PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking

Chong Xiang†, Arjun Nitin Bhagoji‡, Vikash Sehwag†, Prateek Mittal†
†Princeton University ‡University of Chicago

USENIX Security Symposium 2021
PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking

Chong Xiang‡, Arjun Nitin Bhagoji‡, Vikash Sehwag‡, Prateek Mittal‡

‡Princeton University ‡University of Chicago

USENIX Security Symposium 2021
PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking

Chong Xiang†, Arjun Nitin Bhagoji‡, Vikash Sehwag†, Prateek Mittal†
†Princeton University ‡University of Chicago

USENIX Security Symposium 2021
Adversarial Example Attacks: Small Perturbations for Test-Time Model Misclassification

Normal Example \((x)\)
Dog \((y)\)

\[\begin{align*}
\text{Add imperceptible perturbations } \delta
\end{align*}\]

Adversarial Example \((x + \delta)\)
Cat \((y')\)

\[
\max_{\delta} L(M(x + \delta), y)
\]

\(L(\cdot)\) - Loss function; \(M(\cdot)\) - Model

A threat to ML models!
Challenge: Requires global perturbations
Our Focus: Localized Adversarial Patch Attacks

1. All perturbations within one local region (patch)
2. Patch pixels can take arbitrary values
3. Realizable in the physical world – print and attach the patch!
   • A REAL-WORLD threat

4. Patch can be anywhere on the image
5. Patch size should be reasonable (shouldn’t block the entire salient object)

Defense Objective: Provable Robustness on Certified Test Images

Test Image

Threat Model
(patch sizes, shapes, and location set)

Provable Analysis

Ground-truth $y$

"I think it is a dog"

"My prediction will never change"

or

"I can’t say anything for sure"

The prediction is **always** correct; Image certified!

Provable robust accuracy / certified accuracy: the fraction of test images that are

1. **Correctly** classified
2. **Provably robust** to any (adaptive) localized patch attack within the threat model
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

**PatchGuard: A Provably Robust Defense Framework**

- **Small Receptive Field**: Bound the number of corrupted features
- **Secure Feature Aggregation**: Do robust prediction on partially corrupted features
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

PatchGuard: A Provably Robust Defense Framework

**Small Receptive Field**
Bound the number of corrupted features

**Secure Feature Aggregation**
Do robust prediction on partially corrupted features

Tiger Cat (94.4%)
Receptive Field: a Region of the Input Image that an Extracted Feature is Looking at
Receptive Field: a Region of the Input Image that an Extracted Feature is Looking at
Receptive Field: a Region of the Input Image that an Extracted Feature is Looking at

Local feature map
Receptive Field: a Region of the Input Image that an Extracted Feature is Looking at
Receptive Field: a Region of the Input Image that an Extracted Feature is Looking at

Local feature map
Aggregate Local Features for Global Prediction

Local feature map

Global feature

Global prediction / logits

Dog!

cat
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 1: CNN with *large* receptive fields (e.g., ResNet with $483 \times 483$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 1: CNN with large receptive fields (e.g., ResNet with 483 × 483 px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 1: CNN with *large* receptive fields (e.g., ResNet with $483 \times 483$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 1: CNN with large receptive fields (e.g., ResNet with 483 × 483 px)

Note: all feature corrupted!
Little hope for us to do a robust prediction
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with 17 $\times$ 17 px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)
Key insight: the Receptive Field Size Determines the Number of Features Corrupted by the Adversarial Patch

Example 2: CNN with small receptive fields (e.g., BagNet with $17 \times 17$ px)

Note: only one feature corrupted!
A major step towards robust prediction!
Key insight: the Small Receptive Field Size Bounds the Number of Features Corrupted by the Adversarial Patch

Number of corrupted features $k$ (along one axis) satisfies:

$$ k = \frac{p + r - 1}{s} $$

$p$ patch size; $r$ receptive field size; $s$ receptive field stride

(more details are in the paper)

A smaller receptive field gives fewer corrupted features!
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

**PatchGuard: A Provably Robust Defense Framework**

- **Small Receptive Field**: Bound the number of corrupted features
- **Secure Feature Aggregation**: Do robust prediction on partially corrupted features
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

PatchGuard: A Provably Robust Defense Framework

- Small Receptive Field
  - Bound the number of corrupted features

- Secure Feature Aggregation
  - Do robust prediction on partially corrupted features
Vulnerability of Insecure Feature Aggregation

Extremely large malicious values dominate the insecure feature aggregation and global prediction.

Secure feature aggregation to limit the adversarial effect!
• Robust masking to detect and remove large values
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature

Local feature map

Local prediction / logits map

[cat]
[dog]
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature

![Image of a dog with a red box around it](image)

Local feature map

Local prediction / logits map
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature

Local feature map

Local prediction / logits map
Leveraging Local Logits for Robust Masking

Local logits: making local prediction based on the local feature
Leveraging Local Logits for Robust Masking

Aggregating local logits gives the same global logits prediction
Leveraging Local Logits for Robust Masking

Aggregating local logits gives the same global logits prediction

Local prediction / logits map

Global prediction / logits

Local feature map
A Better Visualization: Local Logits Map Slice

- One local logits map slice for one class
- Class evidence: elements of each slice

Local logits map slice for cat:

\[
\begin{bmatrix}
30 & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1
\end{bmatrix}
\]

Cat: 35

Local logits map slice for dog:

\[
\begin{bmatrix}
0 & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4
\end{bmatrix}
\]

Dog: 27
Robust Masking: Algorithm

1. Clip all negative values to zeros
2. Move a sliding window over each local logits slice (1 × 1 window here)
3. Calculate class evidence sum within each window
4. Mask the window with the highest sum

Local logits map slice for cat

\[
\begin{bmatrix}
30 & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1 \\
\end{bmatrix}
\]

**Cat: 35**

Local logits map slice for dog

\[
\begin{bmatrix}
0 & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4 \\
\end{bmatrix}
\]

**Dog: 27**
Robust Masking: Prediction in the Adversarial Setting

Robust Masking:
1. Clip all negative values to zeros
2. Move a **sliding window** over each local logits slice (1 × 1 window here)
3. Calculate class evidence **sum** within each window
4. Mask the window with the **highest** sum

Local logits map slice for cat

- **Cat:** 35

Local logits map slice for dog

- **Dog:** 27
Robust Masking: Prediction in the Adversarial Setting

Robust Masking:
1. Clip all negative values to zeros
2. Move a sliding window over each local logits slice (1 × 1 window here)
3. Calculate class evidence sum within each window
4. Mask the window with the highest sum

The prediction in the adversarial setting is subject to partial feature masking

Local logits map slice for cat
Cat: 5

Local logits map slice for dog
Dog: 20
Robust Masking: Prediction in the Clean Setting

Robust Masking:
1. Clip all negative values to zeros
2. Move a **sliding window** over each local logits slice (1 × 1 window here)
3. Calculate class evidence **sum** within each window
4. Mask the window with the **highest** sum

Local logits map slice for cat
Cat: 5

Local logits map slice for dog
Dog: 28
Robust Masking: Prediction in the Clean Setting

Robust Masking:
1. Clip all negative values to zeros
2. Move a sliding window over each local logits slice (1 × 1 window here)
3. Calculate class evidence sum within each window
4. Mask the window with the highest sum

The prediction in the clean setting is generally invariant to partial feature masking
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

PatchGuard: A Provably Robust Defense Framework

- **Small Receptive Field**: Bound the number of corrupted features
- **Secure Feature Aggregation**: Do robust prediction on partially corrupted features
Our Contribution: PatchGuard Defense Framework with Provable Robustness

PatchGuard aims to prevent the localized patch from dominating the global prediction.

PatchGuard: A Provably Robust Defense Framework

- Small Receptive Field: Bound the number of corrupted features.
- Secure Feature Aggregation: Do robust prediction on partially corrupted features.
Recall: Provable Robustness on Certified Test Images

Test Image

Threat Model
(patch sizes, shapes, and location set)

Provable Analysis

Ground-truth $y$ → The prediction is always correct
“$I$ think it is a dog”
“My prediction will never change”
or
“I can’t say anything for sure”

Provable robust accuracy / certified accuracy: the fraction of test images that are
1. Correctly classified
2. Provably robust to any (adaptive) localized patch attack within the threat model
Recall: Provable Robustness on Certified Test Images

Test Image

Threat Model
(patch sizes, shapes, and location set)

Provable Analysis

Ground-truth $y$

The prediction is **always** correct

“I think it is a dog”

“My prediction will never change”

or

“I can’t say anything for sure”

**Provable robust accuracy / certified accuracy:** the fraction of test images that are

1. **Correctly classified**
2. **Provably robust** to any (adaptive) localized patch attack within the threat model
Recall: Provable Robustness on Certified Test Images

Test Image

Threat Model
(patch sizes, shapes, and location set)

Provable Analysis

Ground-truth $y$

The prediction is always correct

“I think it is a dog”

“My prediction will never change”

or

“I can’t say anything for sure”

Provable robust accuracy / certified accuracy: the fraction of test images that are

1. Correctly classified
2. Provably robust to any (adaptive) localized patch attack within the threat model
Provable Analysis

The adversary can control values within a small window ($1 \times 1$ window here)

\[
\begin{bmatrix}
? & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1
\end{bmatrix}
\]

local logits map slice for cat

Cat: ?

\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4
\end{bmatrix}
\]

local logits map slice for dog

Dog: ?
Provable Analysis: Upper Bound of Class Evidence

The adversary can control values within a small window (1 × 1 window here)

1. The adversary cannot increase the malicious class evidence too much

\[
\begin{bmatrix}
? & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1
\end{bmatrix}
\]

local logits map slice for cat

Cat: ?

\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4
\end{bmatrix}
\]

local logits map slice for dog

Dog: ?
**Provable Analysis: Upper Bound of Class Evidence**

The adversary can control values within a small window (1 × 1 window here)

1. **The adversary cannot increase the malicious class evidence too much**
   - A large value will be masked

```
\[
\begin{bmatrix}
3 & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1 \\
\end{bmatrix}
\]
```

local logits map slice for cat

*Cat:* ?

```
\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4 \\
\end{bmatrix}
\]
```

local logits map slice for dog

*Dog:* ?
The adversary can control values within a small window (1 × 1 window here)

1. **The adversary cannot increase the malicious class evidence too much**
   - A large value will be masked
   - The robust masking imposes an upper bound of the class evidence sum
Provable Analysis: Lower Bound of Class Evidence

The adversary can control values within a small window (1 × 1 window here)

1. The adversary cannot increase the malicious class evidence too much
   - A large value will be masked
   - The robust masking imposes an upper bound of the class evidence sum

2. The adversary cannot decrease the benign class evidence too much

\[
\begin{bmatrix}
2 & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1 \\
\end{bmatrix}
\]

Local logits map slice for cat
Cat: 5

\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4 \\
\end{bmatrix}
\]

Local logits map slice for dog
Dog: ?
The adversary can control values within a small window (1 × 1 window here)

1. **The adversary cannot increase the malicious class evidence too much**
   - A large value will be masked
   - The robust masking imposes an upper bound of the class evidence sum

2. **The adversary cannot decrease the benign class evidence too much**
   - Can only push malicious values to zero
Provable Analysis: Lower Bound of Class Evidence

The adversary can control values within a small window (1 × 1 window here)

1. The adversary cannot increase the malicious class evidence too much
   • A large value will be masked
   • The robust masking imposes an upper bound of the class evidence sum

2. The adversary cannot decrease the benign class evidence too much
   • Can only push malicious values to zero
   • Clipping all negative values imposes a lower bound of the class evidence sum
The adversary can control values within a small window (1 × 1 window here)

1. **The adversary cannot increase the malicious class evidence too much**
   - A large value will be masked
   - The robust masking imposes an upper bound of the class evidence sum

2. **The adversary cannot decrease the benign class evidence too much**
   - Can only push malicious values to zero
   - Clipping all negative values imposes a lower bound of the class evidence sum

We can derive bounds that apply to any attack strategy! (formal proof in the paper)
Provable Analysis: Example

Local logits map slice for cat

\[
\begin{bmatrix}
? & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1 \\
\end{bmatrix}
\]

Cat: ?

Local logits map slice for dog

\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4 \\
\end{bmatrix}
\]

Dog: ?

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Dog</td>
<td>20</td>
<td>27</td>
</tr>
</tbody>
</table>
Provable Analysis: Example

\[
\begin{bmatrix}
? & 0 & 0 \\
1 & 0 & 1 \\
2 & 0 & 1
\end{bmatrix}
\]

local logits map slice for cat

\textbf{Cat: ?}

\[
\begin{bmatrix}
? & 2 & 2 \\
0 & 7 & 6 \\
1 & 5 & 4
\end{bmatrix}
\]

local logits map slice for dog

\textbf{Dog: ?}

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Dog</td>
<td>20</td>
<td>27</td>
</tr>
</tbody>
</table>

- 20 (lower bound of dog) > 5 (upper bound of cat)
  - Provably Robust (always predicts dog)!
- Try all possible patch locations
  - This image is certified :)

Threat Model
(patch sizes, shapes, and location set)
1. PatchGuard achieves substantial provable robustness
   (robustness evaluated against a 2%-pixel square patch anywhere on the image)
Evaluation: Substantial Provable Robustness

<table>
<thead>
<tr>
<th></th>
<th>10-class ImageNette</th>
<th>1000-class ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>Clean</td>
<td>Clean</td>
</tr>
<tr>
<td><strong>PatchGuard</strong></td>
<td>95.0%</td>
<td>54.6%</td>
</tr>
<tr>
<td></td>
<td>86.7%</td>
<td>26%</td>
</tr>
</tbody>
</table>

1. PatchGuard achieves substantial provable robustness
   (robustness evaluated against a 2%-pixel square patch *anywhere* on the image)
Evaluation: Substantial Provable Robustness

<table>
<thead>
<tr>
<th></th>
<th>10-class ImageNette</th>
<th>1000-class ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>Clean</td>
<td>Robust</td>
</tr>
<tr>
<td><strong>PatchGuard</strong></td>
<td>95.0%</td>
<td>86.7%</td>
</tr>
<tr>
<td><strong>PatchGuard-Top-5</strong></td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Clean</td>
<td>Robust</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PatchGuard</strong></td>
<td>54.6%</td>
<td>26%</td>
</tr>
<tr>
<td><strong>PatchGuard-Top-5</strong></td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>76.6%</td>
<td>56.9%</td>
</tr>
</tbody>
</table>

Top-5 accuracy for ImageNet is good!

1. PatchGuard achieves substantial provable robustness
   (robustness evaluated against a 2%-pixel square patch anywhere on the image)
Evaluation: State-of-the-art Clean Accuracy and Provable Robust Accuracy

<table>
<thead>
<tr>
<th></th>
<th>10-class ImageNette</th>
<th>1000-class ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Clean</td>
<td>Robust</td>
</tr>
<tr>
<td><strong>PatchGuard</strong></td>
<td>95.0%</td>
<td>86.7%</td>
</tr>
<tr>
<td>IBP [1]</td>
<td>Computationally infeasible</td>
<td></td>
</tr>
<tr>
<td>CBN [2]</td>
<td>94.9%</td>
<td>60.9%</td>
</tr>
<tr>
<td>DS [3]</td>
<td>92.1%</td>
<td>79.1%</td>
</tr>
</tbody>
</table>

2. IBP is too computationally expensive to scale to high-resolution images
3. PatchGuard significantly outperforms CBN and DS
   - Improvement from CBN on ImageNet:
     - 5% clean accuracy; 19% provable robust accuracy (2x better!)
   - Improvement from DS on ImageNet:
     - 10% clean accuracy; 12% provable robust accuracy (1x better!)

[3] Levine et al., “(De)randomized smoothing for certifiable defense against patch attacks,” NeurIPS 2020
Discussion: Generalizability of PatchGuard

PatchGuard as a general defense framework

<table>
<thead>
<tr>
<th>Provably Robust Defense</th>
<th>Small receptive field</th>
<th>Secure feature aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatchGuard (ours)</td>
<td>BagNet</td>
<td>Robust masking</td>
</tr>
</tbody>
</table>
**Discussion: Generalizability of PatchGuard**

PatchGuard as a general defense framework

<table>
<thead>
<tr>
<th>Provably Robust Defense</th>
<th>Small receptive field</th>
<th>Secure feature aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatchGuard (ours)</td>
<td>BagNet</td>
<td>Robust masking</td>
</tr>
<tr>
<td>Clipped BagNet (CBN) [1]</td>
<td>BagNet</td>
<td>Clipping + Average pooling</td>
</tr>
<tr>
<td>Derandomized Smoothing (DS) [2]</td>
<td>Pixel patches to ResNet</td>
<td>Majority voting</td>
</tr>
</tbody>
</table>
## Discussion: Generalizability of PatchGuard

PatchGuard as a general defense framework

<table>
<thead>
<tr>
<th>Provably Robust Defense</th>
<th>Small receptive field</th>
<th>Secure feature aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatchGuard (ours)</td>
<td>BagNet</td>
<td>Robust masking</td>
</tr>
<tr>
<td>Clipped BagNet (CBN) [1]</td>
<td>BagNet</td>
<td>Clipping + Average pooling</td>
</tr>
<tr>
<td>Derandomized Smoothing (DS) [2]</td>
<td>Pixel patches to ResNet</td>
<td>Majority voting</td>
</tr>
</tbody>
</table>


[2] Levine et al., “(De)randomized smoothing for certifiable defense against patch attacks,” NeurIPS 2020


Discussion: Limitations

1. The small receptive field hurts the clean accuracy (provable robustness vs. clean accuracy trade-off)
   • The accuracy drop is especially obvious for ImageNet (from 76.1% to 56.5%)

<table>
<thead>
<tr>
<th>Model</th>
<th>10-class ImageNette</th>
<th>1000-class ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clean</td>
<td>Robust</td>
</tr>
<tr>
<td>ResNet-50 (483 × 483)</td>
<td>99.6%</td>
<td>--</td>
</tr>
<tr>
<td>BagNet-17 (17 × 17)</td>
<td>95.9%</td>
<td>--</td>
</tr>
<tr>
<td>PatchGuard</td>
<td>95.0%</td>
<td>86.7%</td>
</tr>
<tr>
<td>PatchGuard-Top-5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. The masking operation requires additional parameters (e.g., number of masks, mask size, mask shape)
Takeaways

1. **PatchGuard: a General Defense Framework**
   - Small receptive field
   - Secure feature aggregation

2. **Provably Robust Defense**
   - Predictions are always correct on certified images

3. **State-of-the-art Defense Performance**
   - Clean accuracy
   - Provable robust accuracy
Thank you!

Chong Xiang  
Princeton University  
cxiang@princeton.edu

Arjun Nitin Bhagoji  
University of Chicago  
abhagoji@uchicago.edu

Vikash Sehwag  
Princeton University  
vvikash@princeton.edu

Prateek Mittal  
Princeton University  
pmittal@princeton.edu

Technical Report  
GitHub