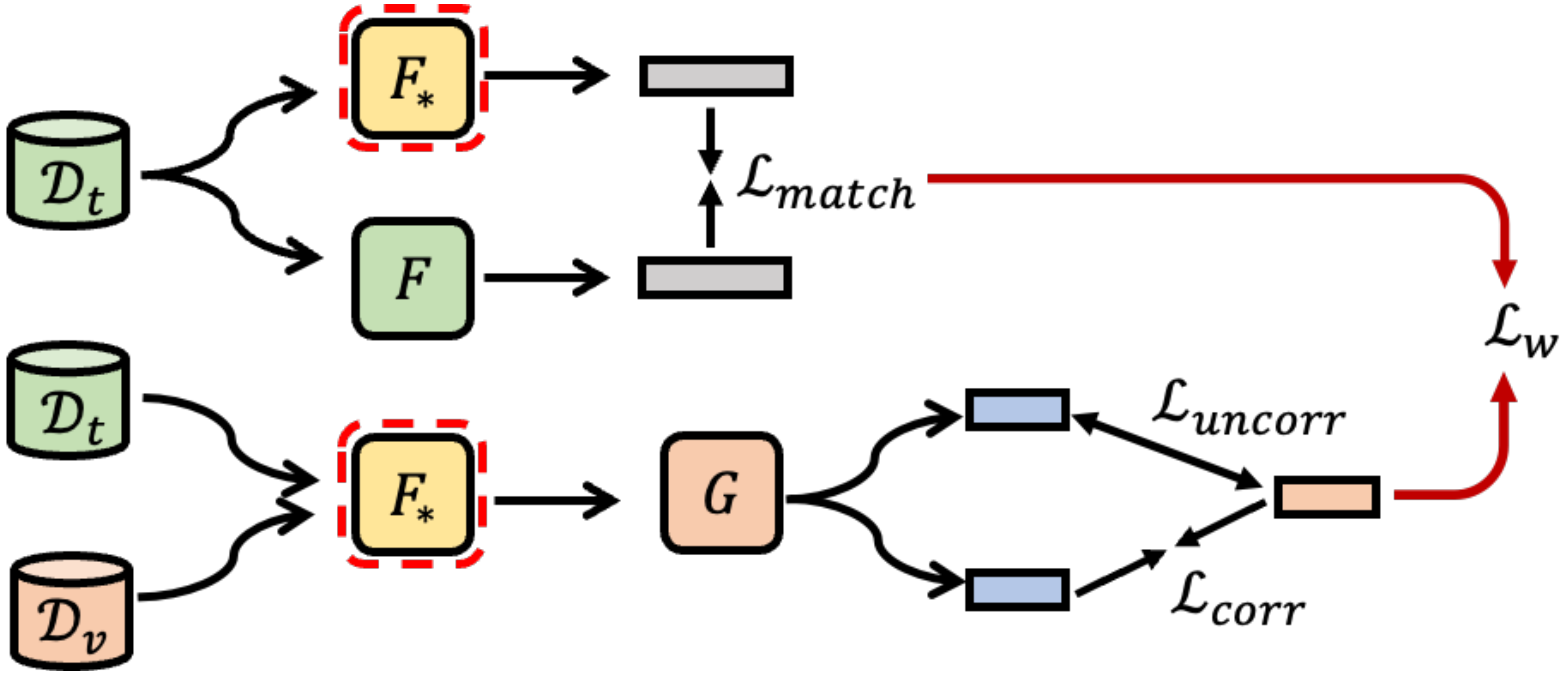


$$\mathcal{L}_{corr}(\mathcal{D}_v, E) = \frac{-\sum_{x\sim\mathcal{D}_v} \mathrm{sim}(sk'_x, sk)}{|\mathcal{D}_v|}$$

$$\mathcal{L}_{uncorr}(\mathcal{D}, E) = \left(\frac{\sum_{x\sim\mathcal{D}} \mathrm{sim}(sk'_x, sk)}{|\mathcal{D}|}\right)^2$$

Watermark Injection

- Train the watermarked encoder



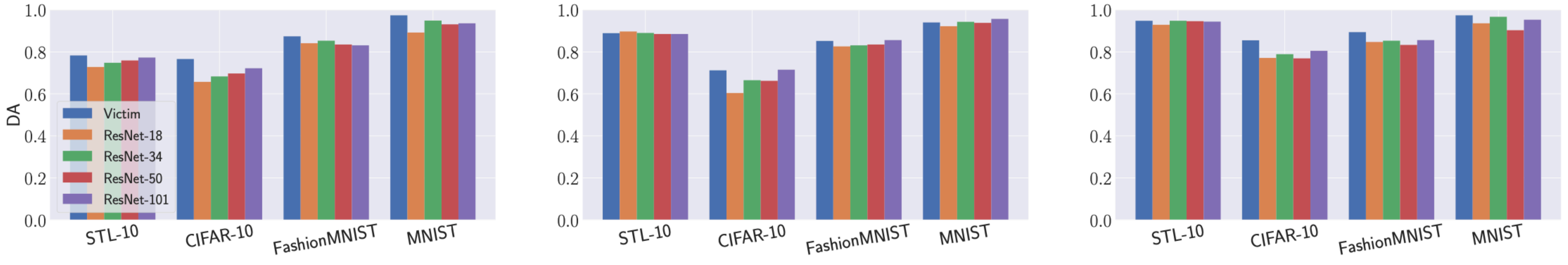
🤔 Improve utility & effectiveness of the watermarked encoder

Utility of the victim encoder

Table 2: Clean downstream accuracy (CDA).

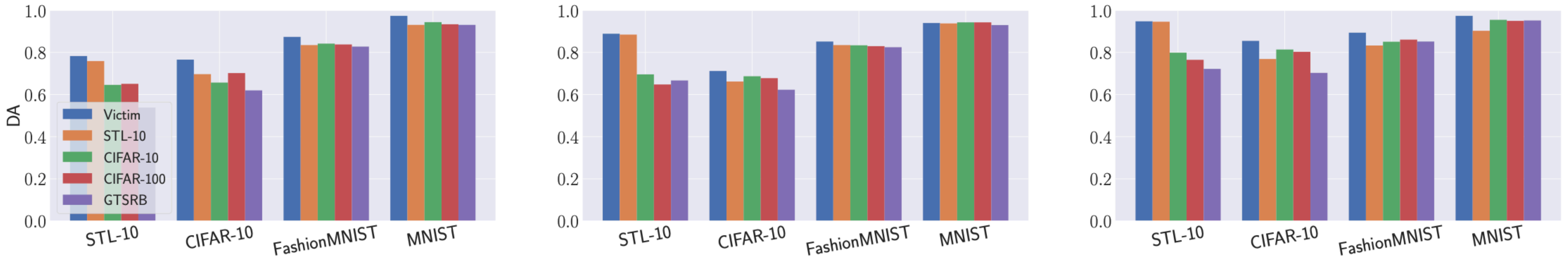
Downstream Task	SimCLR	MoCo v2	BYOL
STL-10	0.783	0.889	0.948
CIFAR-10	0.766	0.712	0.855
MNIST	0.974	0.940	0.974
F-MNIST	0.874	0.852	0.894

Model Stealing Attacks



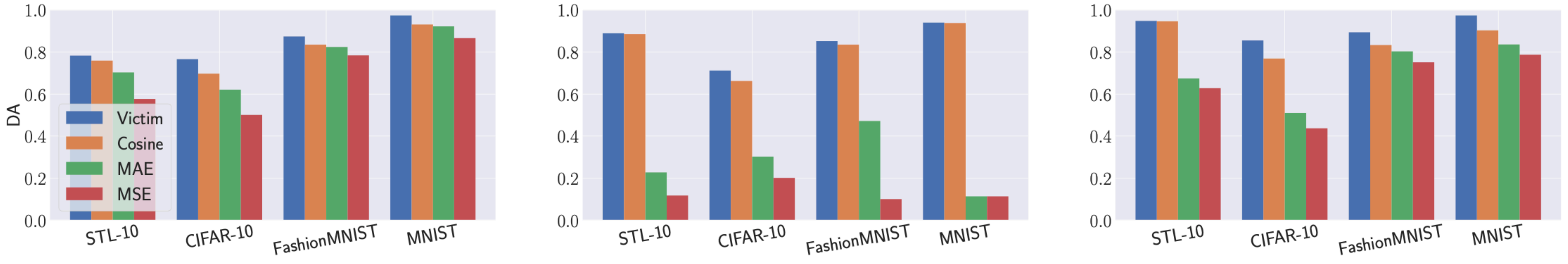
(a) SimCLR (b) MoCo v2 (c) BYOL

Figure 4: The performance of surrogate encoders trained with different architectures.



(a) SimCLR (b) MoCo v2 (c) BYOL

Figure 5: The performance of surrogate encoders trained with different query datasets.



(a) SimCLR (b) MoCo v2 (c) BYOL

Figure 6: The performance of surrogate encoders trained with different loss functions.

Table 3: Monetary Cost ($). Here Res denotes ResNet.

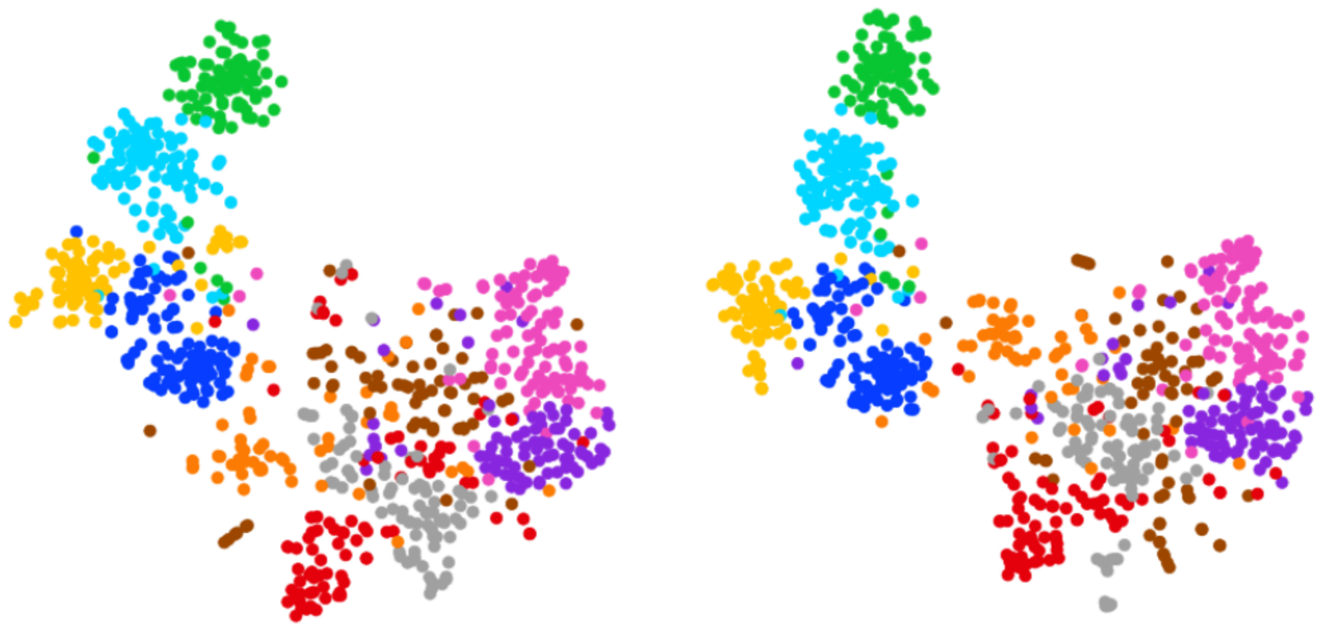
	Pre-training	Stealing			
		Res-18	Res-34	Res-50	Res-101
SimCLR	**1,920.00**	58.24	61.10	66.67	74.50
MoCo v2	**4,206.08**	58.13	61.09	66.55	74.37
BYOL	**5,713.92**	58.16	60.84	64.28	72.49

Performance of SSLGurad

- Fidelity: To minimize the impact of SSLGuard on the legitimate users

Table 5: Fidelity (DA). The value in the parenthesis denotes the difference between CDA.

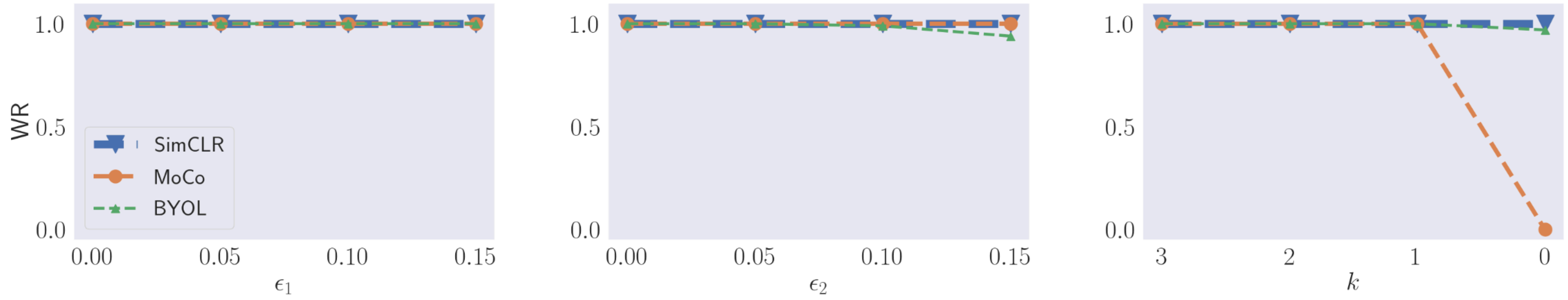
Task	F_*^{simclr}	F_*^{moco}	F_*^{byol}
STL-10	0.781 (**-0.002**)	0.888 (**-0.001**)	0.940 (**-0.008**)
CIFAR-10	0.765 (**-0.001**)	0.701 (**-0.011**)	0.857 (**+0.002**)
MNIST	0.965 (**-0.009**)	0.956 (**+0.016**)	0.966 (**+0.002**)
F-MNIST	0.878 (**+0.004**)	0.845 (**-0.007**)	0.894 (**+0.000**)



(a) F^{byol} **(b) F_*^{byol}**

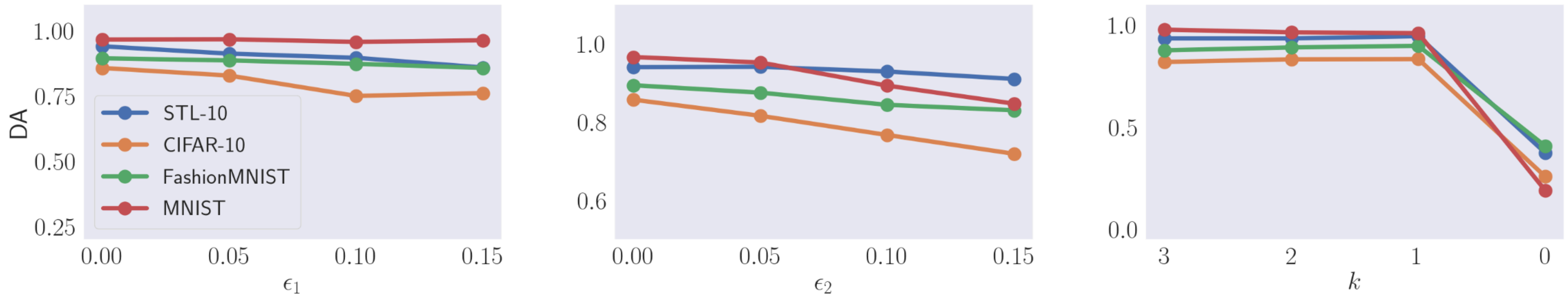
Figure 7: The t-SNE visualizations of features output from F^{byol} and F_*^{byol} when we input 800 samples in 10 classes randomly chosen from the STL-10 training dataset. Each point represents an embedding. Each color represents one class.

Robustness



(a) Input noising **(b) Output noising** **(c) Output truncation**

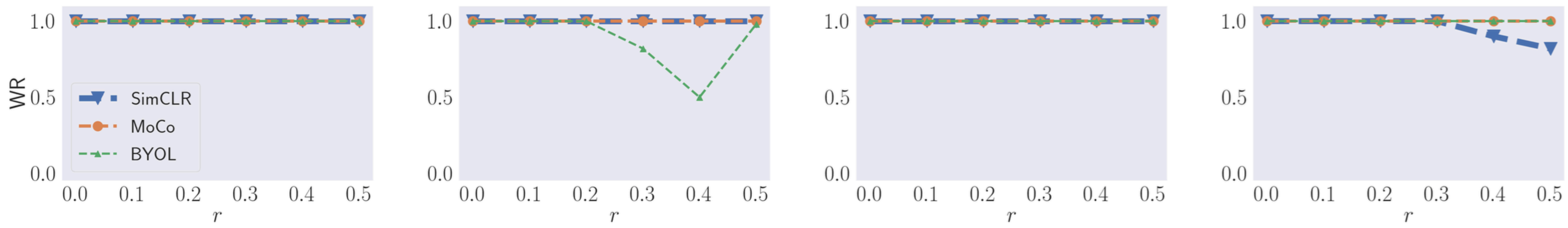
Figure 8: The WR on different watermark removal attacks.



(a) Input noising **(b) Output noising** **(c) Output truncation**

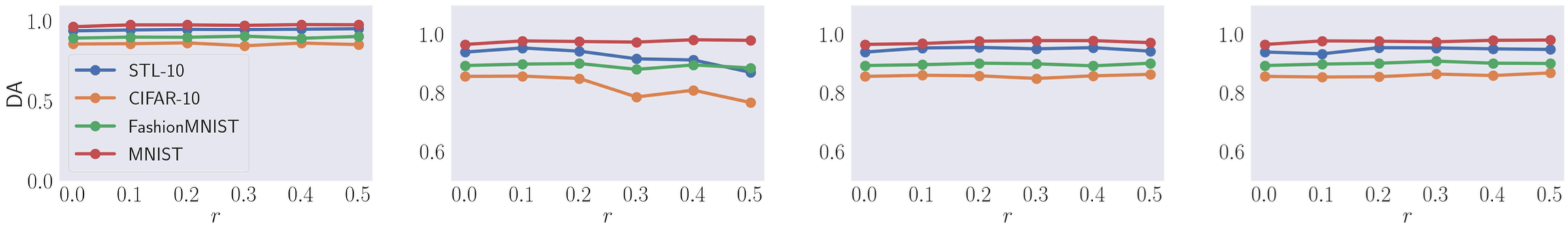
Figure 9: The DA on different watermark removal attacks. The victim encoder is BYOL.

Robustness



(a) Global pruning (b) Local pruning (c) Global fine-tuning (d) Local fine-tuning

Figure 10: The WR of pruned and fine-tuned encoders.



(a) Global pruning (b) Local pruning (c) Global fine-tuning (d) Local fine-tuning

Figure 11: The DA of pruned and fine-tuned encoders. The victim encoder is BYOL.

Robustness

Table 7: Overwriting.

		SimCLR	MoCo v2	BYOL
DA	STL-10	0.785	0.888	0.954
	CIFAR-10	0.765	0.685	0.863
	MNIST	0.962	0.955	0.977
	F-MNIST	0.885	0.837	0.905
WR	Overwriting key	1.00	1.00	0.98
	Original key	**1.00**	**1.00**	**1.00**

Table 9: The DA and WR of model stealing attacks against the watermarked encoders.

Attacks	Metric		SimCLR	MoCo	BYOL
`Steal-1`	DA	STL-10	0.721	0.890	0.938
		CIFAR-10	0.685	0.628	0.791
		F-MNIST	0.832	0.809	0.830
		MNIST	0.928	0.923	0.915
	WR		**1.00**	**0.96**	**1.00**
`Steal-2`	DA	STL-10	0.727	0.871	0.937
		CIFAR-10	0.677	0.628	0.815
		F-MNIST	0.840	0.827	0.865
		MNIST	0.935	0.919	0.961
	WR		**0.99**	**0.90**	**1.00**
`Steal-3`	DA	STL-10	0.732	0.874	0.923
		CIFAR-10	0.677	0.658	0.784
		F-MNIST	0.827	0.823	0.851
		MNIST	0.932	0.940	0.922
	WR		**1.00**	**0.95**	**0.98**

Conclusion

- We are the first to quantify the copyright breaching threats of SSL pre-trained encoders through the lens of model stealing attacks.
- To protect the copyright of the SSL pre-trained encoder, we propose SSLGuard, a robust black-box watermarking scheme.
- Extensive evaluations show that SSLGuard is effective and robust against several watermark removal attacks.

Thank you!

congtianshuo@gmail.com