# DorPatch: Distributed and Occlusion-Robust Adversarial Patch to Evade Certifiable Defenses

*Chaoxiang He, Xiaojing Ma, Bin B. Zhu, Yimiao Zeng, Hanqing Hu, Xiaofan Bai, Hai Jin, and Dongmei Zhang*

*Presented by Bin B. Zhu*

# Background

Adversarial patch attacks pose a great threat in real world applications

Targeted Attack in
Traffic Sign Recognition

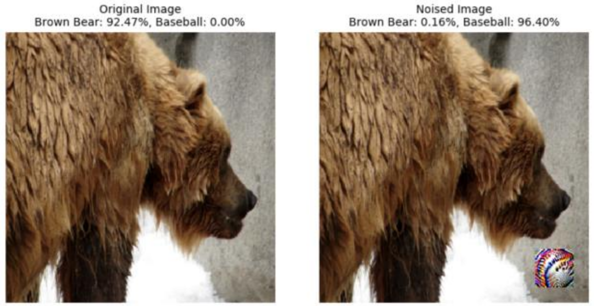

Original Patched Recognition Result

Impersonation Attack in
Biometric Authentication

Original Patched Recognition Result

# Typical Adversarial Patch Attacks

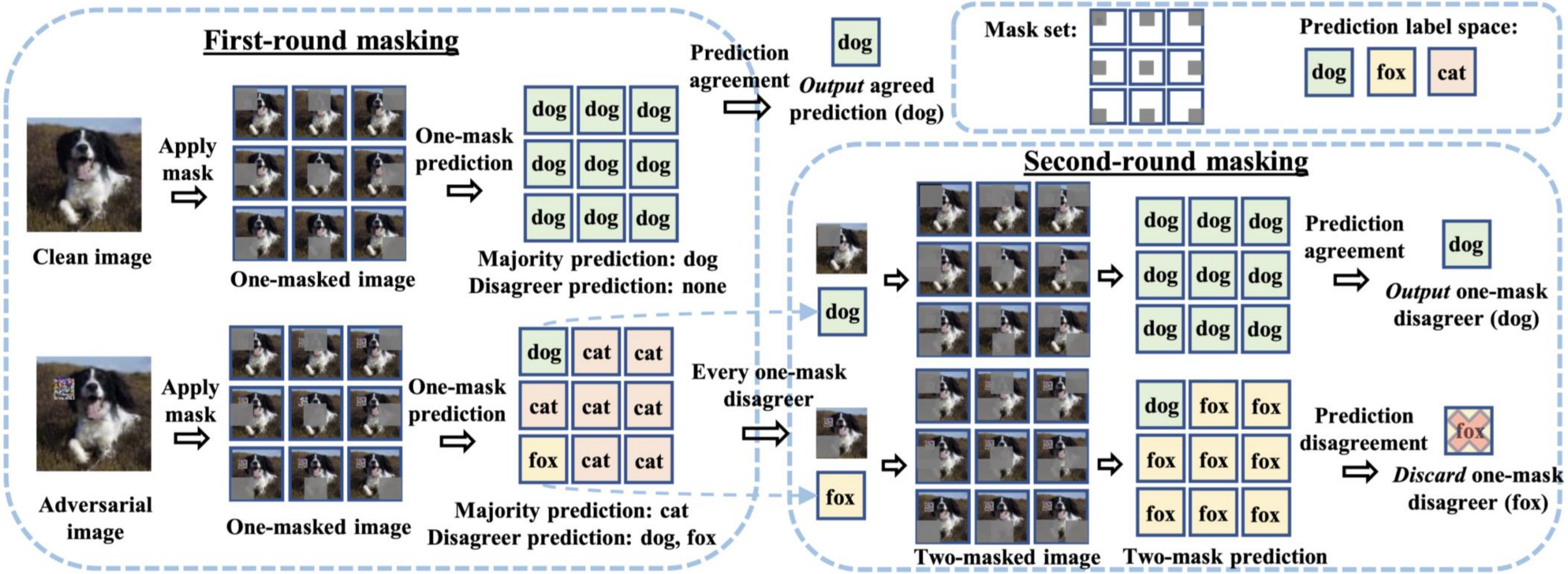
- **LaVAN**: localized patch using prefixed mask:
$$\min_{\Delta} L_{adv}(X_{\Delta})$$
- **LOAP**: also *optimize patch location via moving the patch in different directions.*
- **$RP_2$**: generate a distributed graffiti-like adversarial patch (e.g., sticks)
- **IAP**: generates an *inconspicuous* patch with Adversarial Generative Networks (GAN)




Brown Bear (92.5%) $\rightarrow$ **Baseball** (96.4%)


# Adversarial Patch Defenses - Certifiable

- **PatchCleanser** (the state-of-the-art defense)




Two-round masking operations

# Adversarial Patch Defenses - Certifiable

- **Assumptions of PatchCleanser**
  - The model is robust to occlusion of a small-size mask at arbitrary locations of an input image
    - requires that *the mask should be small enough to avoid significant degradation of the model's clean accuracy*
  - The adversarial patch can be fully occluded by the mask at an appropriate location
    - requires *the mask to be large enough to completely cover the adversarial patch*

# Our Threat Model

White-box access to the DNN model under attack

*Full access to the DNN model, including its architecture and parameters*

Black-box access to potential defenses against DorPatch

*No knowledge of any defense (its characteristics or settings) against DorPatch*

# Limitations of Existing Adv. Patch Attacks

- Existing adversarial patch attacks typically employ a ***localized*** patch.
  - Many attacks use *predetermined and fixed* shape, location, and size of the patch
    - The patch may not be optimal, resulting in a less powerful adversarial attack
  - Adversarial pixels typically *located in a small, restricted region*
    - Exploited by certifiable robustness defenses (e.g. PatchCleanser) to detect and neutralize adversarial patches

# Is Distributed Enough to Evade PatchCleanser?

- RP2 uses a distributed graffiti-like adversarial patch
  - May not be fully covered by a single mask in PatchCleanser
- Distributed adversarial patch is ***insufficient*** to evade PatchCleanser
  - The masking operation in PatchCleanser may *corrupt* the patch, causing it to *lose its adversarialness*
    - PatchCleanser can predict correctly
  - It cannot make adversarially patched examples certifiable by PatchCleanser (*much harder than causing misprediction*)

# Desired Properties of Patch Attacks

## Distributed

- Widely distributed to prevent being fully occluded by a small exploring mask

## Robust to Partial Occlusions

- Robust to partial occlusions at various locations
- Not only to make PatchCleanser mispredict but also to be certifiably robust by PatchCleanser

## Fully Optimized

- Patch is fully optimized, including its shape, location, and pixel values, to achieve the most effective attack within a given patch budget

## Inconspicuous

- To enhance the inconspicuousness and avoid being neutralized by image processing techniques
  - Perturbed pixels should result in *structural indistinguishability* and
  - *Perceptual masking* should be considered when determining the locations and pixel values of perturbed pixels

# Fullfillment of Desired Properties

| Attack\Property | Distributed | Robust to Occlusion | Inconspicuous | Location-optimized |
|---|---|---|---|---|
| **DorPatch** | ✓ | ✓ | ✓ | ✓ |
| LaVAN | | | | |
| LOAP | | | | ✓ |
| IAP | | | ✓ | ✓ |
| $RP_2$ | ✓ | | | ✓ |

# Achieving Desired Properties in DorPatch

Density Regularization → Distributed

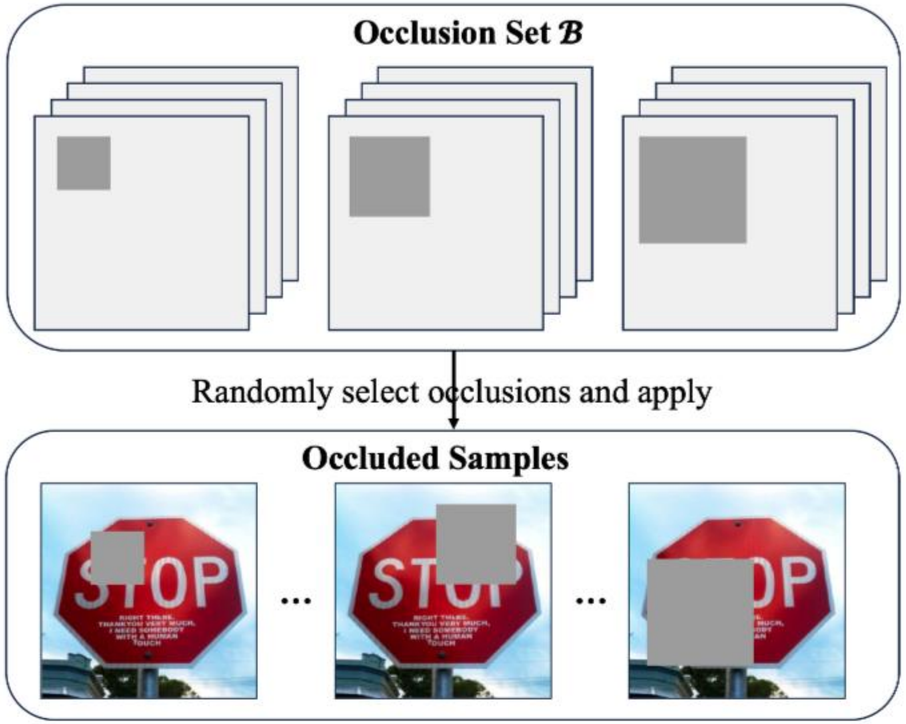
- **Goal**: To encourage a patch to be widely and uniformly distributed
- **Method:**
  - Use a set of sampling regions, $\mathcal{A}$, to divide an image evenly into $|\mathcal{A}|$ parts
  - Make the density of patch pixels in each region similar by minimizing the *standard deviation* of the number of patch pixels in each sampling region over all regions in $\mathcal{A}$

$$L_{den} = \sqrt{\frac{1}{|\mathcal{A}|}\sum_{\mathbf{a}\in\mathcal{A}} \left(M \cdot \mathbf{a} - \mathbb{E}_{\mathbf{a}\in\mathcal{A}}(M \cdot \mathbf{a})\right)^2}$$

# Achieving Desired Properties in DorPatch

Image Dropout → Robust to Partial Occlusions

- **Goal**: robust to partial occlusions and certifiably robust by PatchCleanser
- **Method:** randomly mask out parts of the image during the patch optimization process:
  - Collect a set of possible occlusions, $\mathcal{B}$, such as squares of *different sizes and positions*
  - Generate $\mathcal{N}$ occluded images, $X_\Delta^i, i \in [1, \mathcal{N}]$, from the patched image $X_\Delta$ and optimize them together
    - Randomly choose $n_o$ occlusions from $\mathcal{B}$ and remove the corresponding regions from $X_\Delta$ to obtain each $X_\Delta^i$

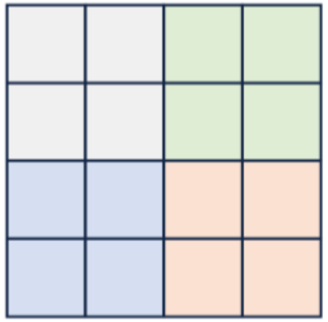


# Achieving Desired Properties in DorPatch

Group Lasso on Mask → Fully Optimized

- **Goal**: fully optimized while physically realizable
- **Method**
  - Patch consists of isolated parts (groups)
    - Each group is large enough and has a regular shape
    - A group is either included in or excluded from the patch as a whole
  - Apply group lasso to the mask M to enforce group sparsity, i.e., to minimize the number of groups in the patch

$$L_{grp} = \sum_{l=1}^{m} \| M \circ G_l \|_2$$



# Achieving Desired Properties in DorPatch

Structural Loss → Inconspicuous

- **Goal**
  - To encourage perturbed pixels to result in continuous and smooth structures
- **Method**

$$L_{str} = \sum_{x_i \in X_\Delta} \frac{1}{V_i} \left( \sum_{x_j \in N(x_i)} (x_i - x_j)^2 \cdot \min_{x_j \in N(x_i)} (x_i - x_j)^2 \right)$$

$V_i$: approximate the *local perceptual masking* power at a pixel $x_i$

**total variation loss:**
- ➢ Encourages smooth changes among neighboring pixels for each perturbed pixel

**minimal variance loss**
- ➢ Small when a neighboring pixel has a similar value
- ➢ Allows preserving a sharply changing pixel as long as at least one neighboring pixel has a similar pixel value (e.g., an edge pixel)

# Generation of Adversarial Patches

- DorPatch's ***optimization problem*** (together with image dropout)

$$\min_{M,\Delta} L_{adv} + \lambda_1 \cdot L_{grp} + \lambda_2 \cdot L_{den} + \lambda_3 \cdot L_{str}$$

$$s.t.\ \|X_\Delta - X\|_p \leq \epsilon$$

- It is a Mixed Integer Programming (MIP) problem
  - Mask M consists of 0s and 1s: cannot be directly optimized
- Solving it with our two-stage method
  - 1st stage: Generate mask
    - Relax the binary constraint on $M$ by allowing continuous values in [0, 1] (i.e., as a *transparency mask*) to obtain a fractional mask $M$
    - Threshold $M$ to obtain a binary mask by selecting the groups with the highest values
  - 2nd stage: Generate patch's pixel values
    - Fix the binary mask M to determine the optimal pixel values of the adversarial patch.

# Attacking Performance against PatchCleanser

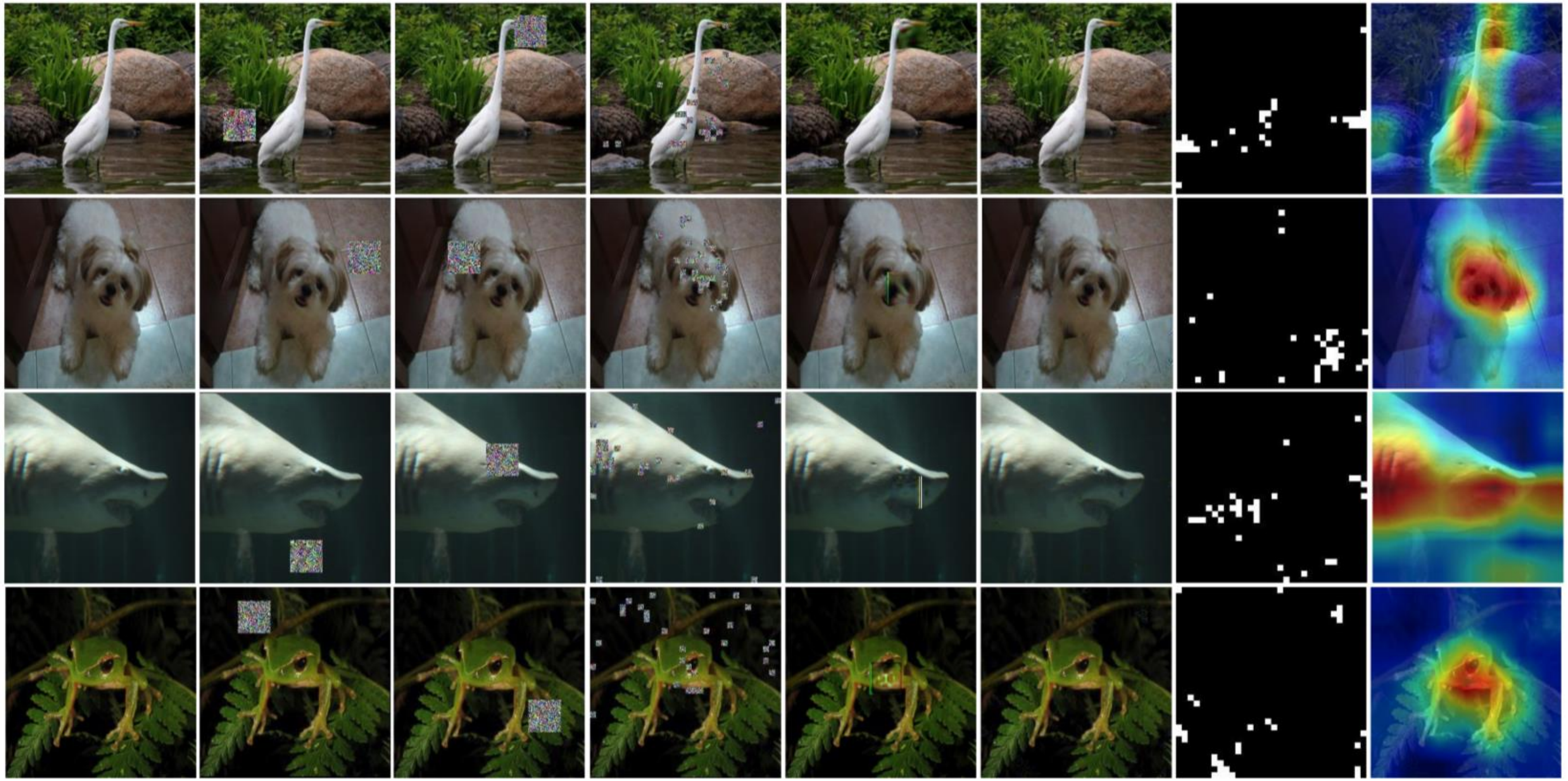
## CIFAR10

| PB | Attack | Robust Accuracy (in %) without Defense | PatchCleanser (MS) 3% | PatchCleanser (MS) 6% | PatchCleanser (MS) 12% | CRPE (in %) PatchCleanser (MS) 3% | 6% | 12% |
|---|---|---|---|---|---|---|---|---|
| 3% | DorPatch | 7.6 | 13.2 | 13.2 | 12.7 | 40.6 | 37.6 | 33.0 |
| | LaVAN | 4.9 | 98 | 95.6 | 94.3 | 0.0 | 0.0 | 0.0 |
| | LOAP | 5.3 | 96.8 | 96.4 | 92.7 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 77.7 | 78.5 | 80.2 | 0.8 | 0.4 | 0.4 |
| 6% | DorPatch | 0.0 | 0.0 | 0.0 | 0.0 | 78.8 | 68.2 | 60.6 |
| | LaVAN | 0.8 | 93.1 | 94.3 | 92.7 | 0.0 | 0.0 | 0.0 |
| | LOAP | 0.8 | 93.5 | 93.5 | 93.1 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 60.7 | 67.6 | 66.8 | 1.21 | 1.21 | 0.8 |
| 12% | DorPatch | 0.0 | 0.0 | 0.0 | 0.0 | 90.9 | 86.4 | 76.3 |
| | LaVAN | 0.0 | 86.6 | 92.3 | 93.5 | 0.0 | 0.0 | 0.0 |
| | LOAP | 0.0 | 85.8 | 92.7 | 92.7 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 42.5 | 44.9 | 52.2 | 1.6 | 1.6 | 0.4 |

## ImageNet

| PB | Attack | Robust Accuracy (in %) without Defense | PatchCleanser (MS) 3% | PatchCleanser (MS) 6% | PatchCleanser (MS) 12% | CRPE (in %) PatchCleanser (MS) 3% | 6% | 12% |
|---|---|---|---|---|---|---|---|---|
| 3% | DorPatch | 4.4 | 10.2 | 9.8 | 11.2 | 49.8 | 44.9 | 38.1 |
| | LaVAN | 6.2 | 89.1 | 90.6 | 86.3 | 0.0 | 0.0 | 0.0 |
| | LOAP | 4.7 | 89.5 | 89.9 | 86.4 | 0.0 | 0.0 | 0.0 |
| | IAP | 36.7 | 80.9 | 78.1 | 78.1 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 56.4 | 58.4 | 63.0 | 0.8 | 0.4 | 0.0 |
| 6% | DorPatch | 0.8 | 1.2 | 1.2 | 1.2 | 80.9 | 69.7 | 57.6 |
| | LaVAN | 1.2 | 82.8 | 86.7 | 85.6 | 0.0 | 0.0 | 0.0 |
| | LOAP | 0.4 | 82.9 | 86.8 | 84.8 | 0.0 | 0.0 | 0.0 |
| | IAP | 27.0 | 71.5 | 71.5 | 71.5 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 25.8 | 38.1 | 43.2 | 2.7 | 0.4 | 0.4 |
| 12% | DorPatch | 0.8 | 1.0 | 1.0 | 1.0 | 87.1 | 83.1 | 75.8 |
| | LaVAN | 0.0 | 76.2 | 78.1 | 81.6 | 0.0 | 0.0 | 0.0 |
| | LOAP | 0.0 | 77.4 | 78.9 | 78.9 | 0.0 | 0.0 | 0.0 |
| | IAP | 25.4 | 61.1 | 63.2 | 63.7 | 0.0 | 0.0 | 0.0 |
| | $RP_2$ | 0.0 | 16.7 | 19.8 | 22.2 | 5.5 | 2.5 | 0.8 |

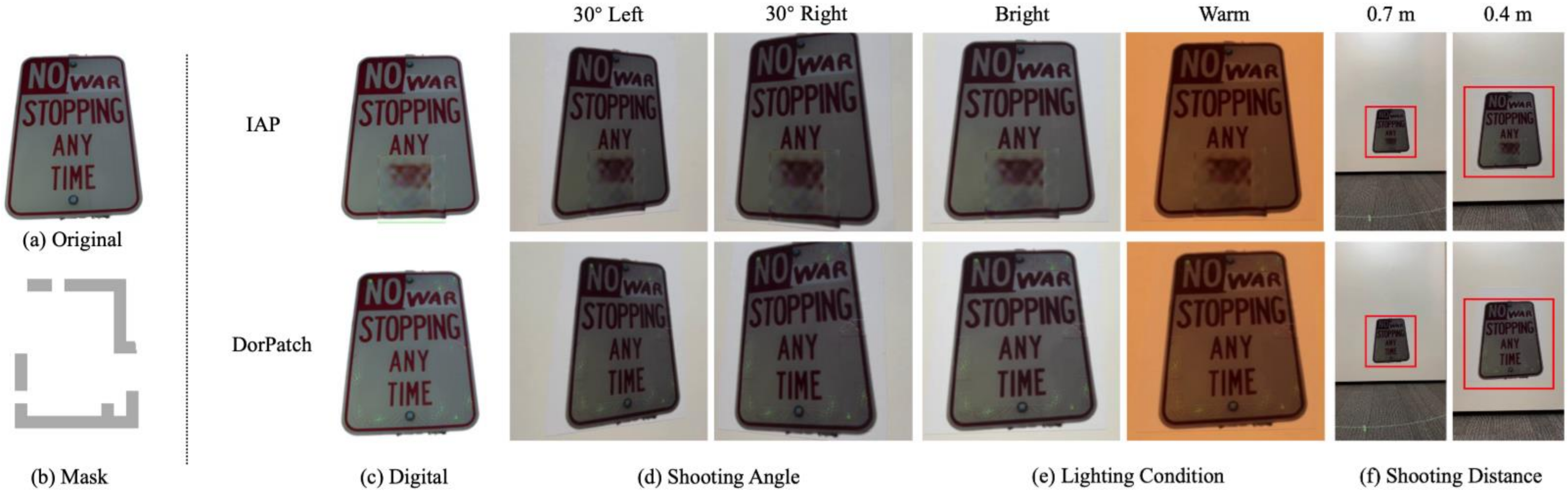
# Perceptual Quality



(a) Origin (b) LaVAN (c) LOAP (d) $RP_2$ (e) IAP (f) DorPatch (g) Mask (h) Saliency Map

# Physical-world Attack Performance

| Attack | without Defense | PatchCleanser (MS) | | |
|---|---|---|---|---|
| | | 3% | 6% | 12% |
| No Attack | 100.0 | 100.0 | 100.0 | 100.0 |
| IAP | 16.0 | 91.4 | 90.4 | 66.4 |
| DorPatch | 0.0 | 2.4 | 1.4 | 2.1 |



# Attacking Performance against Adaptive Defenses

## Adversarial Training

- the robust WRN28-4 model from Hydra is adversarially trained using a PGD attack with 50 steps and an 8/255 $L_\infty$ budget
- the robust ResNet110 model from DOA is trained using a ***rectangular occlusion attack*** with an 11 $\times$ 11 rectangle (patch budget=12%)

| Model | | Patch Budget (in %) | | | |
|---|---|---|---|---|---|
| **Arch.** | **Training Type** | **1.5** | **3** | **6** | **12** |
| WRN28-4 | Normal | 10.8 | 0.6 | 0.7 | 0.1 |
| | Adv. Trained | 70.7 | 57.3 | 31.4 | 13.0 |
| ResNet110 | Normal | 11.9 | 2.3 | 1.2 | 0.4 |
| | Adv. Trained | 64.7 | 29.6 | 7.0 | 0.9 |

CIFAR10

## PatchCleanser Using Multiple Masks

- PatchCleanser can be extended to defend against a distributed adversarial patch comprising multiple separated subpatches by ***applying multiple masks to mask out multiple regions simultaneously***
- The number of model inferences ***explodes exponentially*** as number of subpatches increases

| Patch Budget | Mask Size of PatchCleanser 3% | 6% | 12% |
|---|---|---|---|
| 3% | 16.3 (+6.1) | 15.6 (+5.8) | 15.6 (+4.4) |
| 6% | 3.1 (+1.9) | 3.5 (+2.3) | 4.3 (+3.1) |
| 9% | 2.4 (+1.4) | 2.4 (+1.4) | 3.2 (+2.2) |

ImageNet

# Conclusion

- A novel adversarial patch attack, DorPatch, that can evade both certifiable and empirical defenses against adversarial patch attacks, while being physically realizable for launching real-world attacks
  - Applies ***group lasso to the patch's mask, and employs image dropout, density regularization, and structural loss*** to generate a ***fully optimized, distributed, occlusion-robust, and inconspicuous*** adversarial patch
- Comprehensive experiments
  - DorPatch can **effectively evade PatchCleanser**, the state-of-the-art certifiable defense, and empirical defenses against adversarial patch attacks
  - Moreover, DorPatch can ***make PatchCleanser certify the wrong predictions*** of the adversarially perturbed examples, ***creating a false sense of security for the users***
  - DorPatch achieves the best attack performance and perceptual quality among all adversarial patch attacks
- DorPatch poses a serious challenge to the practical applications of DNN models and urges the development of more robust defenses against such attacks