Jujutsu: A Two-stage Defense against Adversarial Patch Attacks on Deep Neural Networks

Zitao Chen, Pritam Dash, Karthik Pattabiraman
Adversarial attacks

- Input + perturbations $\rightarrow$ misclassification.
- Perturbations with different properties.

Universally malicious

Physically realizable

Moosavi-Dezfooli et al. CVPR’17

Eykolt et al. CVPR’18
Adversarial patch attacks

- Universally malicious and physically realizable.
- Localized adversarial patch to trigger misclassification.

Brown et al. 2017

Universally effective on any position
Defense challenges

| Existing techniques |  |
|---------------------|--|
| Detection performance | Low [1] |
| False positive | High [1-5] |

[1] Chou et al., Sentinet: Detecting localized universal attacks against deep learning systems. SPW’20
[2] Naseer et al., Local gradients smoothing: Defense against localized adversarial attacks. WACV’19
[3] Rao et al., Adversarial Training against Location-Optimized Adversarial Patches. ArXiv’20
[4] Wu et al., Defending Against Physically Realizable Attacks on Image Classification. ICLR’20
[5] Xiang et al., "PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking." USENIX’21.
Defense challenges

|                      | Existing techniques |
|----------------------|---------------------|
| Detection performance| Low [1]             |
| False positive       | High [1-5]          |
| Mitigation performance| Low [2-5]          |

[1] Chou et al., Sentinet: Detecting localized universal attacks against deep learning systems. SPW’20
[2] Naseer et al., Local gradients smoothing: Defense against localized adversarial attacks. WACV’19
[3] Rao et al., Adversarial Training against Location-Optimized Adversarial Patches. ArXiv’20
[4] Wu et al., Defending Against Physically Realizable Attacks on Image Classification. ICLR’20
[5] Xiang et al., "PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking." USENIX’21.
## Defense challenges

| Detection performance | Existing techniques |
|-----------------------|---------------------|
| Low [1]               |                     |
| False positive        | High [1-5]          |
| Mitigation performance| Low [2-5]           |
| Configurable          | Not supported [1-5] |

[1] Chou et al., Sentinet: Detecting localized universal attacks against deep learning systems. SPW’20
[2] Naseer et al., Local gradients smoothing: Defense against localized adversarial attacks. WACV’19
[3] Rao et al., Adversarial Training against Location-Optimized Adversarial Patches. ArXiv’20
[4] Wu et al., Defending Against Physically Realizable Attacks on Image Classification. ICLR’20
[5] Xiang et al., "PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking." USENIX’21.
This work - Jujutsu

|                           | Existing techniques | Jujutsu |
|---------------------------|---------------------|---------|
| Detection performance     | Low [1]             | High    |
| False positive            | High [1-5]          | Low     |
| Mitigation performance    | Low [2-5]           | High    |
| Configurable              | Not supported [1-5] | Supported |

[1] Chou et al., Sentinet: Detecting localized universal attacks against deep learning systems. SPW’20
[2] Naseer et al., Local gradients smoothing: Defense against localized adversarial attacks. WACV’19
[3] Rao et al., Adversarial Training against Location-Optimized Adversarial Patches. ArXiv’20
[4] Wu et al., Defending Against Physically Realizable Attacks on Image Classification. ICLR’20
[5] Xiang et al., "PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking." USENIX’21.
Threat model

Adversary

- White-box adversary.
- Access to a surrogate dataset.
- Goal: Universal targeted misclassification [Brown et al. 2017].

Defender

- A hold-out set (random samples hidden from the adversary).
- Goal: Attack detection & mitigation.
  - Mitigation $\rightarrow$ correct prediction on adv samples.
Jujutsu

Turning the adversary’s strength against the adversary

Patch attacks: Universally malicious

Consistent misclassification on any samples

Exposed for attack detection

Patch attacks: Localized perturbations

Most features are uncorrupted

Utilized by image inpainting for attack mitigation
Jujutsu

Turning the adversary’s strength against the adversary

Patch attacks:
Universally malicious

Consistent misclassification on any samples

Exposed for attack detection

Patch attacks:
Localized perturbations

Most features are uncorrupted

Utilized by image inpainting for attack mitigation
Adversary’s strength

- Adversarial patch is universally malicious.

- Cricket $\rightarrow$ toaster
- Brown bear $\rightarrow$ toaster
- Helmet $\rightarrow$ toaster
Attack detection by Jujutsu

- Expose the **consistent** misclassification by the patch attacks.

1. Target sample
2. toaster
3. Same label → attack detected
Attack detection (HOW TO)

1. How to locate the target image patch?

2. How to perform the patch transplantation?
Locating the target image patch

- Adversarial patch has **high influence** to the output.
- **Saliency map** inspection → Locate **high-influence** region.

(Processed) saliency map
Locating the target image patch

- Adversarial patch has high influence to the output.
- Saliency map inspection → Locate high-influence region

What if the image patch is uncorrupted?
Verify adversarial patch

- Adv patch causes **consistent misclassification** on any sample.
  - Exposed by using the hold-output sample.

```
Target sample

Random hold-out sample
```

```
toaster

Same (mis-classified) label

toaster
```
Verify **benign** patch

- Benign image patch is **not** universally malicious.
Adversarial vs. benign samples.

- Locate target patch $\rightarrow$ patch transplantation $\rightarrow$ pred comparison.
Patch transplantation affects false positive

Target sample

Sloth bear

Same label $\rightarrow$
adv sample (FP)!

Hold-out sample

Sloth bear

Different labels $\rightarrow$
benign sample

corgi
Why false positive?

Target sample

Hold-out sample

Original hold-out sample

Sloth bear

Main feature region

Corgi

Background feature region
Avoiding false positive

Original hold-out sample → Saliency map → Identify background-feature region → Target sample

Different labels
Benign sample

Hold-out sample
Jujutsu

Turning the adversary’s strength against the adversary

Patch attacks:
Universally malicious

Consistent misclassification on any samples

Exposed for attack detection

Patch attacks:
Localized perturbations

Most features are uncorrupted

Utilized by image inpainting for attack mitigation
Adversary’s strength

- Localized perturbations for physically realizable attack.
Attack mitigation by Jujutsu

- The majority of features are **uncorrupted**.
- Utilize uncorrupted features to reconstruct clean samples.
  - image inpainting.

![Diagram showing masking and image inpainting process](image-url)
Use the label on the *inpainted* sample as final output.
Evaluation

- 4 Datasets: ImageNet, ImageNette, CelebA, Place365.
- 6 patch sizes: 5% - 10%.
- 7 architectures: ResNet, DenseNet, VGG, etc.
- Jujutsu: configured with highest defense performance (more in the paper).
Overall results

Adversarial samples

95.93% detected
79.73% mitigated

Match the accuracy on benign samples

Benign samples

0.7% mis-detected
Comparison with related defenses.

Jujutsu outperforms related techniques on both attack detection and attack mitigation.
Physical-world attack

Jujutsu detects & mitigates >95% adversarial samples with 3% FPR.
Adaptive attack

- Jujutsu: Detects adv patch from high-influence region.
- Adversary: Force the adv patch to remain low influence.
  - Approach: Manipulate the saliency map.

Low-influence patch attack suffers from poor attack success (99% → 5%)
Other attack variants

☑ Multi-patch attack.

☑ Patch in a different shape (e.g., rectangular).

✗ Untargeted attack.
Summary

Jujutsu: A two-stage defense against adversarial patch attacks.

**Attack detection**
Adversary: universal attacks
Jujutsu: expose attacks’ consistent misclassification

**Attack mitigation**
Adversary: localized attacks
Jujutsu: utilize the uncorrupted features → clean samples

Code → https://github.com/DependableSystemsLab/Jujutsu
Question → zitaoc@ece.ubc.ca