A. Supplementary Material

A.1. Training Algorithm of Cont-Steal

Algorithm 1: The training process of Cont-Steal.

input : Surrogate training dataset $D_{surrogate}^{train}$, target encoder E_t, surrogate encoder E_s

Initialize E_s's parameters;
for *each epoch* **do**
  for *each batch* **do**
    Sample a batch with N training data samples $x_1, x_2, \cdots, x_N$ from $D_{surrogate}^{train}$
    Generate augmented data samples: $(\widetilde{x}_{1,t}, \widetilde{x}_{1,s}), (\widetilde{x}_{2,t}, \widetilde{x}_{2,s}), \cdots, (\widetilde{x}_{N,t}, \widetilde{x}_{N,s})$, where $\widetilde{x}_{k,t}$ and $\widetilde{x}_{k,s}$ are the two augmented views of x_k
    Feed $\widetilde{x}_{k,t}$ to E_t and $\widetilde{x}_k$ to E_s to calculate the contrastive steal loss: $L_{Cont-Steal} = \frac{\sum_{k=1}^{N} l(k)}{N}$
    Optimize E_s's parameters with the contrastive steal loss $L_{Cont-Steal}$
  end
end
return *Surrogate encoder E_s*

Algorithm 1 presents the training process of contrastive stealing. In each batch, given N training samples, we first generate $2N$ augmented views and feed the target encoder and surrogate encoder with different views generated by the same samples. Then, we optimize the surrogate encoder by minimizing $L_{Cont-Steal}$.

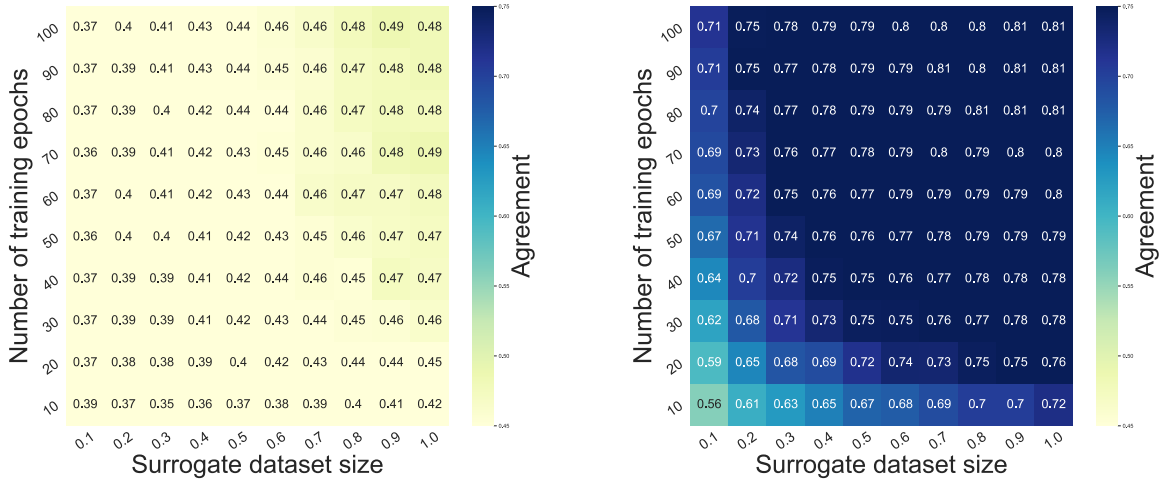
A.2. Ablation Studies on Adversary Training Process

Impact of Surrogate Encoder's Architecture. Previous experiments are based on the assumption that the adversary knows the target encoder's architecture. We then investigate whether the attack against the encoder is still effective when the surrogate encoder has different model architectures compared to the target encoder. Concretely, we perform Cont-Steal against the ResNet18 encoder with the surrogate encoder's architecture as ResNet18, ResNet34, ResNet50, DenseNet161, and MobileNetV2, respectively. As shown in Table 3, we can see that the architecture of the surrogate model only has limited influence on the attack performance. For instance, the adversary can achieve 0.839 accuracy using the same architecture as the target model, while it can even achieve 0.840 accuracy when using a more complex model architecture (ResNet50) on SimCLR. The attack performance will drop a little if the adversary uses DenseNet161 and MobileNetV2. This might be because the architectures of DenseNet161 and MobileNetV2 have larger differences compared to ResNet18. However, the accuracy with DenseNet161/MobileNetV2 as the surrogate encoder's architecture can still achieve 0.828/0.811. This demonstrates that the model architectures of the surrogate encoder only have a limited impact on the attack performance, which makes the attack a more realistic threat.

Table 3. Cont-Steal attack performance of different surrogate architectures. Target encoders (ResNet18) and downstream classifiers are trained on CIFAR10. The surrogate dataset is also CIFAR10.

Framework	Architectures	Agreement	Accuracy
SimCLR	ResNet18	0.835	0.839
	ResNet34	0.837	0.842
	ResNet50	0.844	0.840
	DenseNet161	0.831	0.828
	MobileNetV2	0.815	0.811
MoCo	ResNet18	0.857	0.849
	ResNet34	0.858	0.849
	ResNet50	0.867	0.856
	DenseNet161	0.813	0.811
	MobileNetV2	0.796	0.801
BYOL	ResNet18	0.845	0.842
	ResNet34	0.850	0.847
	ResNet50	0.857	0.855
	DenseNet161	0.845	0.821
	MobileNetV2	0.839	0.847
SimSiam	ResNet18	0.856	0.835
	ResNet34	0.858	0.839
	ResNet50	0.860	0.848
	DenseNet161	0.791	0.783
	MobileNetV2	0.812	0.832

Impact of Surrogate Dataset's Size and Surrogate Model's Training Epoch. We conduct ablation studies here to better illustrate the effectiveness of Cont-Steal. Concretely, we investigate whether conventional attacks and Cont-Steal are still effective under limited surrogate dataset size and the number of training epochs. Ideally, we consider the attack that can reach similar performance but with less surrogate dataset size and fewer training epochs as a better attack as it requires less query and monetary costs. As shown in Figure 9, we observe that both conventional attacks and Cont-Steal can have better performance with a larger surrogate dataset size and more training epochs. For instance, Cont-Steal reaches 0.675 agreement when the surrogate encoder is trained with 10% surrogate dataset for 50 epochs, while the agreement increase to 0.812 with 100% surrogate dataset and 100 training epochs. The second observation is that Cont-Steal outperforms conventional attacks even with limited data and training epochs. For instance, even with only 10% surrogate dataset and 10 training epochs, the surrogate encoder built by Cont-Steal can reach 0.562 agreement, while the conventional attack can only achieve 0.479 agreement with the full surrogate dataset and 100 training epochs. As we mentioned before, this is because Cont-Steal can enforce the surrogate embedding of an image close to its target embedding and also push away



(a) Conventional (b) Cont-Steal

Figure 9. Heatmap of the agreement scores of model stealing attacks. The target model's encoder and downstream classifier are both ResNet18 trained by SimCLR on CIFAR10. The surrogate dataset is STL10. Surrogate dataset's size refers to the proportion of surrogate data we used for the whole surrogate dataset. We show the performance of 100 combinations of different training epochs and the surrogate gate dataset's size.

embeddings of different images irrespective of being generated by the target or the surrogate encoders (see also Table 5 for the necessity of introducing negative pairs from the surrogate encoder). This makes Cont-Steal a more effective model stealing attack against encoders.

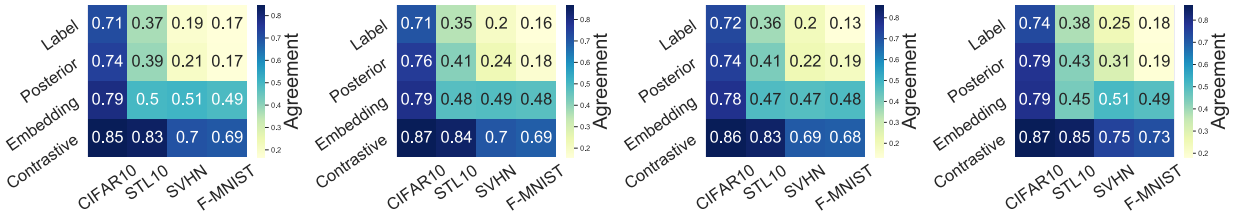
Impact of Surrogate Dataset's Correlation With the Target Dataset. In the meanwhile, since the adversary cannot always have knowledge about the target dataset, the impact of the surrogate dataset's correlation with the target dataset is also worth consideration. We find that Cont-Steal depends less on the surrogate dataset's distribution and can always achieve stable performance. We plot the attack agreement in Figure 10 where the target encoders and downstream classifiers are trained on CIFAR10. We can see that when the adversary conducts a conventional attack against the classifier, the adversary's knowledge of target training data is crucial. For example, when the adversary can only get the predicted label from the target model, he/she can only achieve 0.182 agreement when using F-MNIST to attack the model trained by SimCLR, while it can achieve 0.711 agreement when using CIFAR10 as the surrogate dataset, which is same as target dataset. However, compared to the predicted label or posterior as the response, embedding depends less on the surrogate dataset distribution, and Cont-Steal can better leverage the embedding information, contributing to the less dependent on the surrogate dataset's distribution. For instance, when the target model is trained by SimCLR, Cont-Steal can achieve 0.832 agreement when the surrogate dataset is STL10, which is even better than the best conventional attack (0.781) using the exact same target training dataset as the surrogate dataset and embedding as the response. Such observation better implies that Cont-Steal can always achieve good performance regardless of the surrogate dataset's distribution and can also achieve more generalized performance in practice.

Table 4. Impact of learning rate and batch size. The target dataset and downstream dataset are both CIFAR10. The surrogate dataset is STL10. Note that for different learning rates, we set the batch size as 128. For different batch sizes, we set the learning rate as 0.001

Hyperparamter	Different Settings	Agreement
Learning Rate	0.001	0.813
	0.002	0.801
	0.003	0.805
	0.004	0.819
	0.005	0.809
Batch Size	16	0.827
	32	0.806
	64	0.800
	128	0.813
	256	0.775

Impact of Hyperparameters. In our experiments, we set batch size as 128 and learning rate as 0.001. We show in Table 4 that with reasonable batch size and learning rate, our Cont-Steal can have stable performance.

Impact of Negative Pairs Generated From the Surrogate Encoder. In Cont-Steal's loss functions, besides $D^{-}_{encoder}$, we also consider the distance of negative pairs generated



(a) SimCLR (b) MoCo (c) BYOL (d) SimSiam

Figure 10. Heatmap of the agreement scores of model stealing attacks. We show the performance of 16 combinations of different information that the target model outputs and the adversary's knowledge of target training data. Target models are trained on CIFAR10.

from the surrogate encoder itself, i.e., D^-_{self}. To evaluate the necessity of D^-_{self}, we take the target encoder trained by BYOL on CIFAR10 and the downstream task on STL10 as an example and study the attack performance with and without D^-_{self}. The results are summarized in Table 5. We find that adding D^-_{self} greatly improves the attack performance in both accuracy and agreement. For instance, when the surrogate dataset is STL10, the surrogate model stolen by Cont-Steal with D^-_{self} achieves 0.817 agreement while only 0.314 if without D^-_{self}. The reason behind this is that the negative pairs generated from the surrogate encoder can serve as extra "anchors" to better locate the position of the embedding, which leads to higher agreement. Such observation demonstrates that it is important to introduce D^-_{self} in Cont-Steal as well.

Table 5. The agreement and accuracy of different contrastive losses. We use BYOL trained on STL10 as the target model.

Dataset	Method	BYOL	
		Agreement	Accuracy
CIFAR10	w/o $D^-_{encoder}$	0.242	0.242
	w $D^-_{encoder}$	0.844	0.843
F-MNIST	w/o $D^-_{encoder}$	0.215	0.217
	w $D^-_{encoder}$	0.647	0.641
STL10	w/o $D^-_{encoder}$	0.314	0.320
	w $D^-_{encoder}$	0.817	0.811
SVHN	w/o $D^-_{encoder}$	0.176	0.175
	w $D^-_{encoder}$	0.655	0.650

A.3. Further Attacks Based on Cont-Steal

As we have mentioned in the introduction part, model stealing can be used as a stepping stone for further attacks. In this section, we select adversary sample attacks as a case study to show the importance of model stealing for further attacks on the target model. Normally, the adversary can not obtain the gradient from the target model. But to conduct adversary sample attacks, the adversary needs to obtain the gradient in most attack scenarios. Therefore, the adversary can construct a surrogate model to generate the adversary sample and transfer it to the target model to perform the attack. We consider three widely used mechanisms to generate adversarial examples, including Fast Gradient Sign Attack (FGSM) [16], Basic Iterative Methods (BIM) [29], and Projected Gradient Descent (PGD) [32]. Our target model is SimCLR pre-trained on CIFAR10 and the last layer classifier trained on STL10. We also use STL10 as the surrogate dataset to conduct Cont-Steal and generate adversary samples. Experiments show that the surrogate model can generate adversary samples that are valid for the target model (Table 6). To show the necessity of the surrogate model as a springboard for the attack, we also conduct the baseline attack, which uses another model as the springboard to attack the target model. We choose the normal ResNet18 model trained on SVHN as our baseline model and then apply the adversary example to attack the target model. We observe that compared to the adversarial examples generated from the baseline model, those adversarial examples generated from the surrogate model constructed by Cont-Steal can better transfer to the target model. For instance, with PGD, the adversarial examples obtained from the surrogate model can lead to a lower classification accuracy (0.203) on the target model than those generated from the baseline model (0.246). This implies that the model stealing attack can be a valid stepping stone for more effective further attacks.

Table 6. The different methods to create adversary sample to attack on surrogate model and target model. [Lower is better]

Method	Surrogate model (acc)	Target model (acc)	Baseline (acc)
FGSM [16]	0.097	0.131	0.194
BIM [29]	0.054	0.192	0.235
PGD [32]	0.092	0.203	0.246

A.4. More Results on Conventional Attacks

Figure 11, Figure 12, and Figure 13 show the results of the conventional attacks on target models whose encoders are pre-trained on CIFAR10 and downstream classifiers are trained on STL10, F-MNIST, and SVHN, respectively. Figure 14, Figure 15, and Figure 16 show the results

of the conventional attacks on classifiers whose encoders are pre-trained on ImageNet100 and downstream classifiers are trained on STL10, F-MNIST, and SVHN, respectively.

A.5. More Results on Cont-Steal

Figure 17, Figure 18, and Figure 19 show the results of the Cont-Steal on target models whose encoders are pre-trained on CIFAR10 and downstream classifiers are trained on STL10, F-MNIST, and SVHN, respectively. Figure 20, Figure 21, Figure 22 show the results of the Cont-Steal on target models whose encoders are pre-trained on ImageNet100 and downstream classifiers are trained on CIFAR10, STL10, and SVHN, respectively.

A.6. Attacks Performance on Other Visual Models

Apart from four contrastive models we tried in the paper, we also conduct our Cont-Steal on other large, state-of-the-art models such as ViT and CLIP. We show that Cont-Steal can perform very well on ViT, MAE, and the image encoder of CLIP in Table 7. The results demonstrate the scalability of Cont-Steal.

A.7. Compare With Other Existing Works

Note that we are the first work to systematically propose model stealing attacks against image encoders. There are also some parallel and follow-up works on this domain proposed after our work. Here, we compare our works with other existing methods. The main difference between our work and recent works is our designed contrastive steal loss and the usage of data augmentation. Compared to StolenEncoder [31], our loss focuses on the comparison of positive and negative samples, while StolenEncoder focuses on the combination of augmentation and non-augmentation loss. The main difference between our work and the methods listed in [14] is that 1) we leverage data augmentation as part of the methods. 2) we design the loss function ourselves to consider more negative examples compared to the INFONCE loss. We show in Table 9 that our method works better. Note that KL divergence is also a loss function used by knowledge distillation. As knowledge distillation is a similar task to model stealing, we also report the results of KL divergence.

Table 7. The performance of Cont-Steal and conventional attacks against state-of-the-art models. Note that all of our target encoders are pre-trained encoders available online and downstream classifiers are trained on CIFAR10.

Surrogate Dataset	Metric	Attacks	ViT	MAE	CLIP
Original performance	Accuracy	NaN	0.896	0.900	0.903
CIFAR10	Agreement	Conventional	0.745	0.555	0.815
	Agreement	Cont-Steal	**0.967**	**0.712**	**0.889**
STL10	Agreement	Conventional	0.553	0.451	0.550
	Agreement	Cont-Steal	**0.942**	**0.624**	**0.905**
SVHN	Agreement	Conventional	0.587	0.419	0.578
	Agreement	Cont-Steal	**0.944**	**0.548**	**0.893**
F-MNIST	Agreement	Conventional	0.602	0.395	0.465
	Agreement	Cont-Steal	**0.696**	**0.501**	**0.598**

Table 8. Dataset inference performance on Cont-Steal.

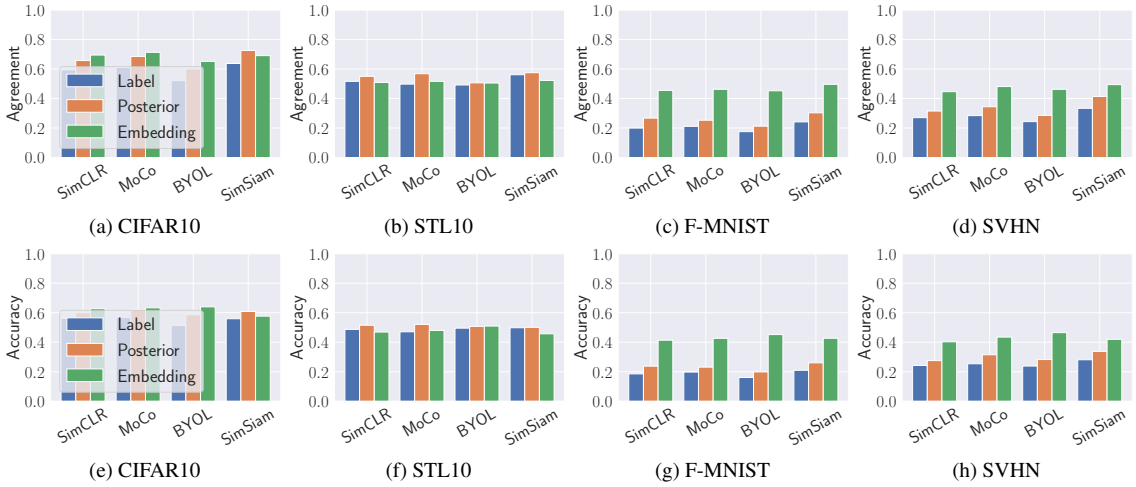
Model	Dataset	$S(\cdot, E_T)$	$C(\cdot, E_T)$
Target Encoder	CIFAR10	1.000	1.000
Surrogate Encoder	SVHN	0.412	0.393
Surrogate Encoder (fine-tuning)	SVHN	0.17	0.00
Independent Encoder	SVHN	0.11	0.00

Table 9. The comparison of Cont-Steal and other existing works. Both the target encoder and downstream classifier are trained on CIFAR10. Note that our results are different from the original paper of [14] because we test the surrogate encoder on the original task.

	CIFAR10		STL10	
	Agreement	Accuracy	Agreement	Accuracy
Baseline	0.785	0.790	0.499	0.500
StolenEncoder	0.811	0.808	0.766	0.767
KL Divergence	0.213	0.203	0.178	0.162
INFONCE	0.826	0.828	0.806	0.797
Cont-Steal	**0.845**	**0.854**	**0.829**	**0.828**

A.8. More Defenses.

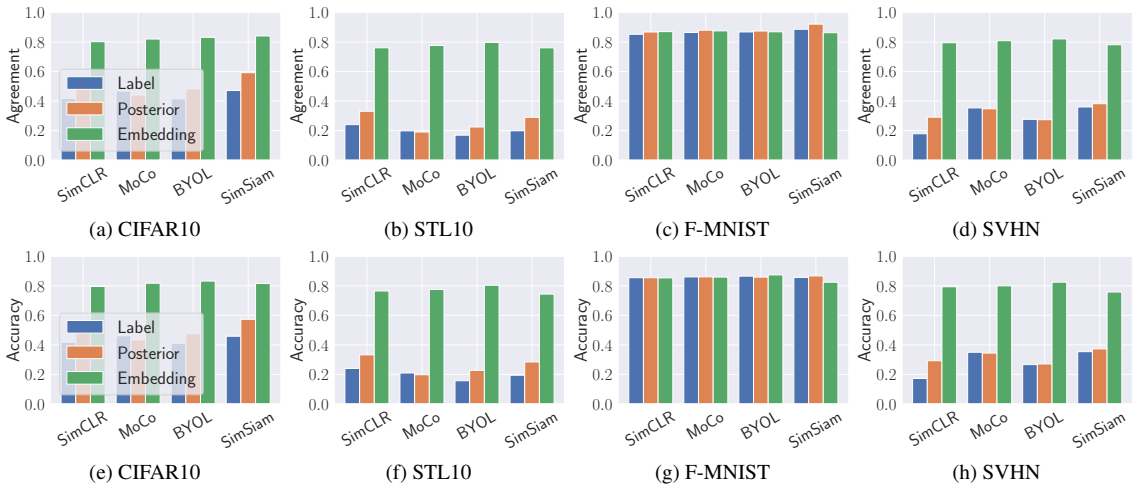
We implement the dataset inference defense in [15] (see Table 8). $S(\cdot, E_T)$/$C(\cdot, E_T)$ represents the mutual information/cosine similarity between the given model and the target model (the higher, the more similar). Note that the surrogate encoder will be fine-tuned for downstream tasks. We find the fine-tuning process [18, 55] will disable the defense. Normally, the open-source encoders are trained on very large public datasets instead of limited private datasets, which makes the defense less practical.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

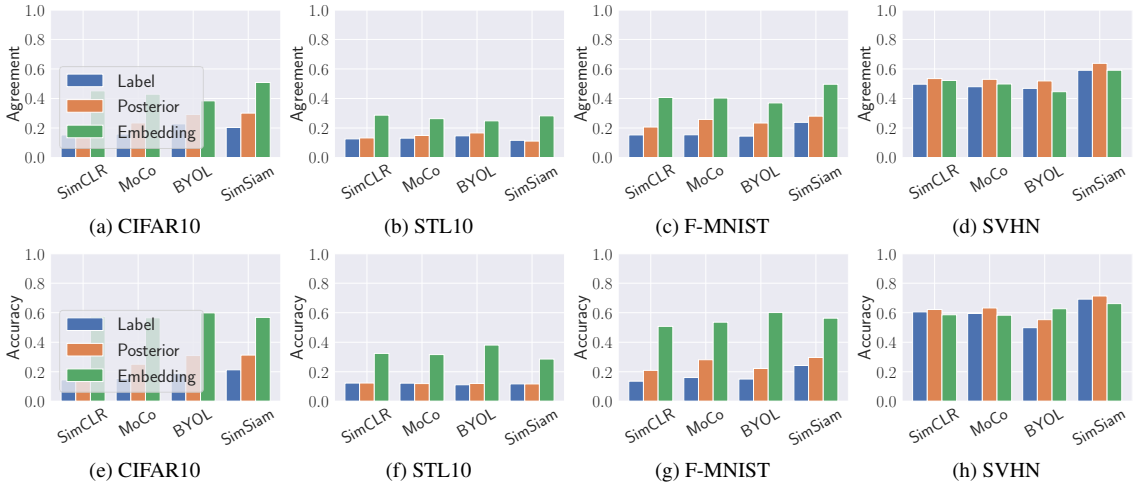
Figure 11. The performance of model stealing attack against target encodes and downstream classifiers trained on CIFAR10 and STL10. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

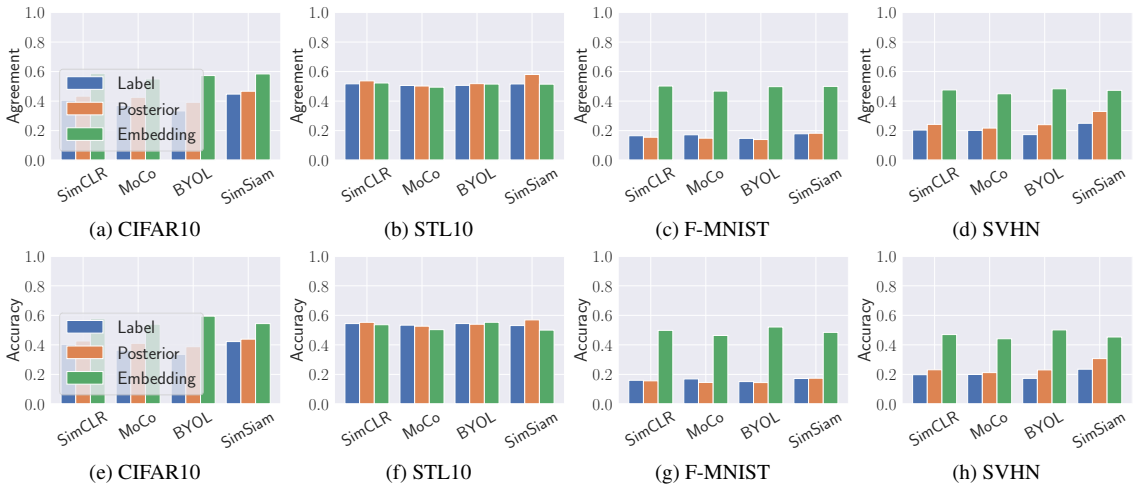
Figure 12. The performance of model stealing attack against target encodes and downstream classifiers trained on CIFAR10 and Fashon-MNIST. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

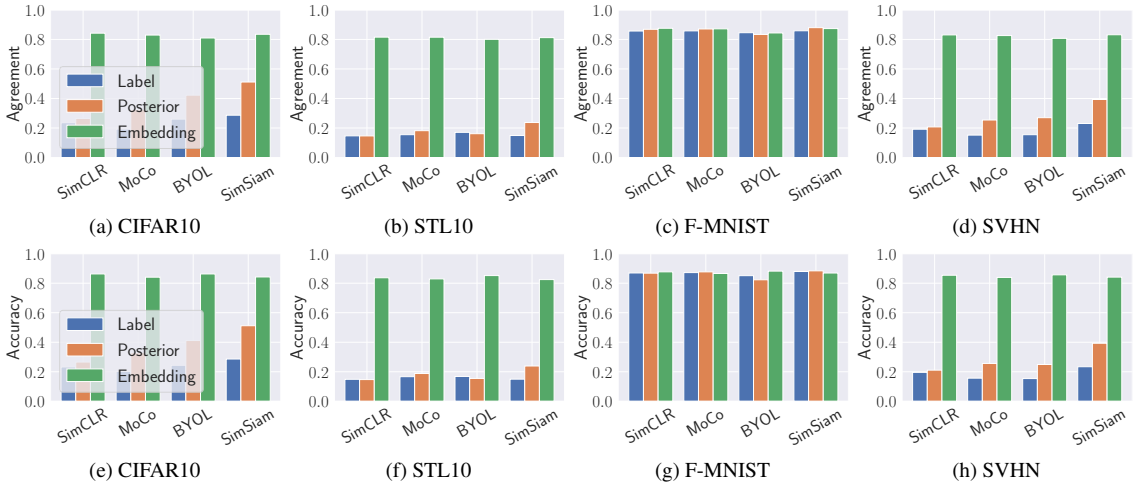
Figure 13. The performance of model stealing attack against target encodes and downstream classifiers trained on CIFAR10 and SVHN. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

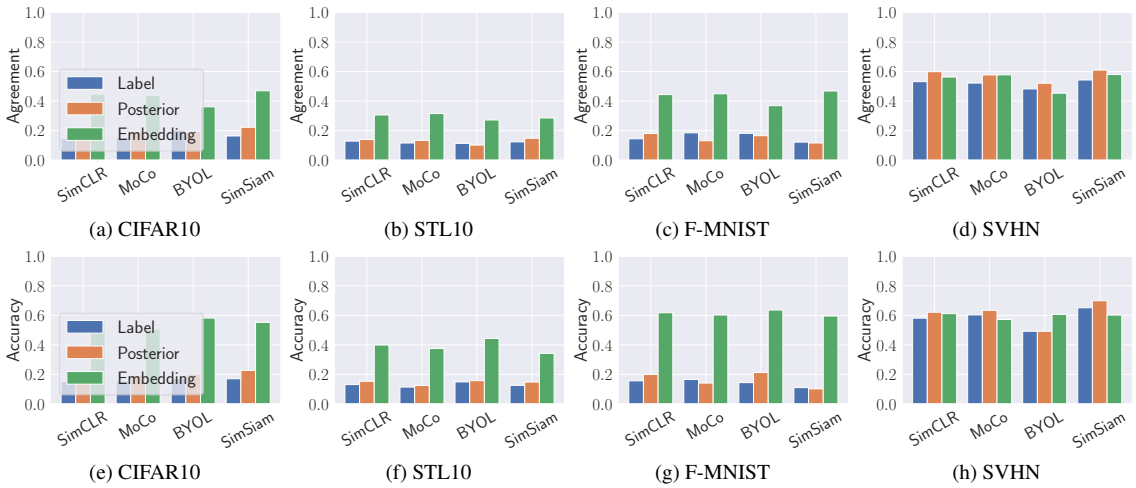
Figure 14. The performance of model stealing attack against target encodes and downstream classifiers trained on ImageNet and STL10. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

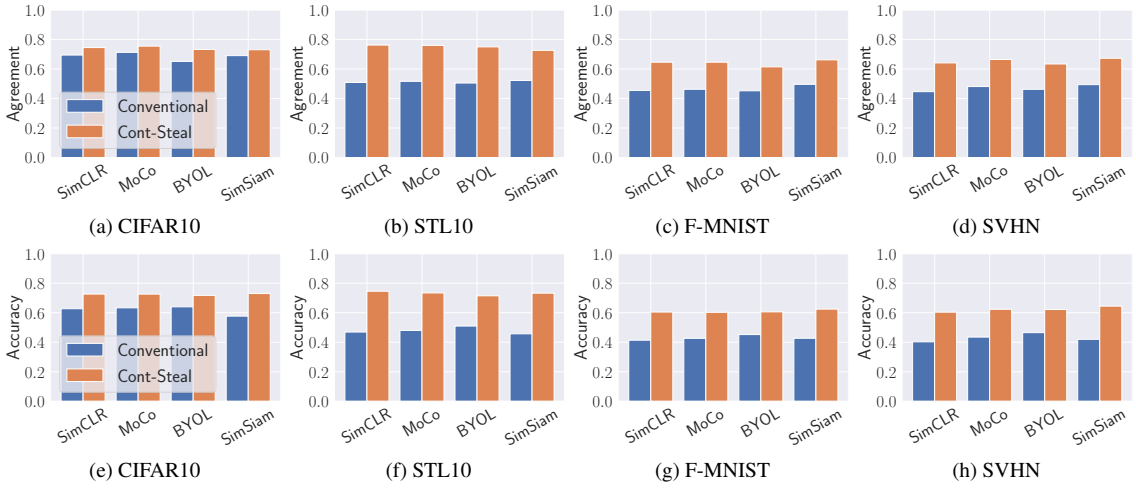
Figure 15. The performance of model stealing attack against target encodes and downstream classifiers trained on ImageNet and Fashion-MNIST. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

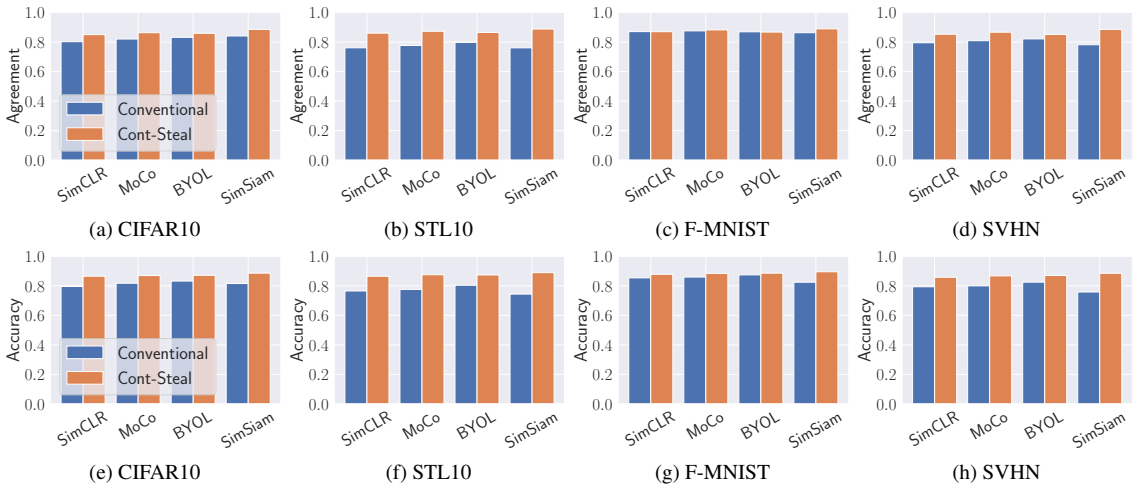
Figure 16. The performance of model stealing attack against target encodes and downstream classifiers trained on ImageNet and SVHN. Target models can output predicted labels, posteriors, or embeddings. The adversary uses CIFAR10, STL10, Fashion-MNIST (F-MNIST), SVHN to conduct model stealing attacks. The x-axis represents different kinds of target models. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

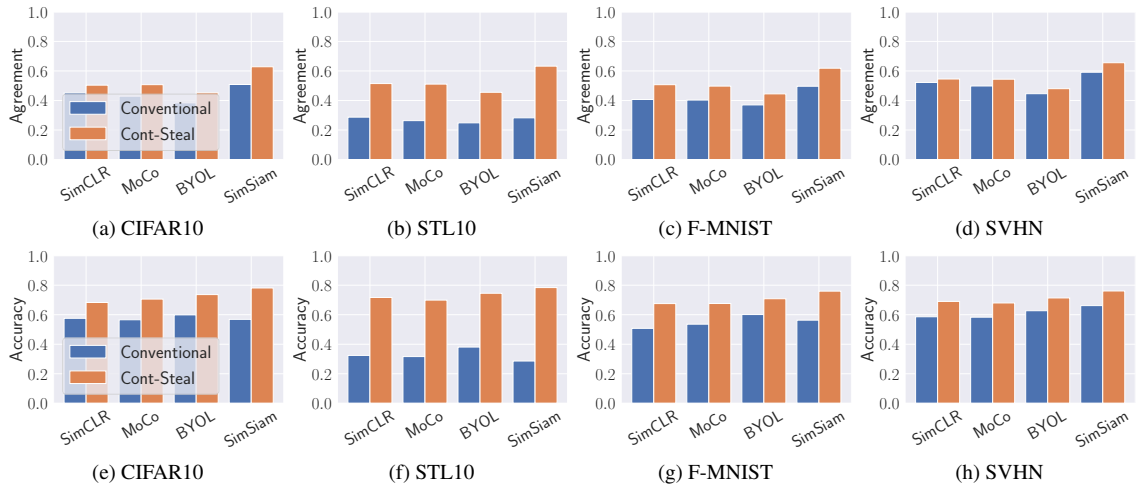
Figure 17. The performance of Cont-Steal and conventional attack against target encoders trained on CIFAR10. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses STL10 as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

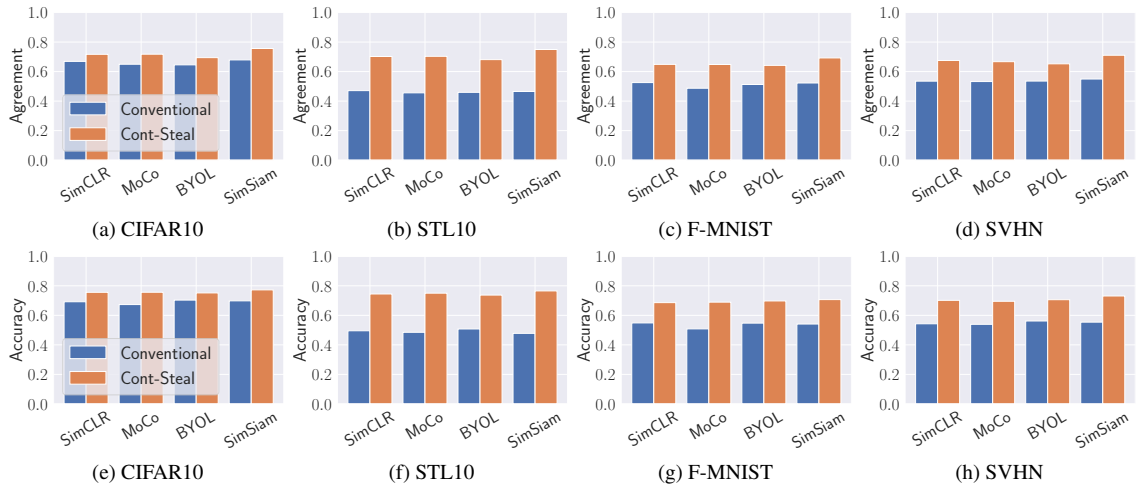
Figure 18. The performance of Cont-Steal and conventional attack against target encoders trained on CIFAR10. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses F-MNIST as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

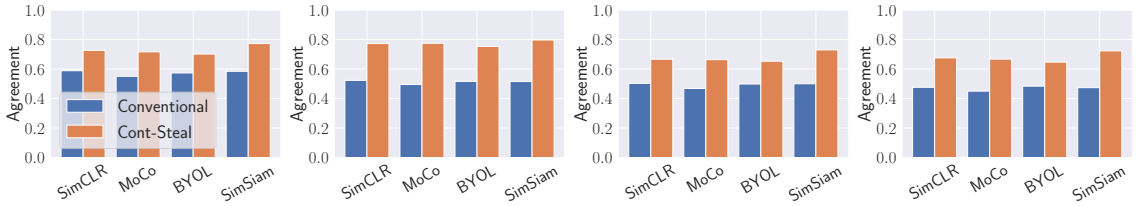
Figure 19. The performance of Cont-Steal and conventional attack against target encoders trained on CIFAR10. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses SVHN as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



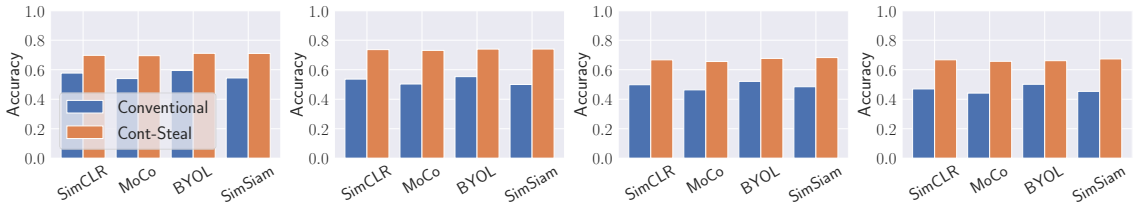
(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

Figure 20. The performance of Cont-Steal and conventional attack against target encoders trained on ImageNet100. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses CIFAR10 as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.

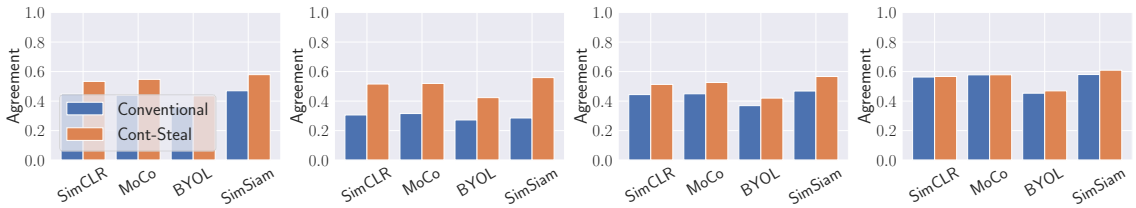


(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN

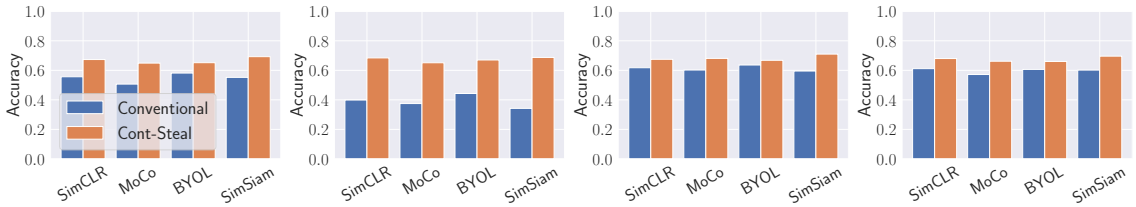


(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

Figure 21. The performance of Cont-Steal and conventional attack against target encoders trained on ImageNet100. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses F-MNIST as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.



(a) CIFAR10 (b) STL10 (c) F-MNIST (d) SVHN



(e) CIFAR10 (f) STL10 (g) F-MNIST (h) SVHN

Figure 22. The performance of Cont-Steal and conventional attack against target encoders trained on ImagNet100. The adversary uses CIFAR10, STL10, F-MNIST, and SVHN to conduct model stealing attacks. The adversary uses SVHN as the downstream task to evaluate the attack performance. The x-axis represents different kinds of the target model. The first line's y-axis represents the agreement of the model stealing attack. The second line's y-axis represents the accuracy of the model stealing attack.