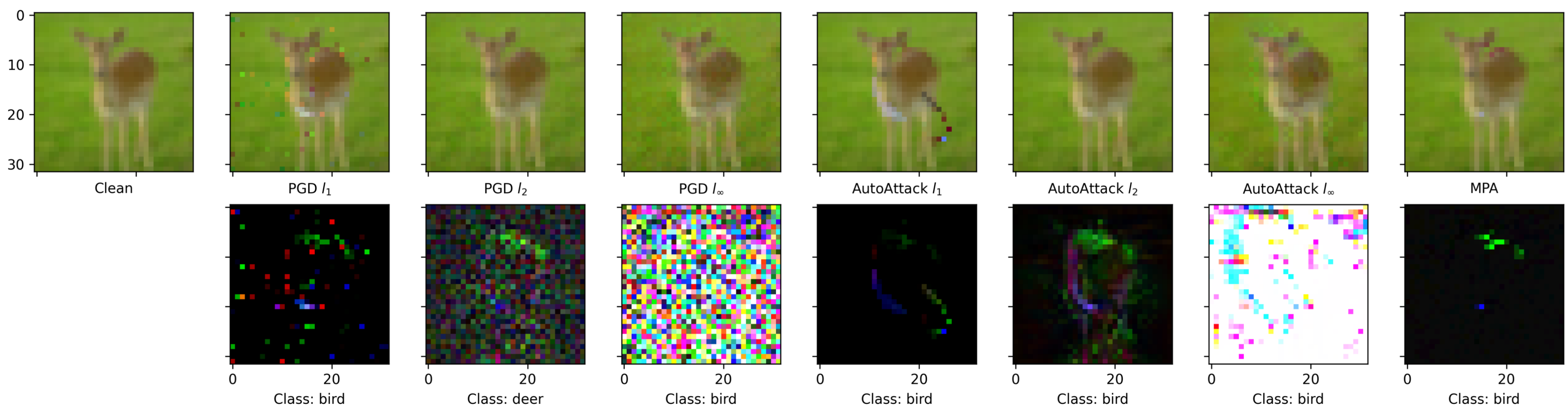
Multiple Perturbation Attack: Attack Pixelwise Under Mixed ℓp-norms For Better Adversarial Performance

Ngoc N. Tran, Anh Bui, Dinh Phung, Trung Le

VinAI Research, Hanoi, Vietnam

INTRODUCTION

- Existing Lp attacks have compromises, where L1 is strong but visible, L2 is invisible yet weak, and L∞ is a balanced tradeoff between performance and visual quality
- We can combine these different attacks by selecting the perturbations per pixel to leverage the strong suit of each to create a better adversarial attack
- Since this attack is multi-normed, it works well against novel multinorm defenses, that simultaneously guard against adversaries under different Lp norms.



ALGORITHM

- Attack different Lp norms to obtain different perturbations
- Select the best perturbations per-pixel by optimizing a low-temperature softmax mixing coefficients, then use a hardmax at the end
- Use a custom per-pixel projection operator to ensure visual quality

Algorithm 1: Combining adversarial perturbations under multiple imperceptibility criteria, with custom mixed projection operation.

Input: Adversarial image $\mathbf{x}_{\text{adv}} \in \mathbb{R}^d$, clean image $\mathbf{x} \in \mathbb{R}^d$, set of norms $\mathcal{P}$, mixing weights $\mathbf{c} \in \mathbb{R}^{d \times |\mathcal{P}|}$, maximum budgets $\{\epsilon_p | p \in \mathcal{P}\}$
Output: Projected adversarial image $\mathbf{x}_{\text{adv}} \in \mathbb{R}^d$
for $p \in \mathcal{P}$ **do**
  `// Get indices where norm-p perturbation will be used`
  $S_p \leftarrow \{i | i \in 1..d, \forall q \in \mathcal{P} : \mathbf{c}_p^i \geq \mathbf{c}_q^i\}$;
  `// Add perturbation for each norm`
  $\mathbf{x}_{\text{adv}}[S_p] \leftarrow \mathbf{x}_{\text{adv}}[S_p] + \nabla_p[S_p]$;
  `// Project each sub-image to their respective norm as in [14]`
  $\mathbf{x}_{\text{adv}}[S_p] \leftarrow \text{Proj}(\mathbf{x}_{\text{adv}}[S_p], p, \epsilon_p)$;
end
return $\mathbf{x}_{\text{adv}}$

Algorithm 2: Multiple Perturbation Attack (MPA) Algorithm.

Input: Differentiable classifier function f, clean image $\mathbf{x} \in \mathbb{R}^d$, clean label y, number of iterations n, number of mixing coefficient optimization iterations n', set of norms $\mathcal{P} = \{1, 2, \infty\}$, maximum budgets $\{\epsilon_p | p \in \mathcal{P}\}$, step sizes $\{\delta_p | p \in \mathcal{P}\}$, coefficient step size δ_c, softmax temperature t
Output: Adversarial image $\mathbf{x}_{\text{adv}} \in \mathbb{R}^d$
Initialize $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x}$;
Initialize $\mathbf{c} \in \mathbb{R}^{d \times |\mathcal{P}|}$;
for $i = 1..n$ **do**
  $\nabla \leftarrow \dfrac{\partial \mathcal{L}(f(\mathbf{x}_{\text{adv}}), y)}{\partial \mathbf{x}_{\text{adv}}}$;
  for $p \in \mathcal{P}$ **do**
    `// Follow the steepest ascending direction as described in [14]`
    $\nabla_p \leftarrow \text{NormalizedSteepestAscent}(\nabla, p, \delta_p)$;
  end
  for $j = 1..n'$ **do**
    `// Use σ = softmax to choose which gradient to be used per pixel`
    $\mathbf{c} \leftarrow \mathbf{c} + \delta_c \dfrac{\partial \mathcal{L}(f\left(\mathbf{x}_{\text{adv}} + (\sigma(\mathbf{c}/\tau) \odot \begin{bmatrix}\nabla_{p_1} & \dots & \nabla_{p_{|\mathcal{P}|}}\end{bmatrix}^\top)\mathbb{1}^{|\mathcal{P}|}\right), y)}{\partial \mathbf{c}}$;
  end
  `// Use hard decision to choose gradient, then custom project`
  $\mathbf{x}_{\text{adv}} \leftarrow \text{Combine}(\mathbf{x}_{\text{adv}}, \mathbf{x}_0, \mathcal{P}, \mathbf{c}, \{\nabla_p | p \in \mathcal{P}\})$;
  if $f(\mathbf{x}_{\text{adv}}) \neq \mathbf{y}$ **then**
    `// Stop early if attack succeeds`
    return $\mathbf{x}_{\text{adv}}$
  end
end
return $\mathbf{x}_{\text{adv}}$

DISCUSSION

- Intuition behind strength vs. visual quality tradeoff of AutoAttack: L1 perturbation is sensitive to *target* class; L2 is sensitive to the *source* class; L∞ is just random noise.
- By selecting the best perturbation per-pixel, we can harness the best of all worlds
- Our method tradeoff is in running time, since we have to backprop at every iteration.
- For offensive security, MPA may hold ethical implications. Regardless, we hope that our research will to more robust defenses against stronger and diverse attacks.

HYPERPARAMETER SELECTION

- We reuse the mixing weights after each iterations
- Softmax temperature is set to 0.01
- 17 attack iterations yield the best result

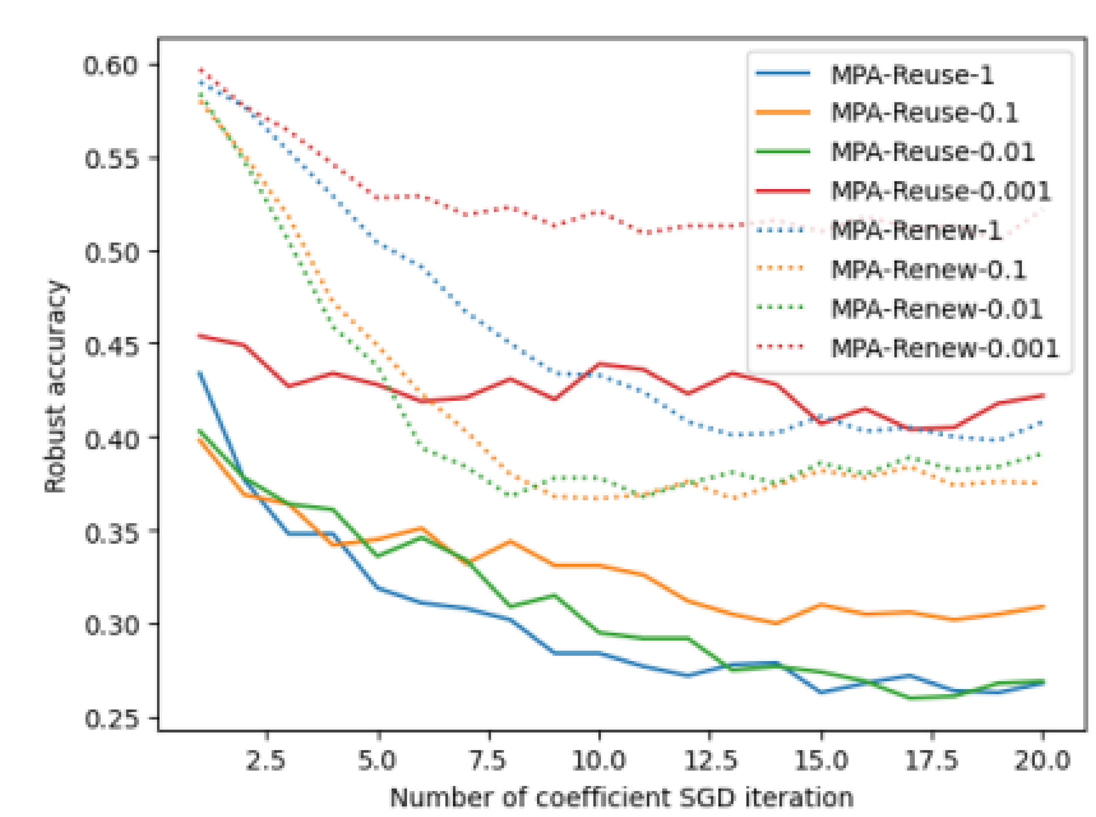


Figure 2. Robust accuracy of models under MPA across different attack hyperparameters.

EXPERIMENTAL RESULTS

We compare our method with standard PGD and AutoAttack ensemble.

- For ImageNet, our method outperforms other attacks significantly
- For CIFAR, we outperform all other attacks except AA-L1, while not degrading image quality obviously.
- For multinorm defenses (Maini *et. al*), our attack also yield noticeably lower robust accuracy.

Table 1. Robust accuracy for adversarial-trained models under different attacks on ImageNet (lower is better)

Model	Clean	Projected Gradient Descent			AutoAttack			MPA
		PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	AA-ℓ_1	AA-ℓ_2	AA-ℓ_∞	
Debenedetti *et. al*, 2022 [7]	79.98%	77.96%	78.78%	69.02%	71.32%	77.38%	55.40%	**53.46%**
Salman *et. al*, 2020 [17]	74.82%	69.64%	72.68%	62.72%	50.64%	69.66%	46.96%	**39.36%**
Engstrom *et. al*, 2019 [8]	69.96%	65.28%	67.98%	55.90%	44.36%	65.00%	37.90%	**31.70%**

Table 2. Robust accuracy for adversarial-trained models under different attacks on CIFAR-10 (lower is better).

Model	Clean	Projected Gradient Descent			AutoAttack			MPA
		PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	AA-ℓ_1	AA-ℓ_2	AA-ℓ_∞	
Rebuffi *et. al*, 2021 [16]	92.9%	41.2%	74.9%	72.1%	**10.7%**	68.8%	67.3%	20.7%
Gowal *et. al*, 2021 [11]	89.5%	39.8%	71.3%	70.8%	**8.6%**	64.1%	67.6%	21.3%
Gowal *et. al*, 2020 [10]	90.7%	39.9%	73.1%	70.7%	**7.1%**	66.6%	67.0%	20.7%
Maini *et. al*, 2020 [14]	83.5%	62.8%	68.4%	49.4%	49.0%	65.9%	44.1%	**26.0%**

Table 4. Robust accuracy for adversarial-trained models under different attacks on CIFAR-100 (lower is better).

Model	Clean	Projected Gradient Descent			AutoAttack			MPA
		PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	AA-ℓ_1	AA-ℓ_2	AA-ℓ_∞	
Gowal *et. al*, 2020 [10]	69.3%	16.7%	45.8%	41.1%	**4.9%**	39.5%	35.7%	10.3%
Debenedetti *et. al*, 2022 [7]	70.1%	27.8%	51.6%	39.4%	**11.9%**	46.0%	35.1%	14.1%
Rebuffi *et. al*, 2021 [16]	62.3%	20.3%	43.7%	38.4%	**7.3%**	39.1%	34.3%	10.8%
Maini *et. al*, 2020 [14]	56.6%	38.9%	42.1%	25.8%	27.4%	39.0%	22.2%	**14.0%**