On the Difficulty of Defending Self-Supervised Learning against Model Extraction

Adam Dziedzic, Nikita Dhawan, Muhammad Ahmad Kaleem, Jonas Guan, Nicolas Papernot

International Conference on Machine Learning (ICML)
July 17th - 23rd, 2022
Supervised Learning API

Unlabeled Data \rightarrow \text{Query} \rightarrow \text{Supervised API}

\text{Query} \leftarrow \text{Low-dimensional outputs (e.g., labels)}
Supervised vs Self-Supervised Learning APIs

Unlabeled Data → Query → Supervised API

Low-dimensional outputs (e.g., labels) → Model

Unlabeled Data → Query → Self-Supervised Learning (SSL) API

Low-dimensional outputs (e.g., labels) → Predictor

High-dimensional representations → Encoder
High Cost of Creating Self-Supervised APIs

Collect Data

Tune Hyper-parameters

Run on GPU/TPU/CPU

Low-dimensional outputs (e.g., labels)

Query

SSL API

High-dimensional representations
High Cost of Creating Self-Supervised APIs

Collect Data
Tune Hyper-parameters
Run on GPU/TPU/CPU

Unlabeled Data

Query
SSL API

Low-dimensional outputs (e.g., labels)

High-dimensional representations

$12M
GPT-3
Efficient Attacks & Inadequate Defenses

Low-dimensional outputs (e.g., labels) → Unlabeled Data → Query → SSL API

Unlabeled Data

Query

SSL API

Encoder

High-dimensional representations
Efficient Attacks & Inadequate Defenses

1. Attacks against SSL models are query efficient.
2. Existing defenses against stealing supervised models are inadequate for SSL models.
Framework for Stealing Encoders

Input image

View

Representation

Maximize Agreement

\( \mathcal{L}(y, y') \)

Stolen

Victim

\( x \)

\( t \)

\( t' \)

\( v \)

\( v' \)

\( f' \)

\( f \)
# Impact of Loss Functions on Encoder Stealing

<table>
<thead>
<tr>
<th>Loss\Downstream Task</th>
<th>STL10</th>
<th>CIFAR10</th>
<th>STL10</th>
<th>CIFAR10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim baseline</td>
<td>67.9</td>
<td>79.0</td>
<td>50.6</td>
<td>57.5</td>
</tr>
<tr>
<td>MSE</td>
<td>64.8</td>
<td>75.5</td>
<td>46.3</td>
<td>51.2</td>
</tr>
<tr>
<td>InfoNCE</td>
<td>64.6</td>
<td>75.5</td>
<td>50.4</td>
<td>56.3</td>
</tr>
<tr>
<td>SoftNN</td>
<td><strong>67.1</strong></td>
<td>76.9</td>
<td>44.6</td>
<td>48.4</td>
</tr>
<tr>
<td>SupCon (uses labels)</td>
<td>63.1</td>
<td><strong>78.5</strong></td>
<td>33.9</td>
<td>42.3</td>
</tr>
<tr>
<td>Wasserstein</td>
<td>50.8</td>
<td>63.9</td>
<td>40.1</td>
<td>46.4</td>
</tr>
<tr>
<td>Barlow</td>
<td>26.6</td>
<td>26.9</td>
<td>16.3</td>
<td>17.9</td>
</tr>
</tbody>
</table>
## Impact of Loss Functions on Encoder Stealing

<table>
<thead>
<tr>
<th>Loss\Downstream Task</th>
<th>STLOnly</th>
<th>CIFAR10</th>
<th>STLOnly</th>
<th>CIFAR10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Victim baseline</strong></td>
<td>67.9</td>
<td>79.0</td>
<td>50.6</td>
<td>57.5</td>
</tr>
<tr>
<td>MSE</td>
<td>64.8</td>
<td>75.5</td>
<td>46.3</td>
<td>51.2</td>
</tr>
<tr>
<td>InfoNCE</td>
<td>64.6</td>
<td>75.5</td>
<td>50.4</td>
<td>56.3</td>
</tr>
<tr>
<td>SoftNN</td>
<td><strong>67.1</strong></td>
<td>76.9</td>
<td>44.6</td>
<td>48.4</td>
</tr>
<tr>
<td>SupCon (uses labels)</td>
<td>63.1</td>
<td><strong>78.5</strong></td>
<td>33.9</td>
<td>42.3</td>
</tr>
<tr>
<td>Wasserstein</td>
<td>50.8</td>
<td>63.9</td>
<td>40.1</td>
<td>46.4</td>
</tr>
<tr>
<td>Barlow</td>
<td>26.6</td>
<td>26.9</td>
<td>16.3</td>
<td>17.9</td>
</tr>
</tbody>
</table>

Contrastive losses perform the best for stealing encoders.
# Queries | Data for Stealing | CIFAR10 | CIFAR100 | STL10 | SVHN | F-MNIST
--- | --- | --- | --- | --- | --- | ---
**Victim ImageNet Encoder Baseline** |  | 90.33 | 71.45 | 94.9 | 79.39 | 91.9
60K | CIFAR10 | 83.3 | 57.0 | 71.2 | 73.8 | 90.7
50K | SVHN | 73.3 | 47.1 | 58.2 | 78.8 | 90.4
250K | SVHN | 77.1 | 52.6 | 61.9 | **80.2** | **91.4**
50K | ImageNet | 65.2 | 35.1 | 64.9 | 62.1 | 88.5
250K | ImageNet | 80.0 | **57.0** | **85.8** | 71.5 | 90.2

Stealing a Pre-trained ImageNet Encoder
Stealing a Pre-trained ImageNet Encoder

<table>
<thead>
<tr>
<th># Queries</th>
<th>Data for Stealing</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>STL10</th>
<th>SVHN</th>
<th>F-MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Victim ImageNet Encoder Baseline</strong></td>
<td></td>
<td>90.33</td>
<td>71.45</td>
<td>94.9</td>
<td>79.39</td>
<td>91.9</td>
</tr>
<tr>
<td>60K</td>
<td>CIFAR10</td>
<td><strong>83.3</strong></td>
<td><strong>57.0</strong></td>
<td>71.2</td>
<td>73.8</td>
<td><strong>90.7</strong></td>
</tr>
<tr>
<td>50K</td>
<td>SVHN</td>
<td>73.3</td>
<td>47.1</td>
<td>58.2</td>
<td>78.8</td>
<td><strong>90.4</strong></td>
</tr>
<tr>
<td>250K</td>
<td>SVHN</td>
<td>77.1</td>
<td>52.6</td>
<td>61.9</td>
<td><strong>80.2</strong></td>
<td><strong>91.4</strong></td>
</tr>
<tr>
<td>50K</td>
<td>ImageNet</td>
<td>65.2</td>
<td>35.1</td>
<td>64.9</td>
<td>62.1</td>
<td>88.5</td>
</tr>
<tr>
<td>250K</td>
<td>ImageNet</td>
<td>80.0</td>
<td><strong>57.0</strong></td>
<td><strong>85.8</strong></td>
<td>71.5</td>
<td><strong>90.2</strong></td>
</tr>
</tbody>
</table>

number of stealing queries < 1/5\(^{th}\) number of training data points
Adapt Defenses against Stealing Encoders

Active

\[ u = -\nabla_w L(\cdot, y) \]
\[ a = -\nabla_w L(\cdot, \hat{y}) \]

Poison Attacker’s Objective
Prediction Poisoning [Orekondy et al. 2020]

Pro-Active

Higher cost for more information
Callibrated PoW with PATE [Dziedzic et al. 2022]

Passive

Detect Attack & Stop Responding
PRADA [Juuti et al. 2019]

Reactive

Image

Copyright

Watermarked Image
Embed Rotation Task to Defend Encoders

Watermarked Encoder

Embedding

Rotation in Range: [0, 180] or [180, 360]
Transferability of the Rotation Watermark
Conclusions & Future Work

High Performance of Stolen Encoders

Contrastive Loss Functions

Design New Defenses
Thank you

🌐 https://cleverhans-lab.github.io
✉️ {adam.dziedzic,nicolas.papernot}@utoronto.ca