The 18th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2023)

Boost Off/On-Manifold Adversarial Robustness for Deep Learning with Latent Representation Mixup

Mengdie Huang[1], Yi Xie[1], Xiaofeng Chen[1], Jin Li[2], Changyu Dong[3], Zheli Liu[4], Willy Susilo[5]

[1] Xidian University

[2] Guangzhou University

[3] Newcastle University

[4] Nankai University

[5] University of Wollongong

Overview

Contents

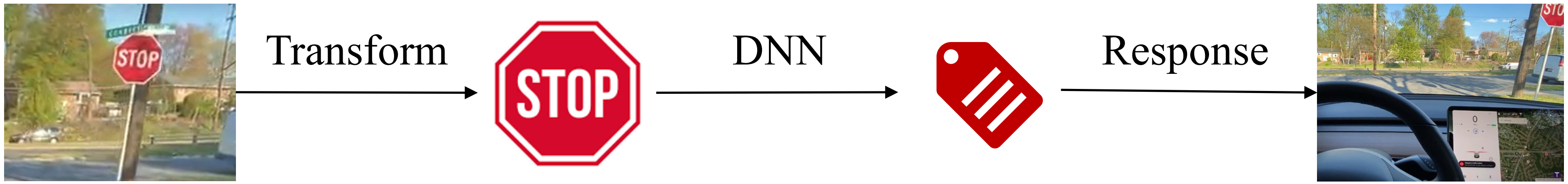
- Background
- Problem
- Solution
- Evaluation
- Conclusion

Keywords

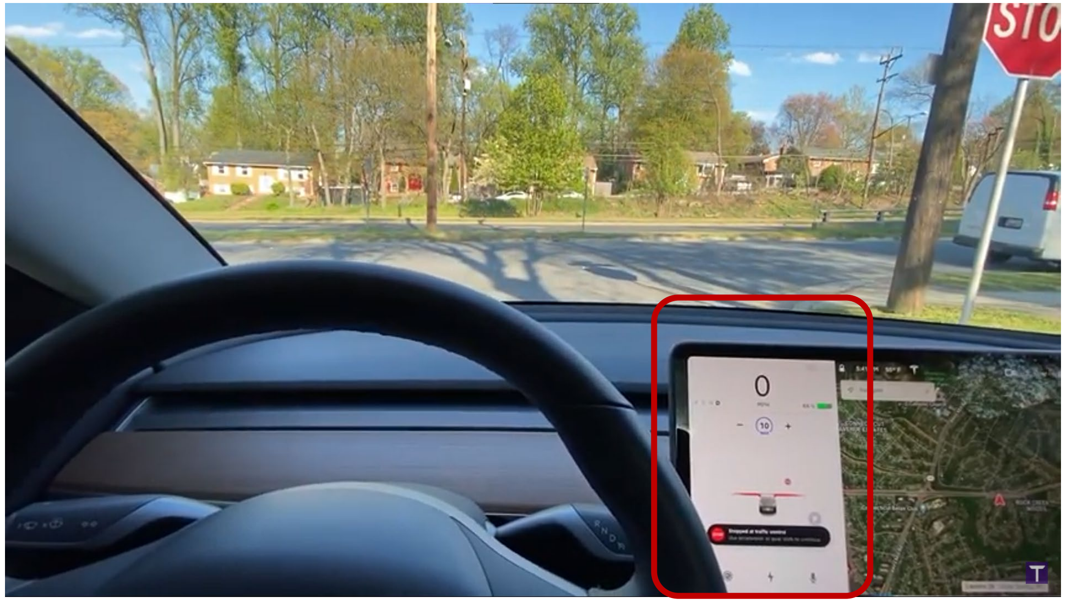
- Deep Neural Network
- Off-manifold Adversarial Attack
- On-manifold Adversarial Attack
- Mixup Training
- Adversarial Robustness
- Representation Learning

Practical Case - Auto Driving

- Traffic sign must be read correctly



- Normal looking Stop sign can be ignored

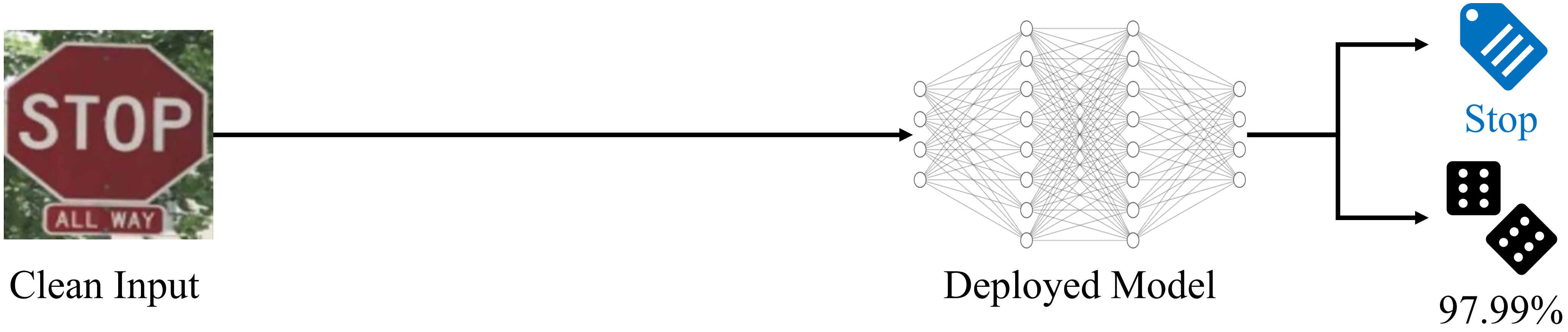


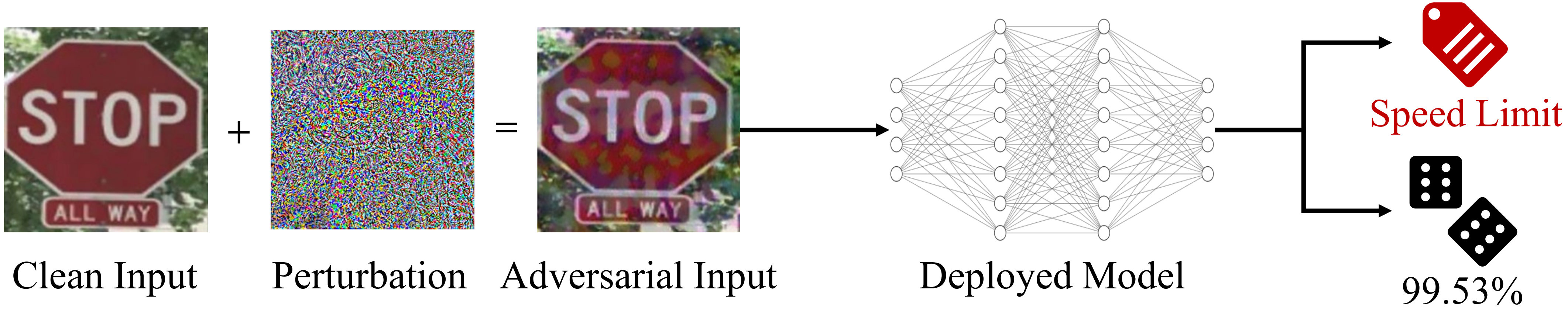
Autopilot action: Stop

Autopilot action: Speed limit

Threats to Deep Neural Networks (DNNs)

- Adversarial Example



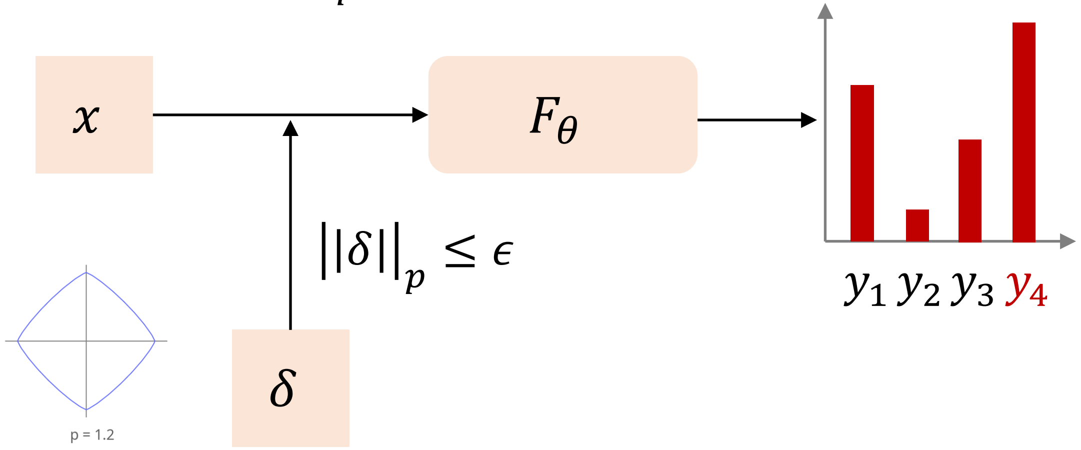


L_p Threats to Deep Neural Networks (DNNs)

Off-manifold Adversarial (Example) Attack

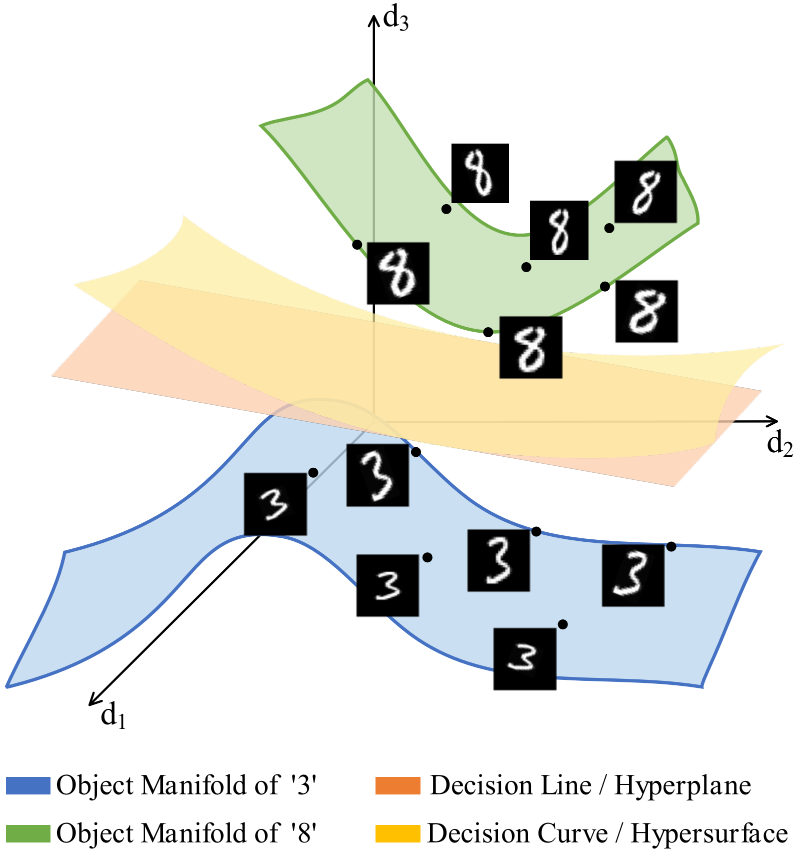
- Aka:
  - Regular adversarial attack
  - Input-space adversarial attack
  - Pixel-space adversarial attack
- Optimization objective

$$\max_{||\delta||_p \leq \epsilon} \mathcal{L}(F_\theta(x+\delta), y_{true})$$



- FGSM, PGD, JSMA, DeepFool, CW, AutoAttack

Object (Class) Manifold



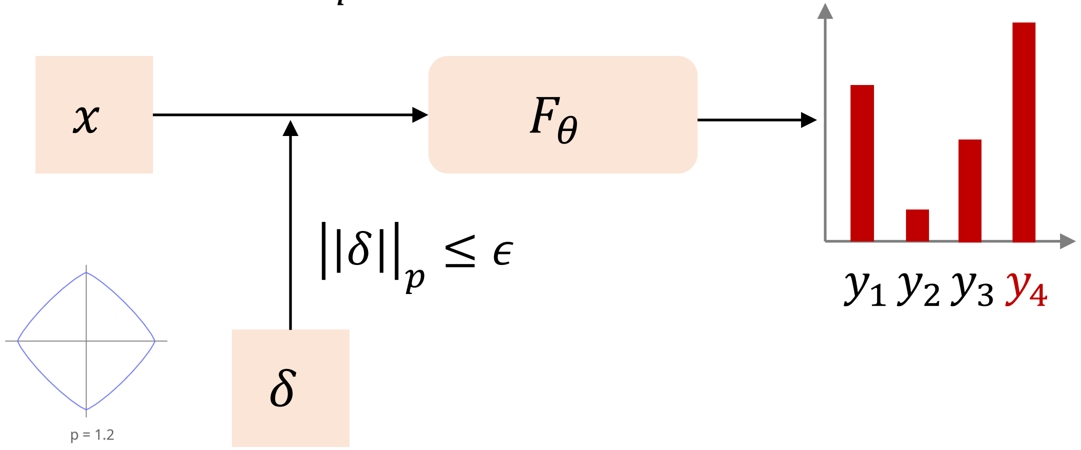
Input space: 28x28 pixels→728 dimensions

L_p Threats to Deep Neural Networks (DNNs)

Off-manifold Adversarial (Example) Attack

- Aka:
  - Regular adversarial attack
  - Input-space adversarial attack
  - Pixel-space adversarial attack
- Optimization objective

$$\max_{||\delta||_p \leq \epsilon} \mathcal{L}(F_\theta(x+\delta), y_{true})$$

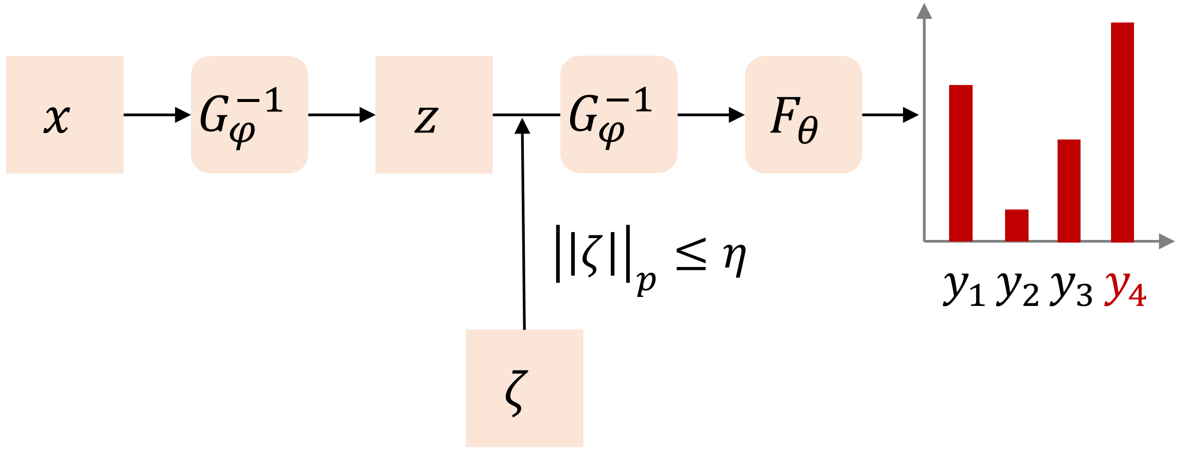


- FGSM, PGD, JSMA, DeepFool, CW, AutoAttack

On-manifold Adversarial (Example) Attack

- Aka:
  - latent-space adversarial attack
- Optimization objective

$$\max_{||\zeta||_p \leq \eta} \mathcal{L}(F_\theta(G_\varphi(z+\zeta)), y_{true})$$



- OM-FGSM, OM-PGD

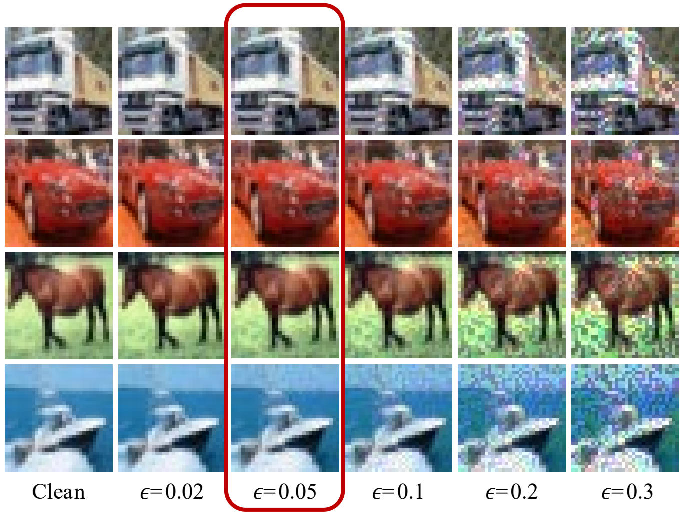
L_p Threats to Deep Neural Networks (DNNs)

Off-manifold Adversarial (Example) Attack

- Aka:
  - Regular adversarial attack
  - Input-space adversarial attack
  - Pixel-space adversarial attack
- Optimization objective

$$\max_{||\delta||_p \leq \epsilon} \mathcal{L}(F_\theta(x+\delta), y_{true})$$

PGD
CIFAR-10

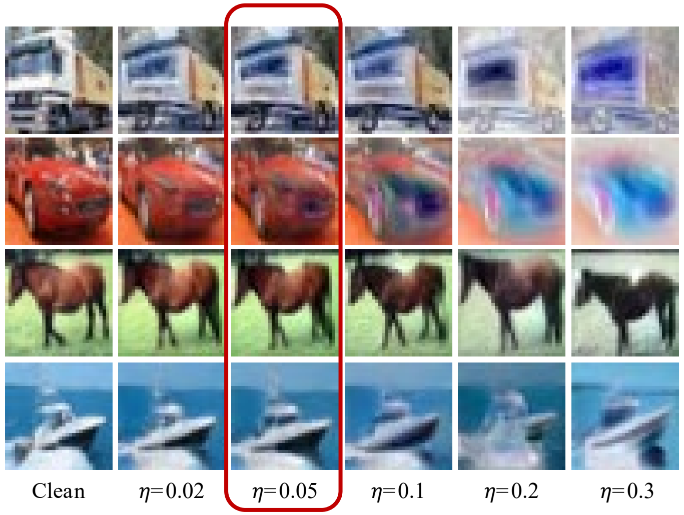


On-manifold Adversarial (Example) Attack

- Aka:
  - latent-space adversarial attack
- Optimization objective

$$\max_{||\zeta||_p \leq \eta} \mathcal{L}(F_\theta(G_\varphi(z+\zeta)), y_{true})$$

OM-PGD
CIFAR-10



L_p Threats to Deep Neural Networks (DNNs)

Off-manifold Adversarial (Example) Attack

- Aka:
  - Regular adversarial attack
  - Input-space adversarial attack
  - Pixel-space adversarial attack
- Optimization objective

$$\max_{||\delta||_p \leq \epsilon} \mathcal{L}(F_\theta(x+\delta), y_{true})$$

PGD
SVHN

Clean ϵ=0.02 ϵ=0.05 ϵ=0.1 ϵ=0.2 ϵ=0.3

On-manifold Adversarial (Example) Attack

- Aka:
  - latent-space adversarial attack
- Optimization objective

$$\max_{||\zeta||_p \leq \eta} \mathcal{L}(F_\theta(G_\varphi(z+\zeta)), y_{true})$$

OM-PGD
SVHN

Clean η=0.02 η=0.05 η=0.1 η=0.2 η=0.3

Invalid OM-PGD

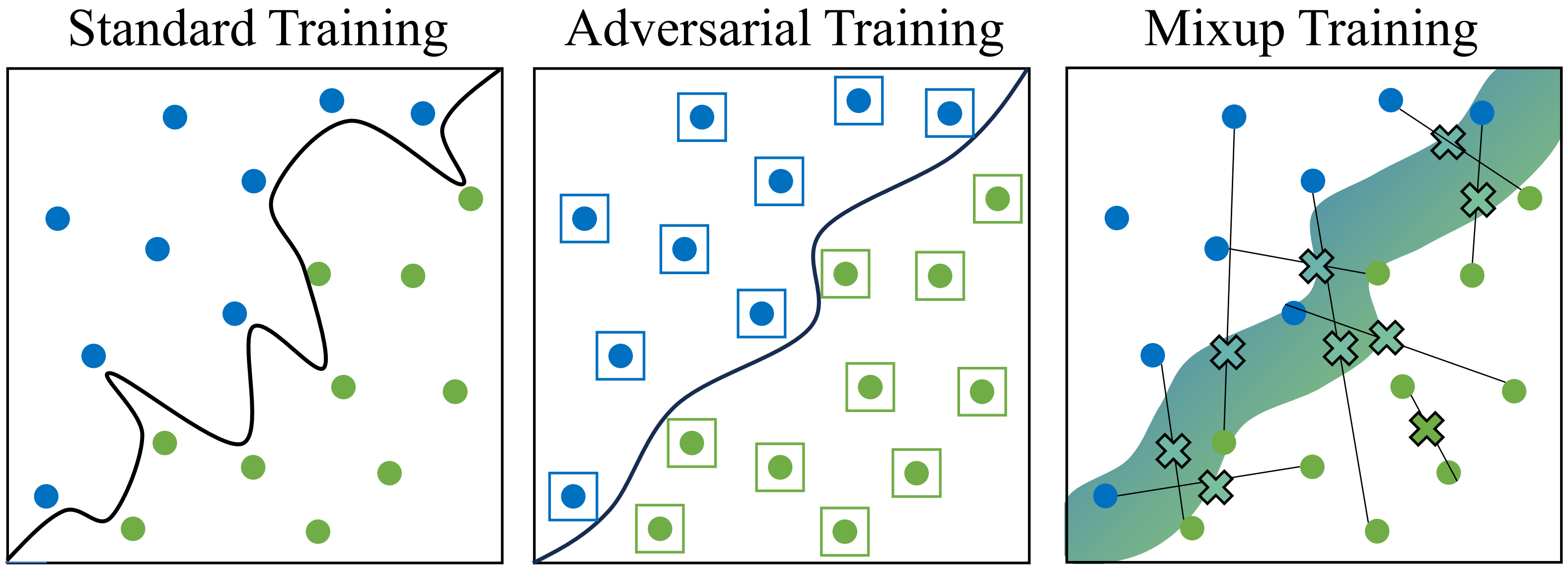
Defense Methods Focused on Improving Adversarial Robustness of DNN

Against Off-manifold Adversarial Attack

- Adversarial Training (AT): $(x + \delta, y_{true})$
  - Input-space AT
    - FGSM-AT
    - PGD-AT
- Mixup Training: $(\alpha x_1 + (1-\alpha)x_2, \alpha y_1 + (1-\alpha)y_2)$
  - Input-space Mixup
    - InputMixup
    - CutMix
    - PuzzleMixup
  - Hidden-space Mixup
    - ManifoldMixup
    - PatchUp

Against On-manifold Adversarial Attack

- On-Manifold Adversarial Training (OMAT):
  - Latent-space AT $(G_\varphi(z + \zeta), y_{true})$
    - Dual Manifold-AT (DMAT)
      - FGSM-AT + OM-FGSM-AT
      - PGD-AT + OM-PGD-AT



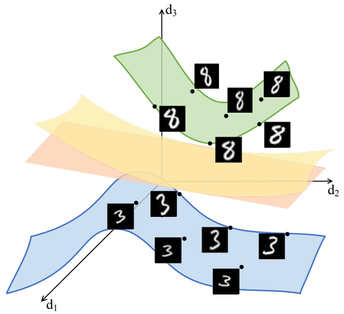
Improve Off/On-Manifold Adversarial Robustness

- **Issue 1:**
  - AT defenses require the defender to have some knowledge of the attack in advance, so that the defender can actively generate adversarial examples for training.

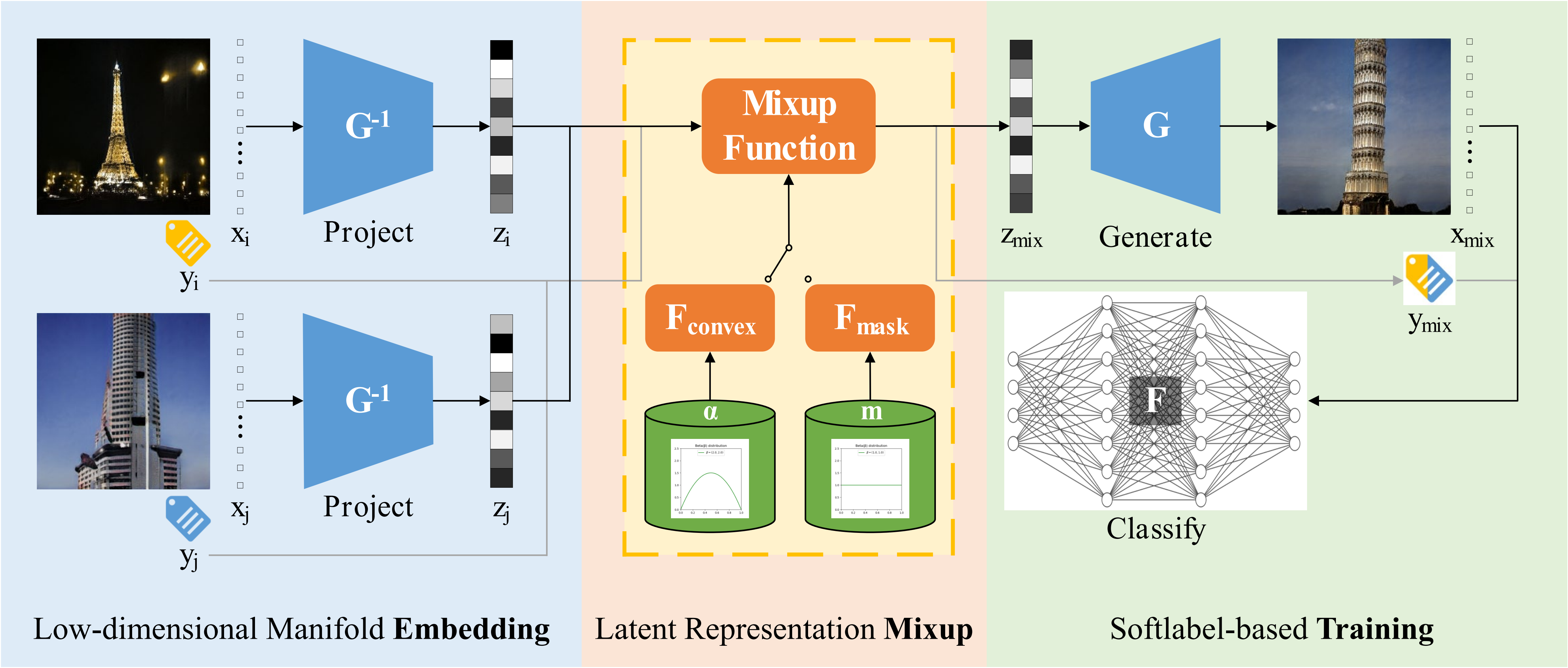
- **Issue 2:**
  - All of existing Mixup defenses focused on improving robustness to off-manifold adversarial attacks but ignores on-manifold adversarial attacks and non-L_p attacks.

- **Problem to be solved:**
  - Assume the attack knowledge is completely unknown, defender try to enhance the robustness against the off-manifold and on-manifold adversarial attacks at the same time.

- **Idea:**
  - Construct interpolation samples in the latent space where embedded with the approximately exact manifold.
    - Off-manifold interpolation points → off-manifold robustness
    - On-manifold interpolation points → on-manifold robustness
  - Use the mixed label to supervise the learning, so that the model is encouraged to assign class probabilities based on the interpolated proportion.



Framework of Proposed LarepMixup Training



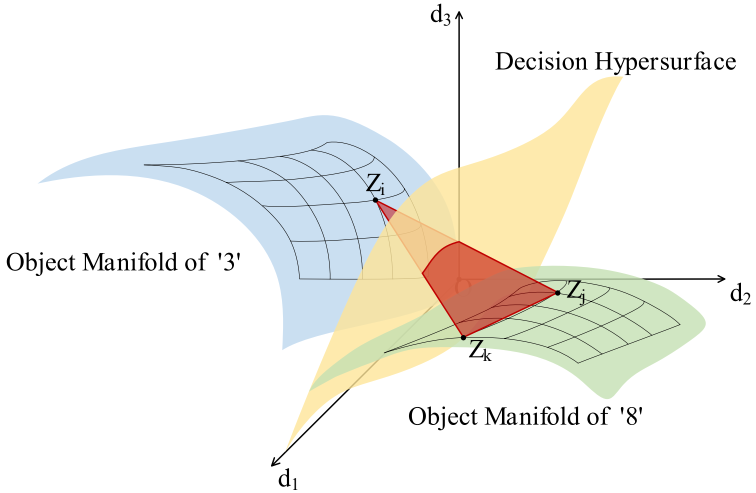
Proposed Multi-mode Manifold Interpolation Strategy

Convex Combination-based Interpolation

- Mixed Sample $z_{mix} = \alpha_1 z_1 + \cdots + \alpha_k z_k$
- Mixed Label $y_{mix} = \alpha_1 y_1 + \cdots + \alpha_k y_k$
- Coefficient vector

$$\alpha \in A \coloneqq \{R^k, \alpha_i \in [0,1], \sum_{i=0}^{k} \alpha_i = 1\}$$

  - Case $k = 2$, sample α from $Beta(\beta)$.
  - Case $k > 2$, sample α from $Dirichlet(\gamma)$.

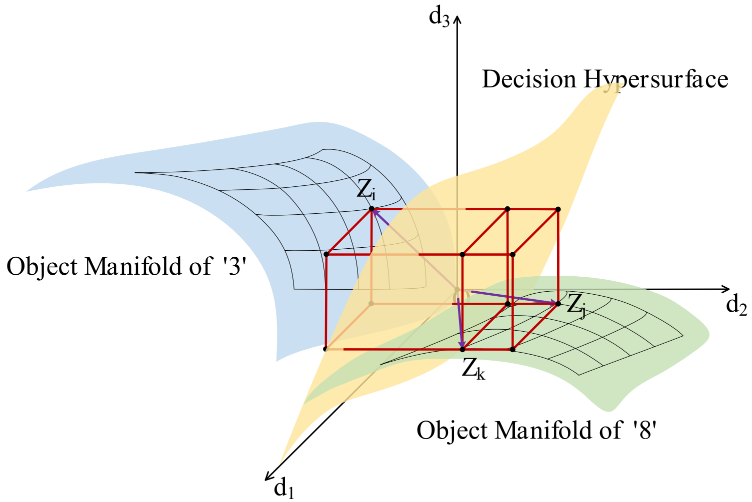


Binary Mask Combination-based Interpolation

- Mixed Sample $z_{mix} = m_1 z_1 \odot \ldots \odot m_k z_k$
- Mixed Label $y_{mix} = \lambda_1 y_1 + \cdots + \lambda_k y_k$
- Coefficient vector

$$m_i \in B \coloneqq \{0,1\}^n, \sum_{i=0}^{k} m_i = 1_B$$

$$\lambda_i = \frac{Num_{m_i=1}}{n}$$



  - Case $k = 2$, sample m_1 from n-fold $Bernoulli(p)$, n is the dimension of z.
  - Case $k > 2$, sample m_2 from q-fold $Bernoulli(p)$, q is the number of non-zero elements in the vector $1_B - m_1$.
  - Sample p from $Uniform(0,1)$.

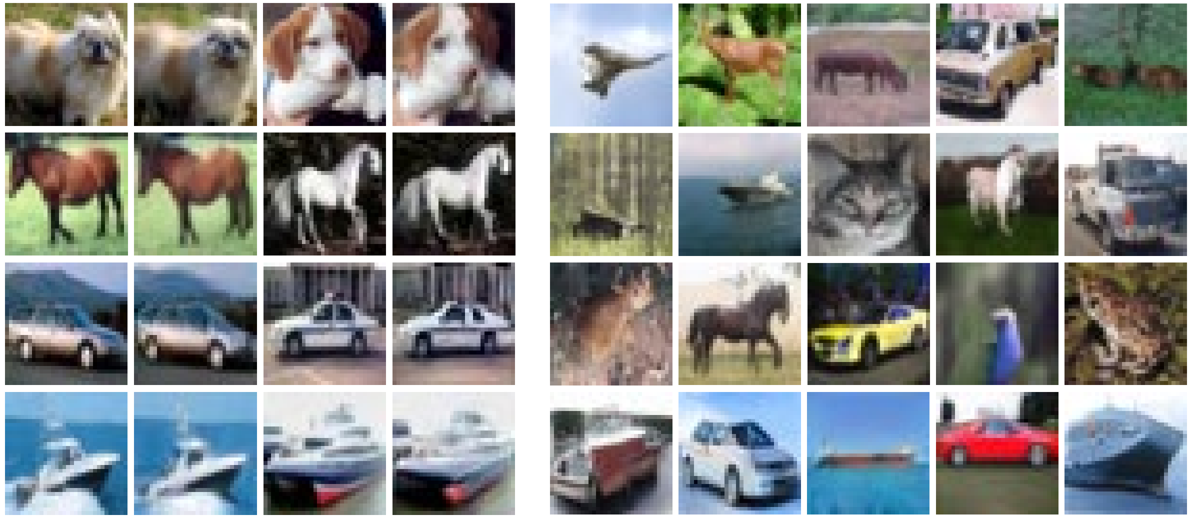
Embedding from Input Space to Latent Representation Space

$$(x, y_{true}) \rightarrow (z, y_{true})$$

- Embedding network: trained styleGAN
- Embedding algorithm:
  - Sample w randomly from $Normal(0,1)$
  - $t = 0$
  - $z_t = F_{map}(w)$
  - While $t < T$ do
    - $G(z_t)$
    - $z_{t+1} = z_t - \eta(\nabla_{z_t} L_{styleGAN}(G(z_t), x))$
    - $t = t + 1$
  - End While
  - $z = z_t$

Visualization

- Indirectly demonstrates the quality of the learned data manifold, composed of several object manifolds.
- $G(z)$ from D_{test}: Data distribution supported by the learned manifold is close to the true data distribution.
- Unseen $G(z)$ by sampling z with random seeds.



x in D_{train} G(z) x in D_{test} G(z) G(z)

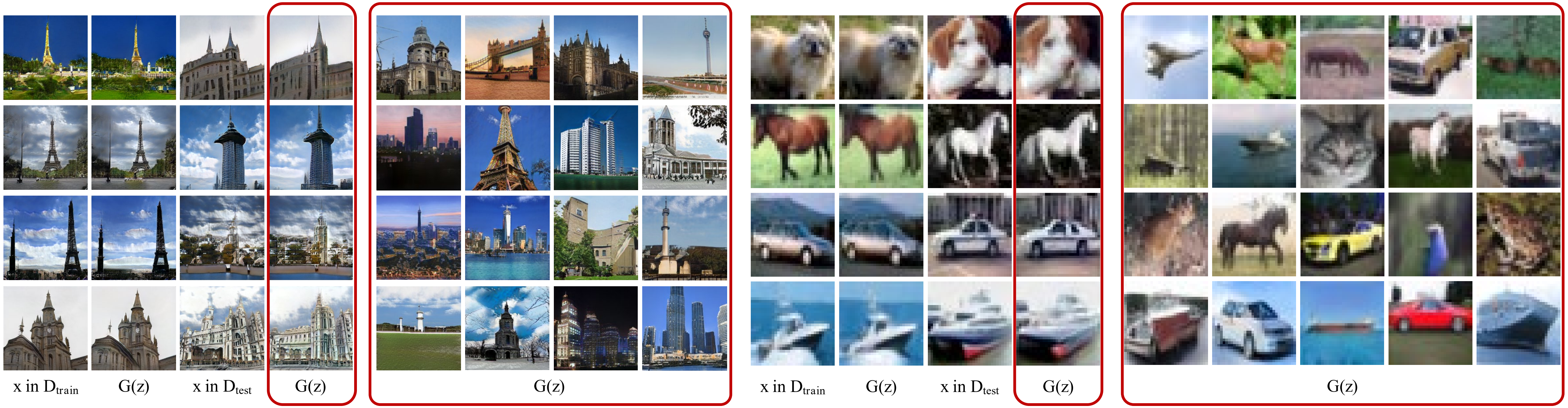
Embedding from Input Space to Latent Representation Space

$$(x, y_{true}) \rightarrow (z, y_{true})$$

- Embedding network: trained styleGAN
- Embedding algorithm:
  - ➢ Sample w randomly from $Normal(0,1)$
  - ➢ $t = 0$
  - ➢ $z_t = F_{map}(w)$

Visualization

- Indirectly demonstrates the quality of the learned data manifold, composed of several object manifolds.
- $G(z)$ from D_{test}: Data distribution supported by the learned manifold is close to the true data distribution.
- Unseen $G(z)$ by sampling z with random seeds.



Mapping from Latent Representation Space to Input Space

$$(z_{mix}, y_{mix}) \rightarrow (x_{mix}, y_{mix})$$

- Generate network: trained styleGAN
- Generate Function: $x_{mix} = G(z_{mix})$
- Dual / Ternary LarepMixup
  - Convex Combination
  - Binary Mask combination

Visualization

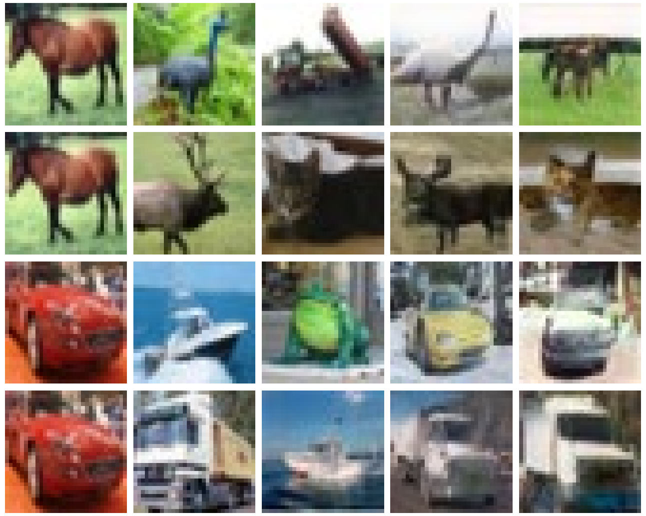
- For convex mixup, coefficient α can take a value from the continuous range, $[0, 1]$.
- For binary mask mixup, coefficient m is discrete and can only be taken from the binary set $\{0, 1\}^n$.

✓ Convex mixup: mixed examples show more smooth mixed characteristics between source features.

✓ Binary mask mixup: mixed examples show fewer transitions between source features.

x_i x_j convex mix

x_i x_j convex mix mask mix



x_i x_j x_k convex mix mask mix

Fine Tuning Vanilla DNN with Mixed Samples and Mixed Labels

Standard Train

- We train the DNN on the original clean trainset

$$D_{ori\_tra} = \{(x, y_{true})\}$$

- One-hot label-based Cross entropy loss
  - One hot coding $y_{true} \in \{0,1\}^C$

$$L(f(x), y_{true}) = -\sum_{i=1}^{C} y_i \log(p_i)$$

- Optimization objective

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D_{ori\_tra}} L(f_\theta(x, y))$$

Full Fine Tuning

- We retrain the vanilla DNN on the augmented dataset

$$D_{fin\_tun} = D_{mix} \cup D_{ori\_tra}$$

- Soft label-based cross entropy loss

$$\begin{aligned} &L_{soft}(f(x), y_{mix}) \\ &= L_{soft}(f(x), \alpha_1 y_1 + \cdots + \alpha_k y_k) \\ &= \alpha_1 L(f(x), y_1) + \cdots + \alpha_k L(f(x), y_k) \end{aligned}$$

- Optimization objective

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D_{fin\_tun}} L_{soft}(f_\theta(x, y))$$

Experimental Setup

Datasets and Models

- Environment
  - PyTorch 1.8.1, CUDA V11.1.74
  - NVIDIA GV102 GPU
  - Adversarial Robustness Toolbox, advertorch
- Dataset
  - CIFAR-10, SVHN
  - ImageNet-Mixed10 (a subset of 10 categories)
- Model
  - Convolutional block-based: Alexnet and VGG
  - Residual block-based: ResNet, DenseNet, PreActResNet, and WideResNet
  - Inception block-based: GoogLeNet

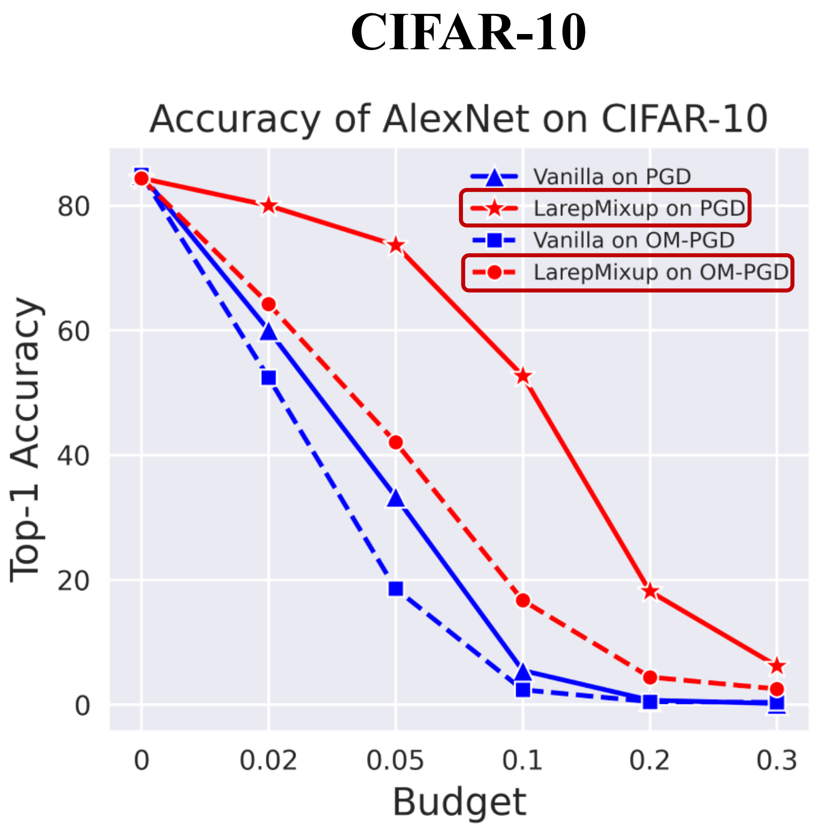
Baselines

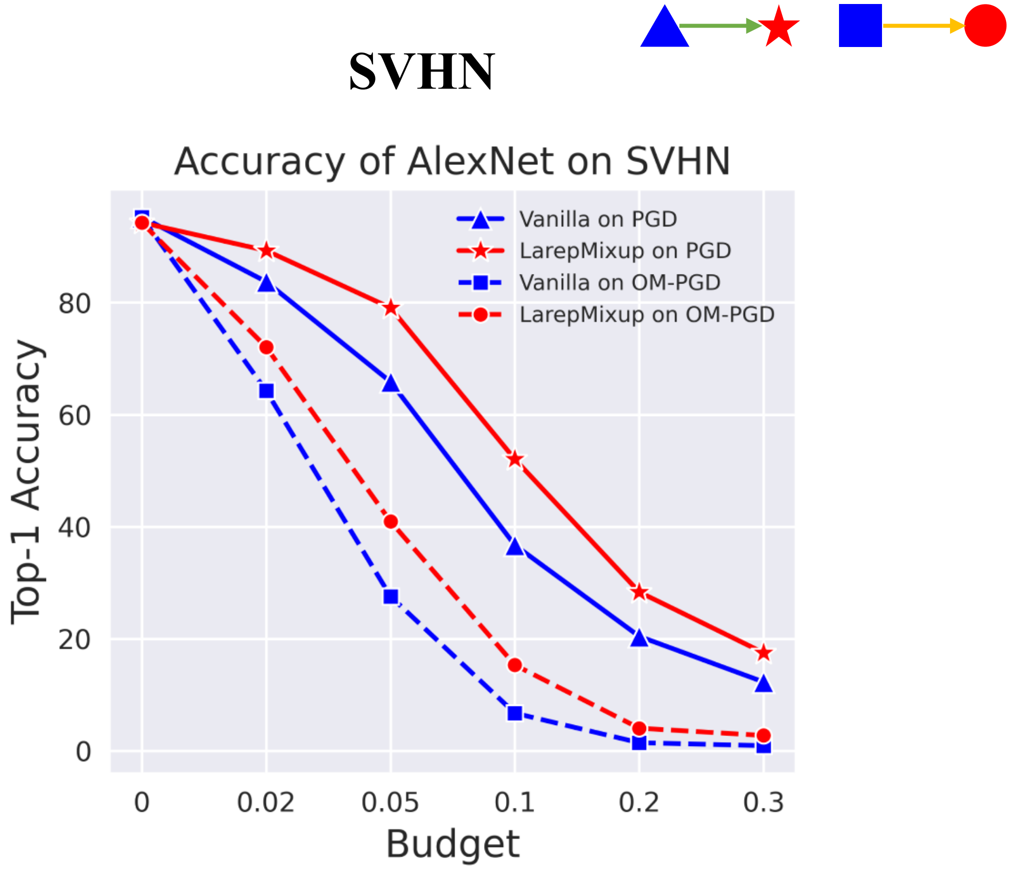
- Attack methods
  - Off-manifold attack: FGSM, PGD, AutoAttack, DeepFool, CW
  - On-manifold attack: OM-FGSM, OM-PGD
- Defense methods
  - Mixup training methods (5)
  - Adversarial training methods (2)

Method	Attack Surfaces	Attack Algorithm	Augmentation
PGD-AT[36]	Off-manifold	Known	Input Space
PGD-DMAT[35]	Off/On-manifold	Known	Input/Latent Space
InputMixup[56]	Off-manifold	Unknown	Input Space
CutMix[54]	Off-manifold	Unknown	Input Space
PuzzleMixup[29]	Off-manifold	Unknown	Input Space
ManifoldMixup[52]	Off-manifold	Unknown	Latent Space
PatchUp[14]	Off-manifold	Unknown	Latent Space
LarepMixup(Ours)	Off/On-manifold	Unknown	Latent Space

Exp 1: Robustness against Different L_p Adversarial Attack Budgets: $||\delta||_p \leq \epsilon$ and $||\zeta||_p \leq \eta$

❖ Exp Setup: Off-manifold perturbation δ budget $\epsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.02. On-manifold perturbation ζ budget $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.005.



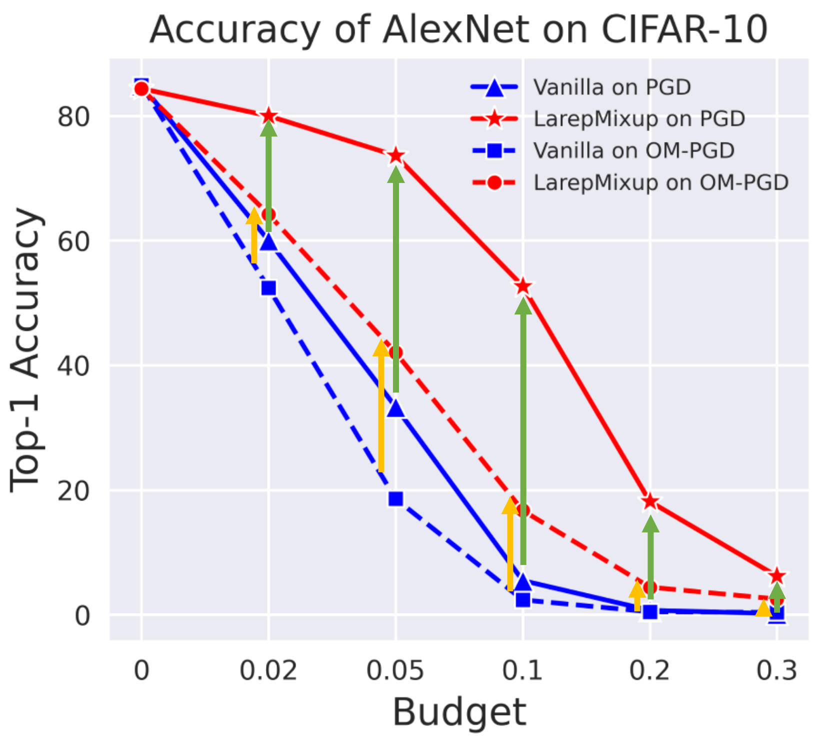


Exp 1: Robustness against Different L_p Adversarial Attack Budgets: $||\delta||_p \le \epsilon$ and $||\zeta||_p \le \eta$

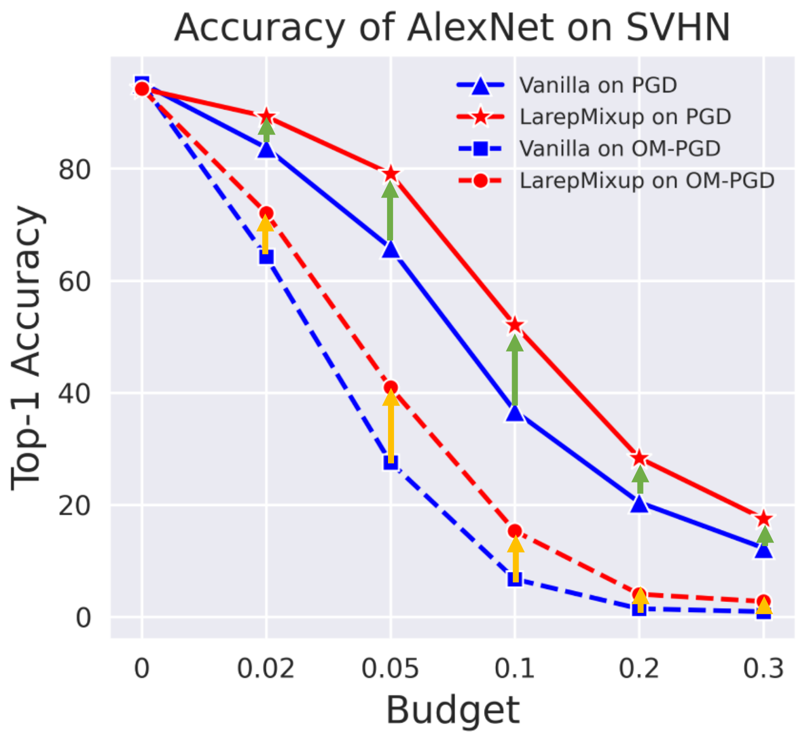
❖ Exp Setup: Off-manifold perturbation δ budget $\epsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.02. On-manifold perturbation ζ budget $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.005.

- Finding 1: Against PGD and OM-PGD attacks with five strengths, LarepMixup trained AlexNet models always performs better than standard trained models.

CIFAR-10



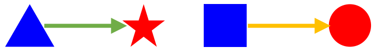
SVHN



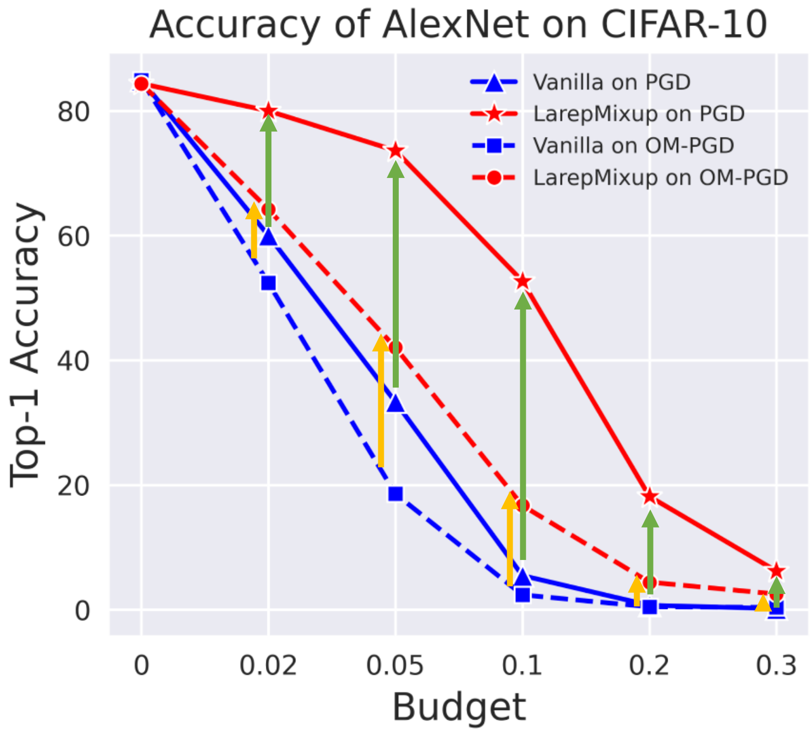
Exp 1: Robustness against Different L_p Adversarial Attack Budgets: $||\delta||_p \leq \epsilon$ and $||\zeta||_p \leq \eta$

❖ Exp Setup: Off-manifold perturbation δ budget $\epsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.02. On-manifold perturbation ζ budget $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.005.

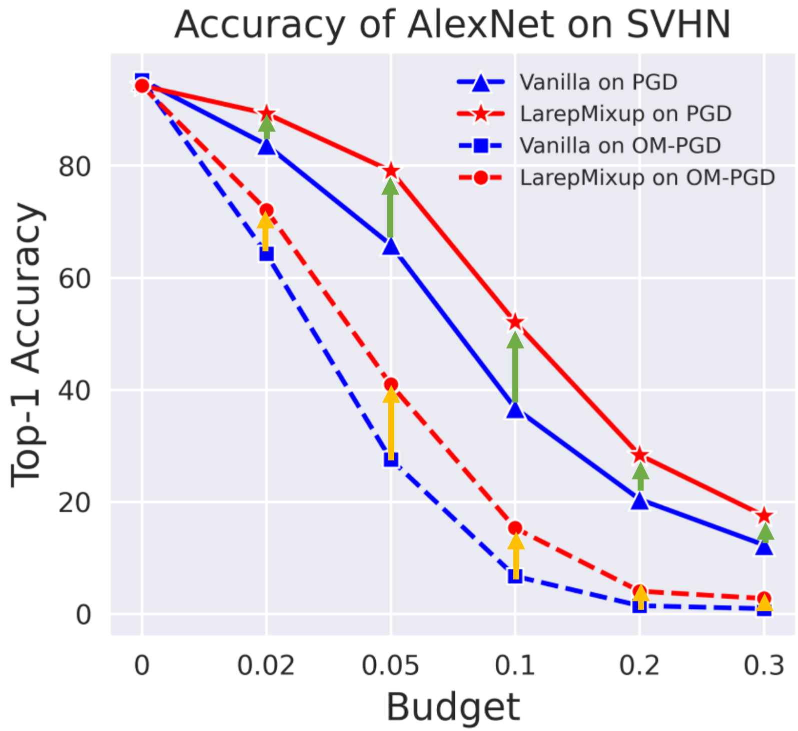
- Finding 2: The model has the best defense against attacks with medium budgets. For PGD and OM-PGD attacks, the robustness against $\epsilon = 0.1$ and $\eta = 0.05$ increase most, respectively.



CIFAR-10



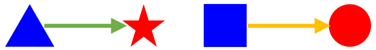
SVHN



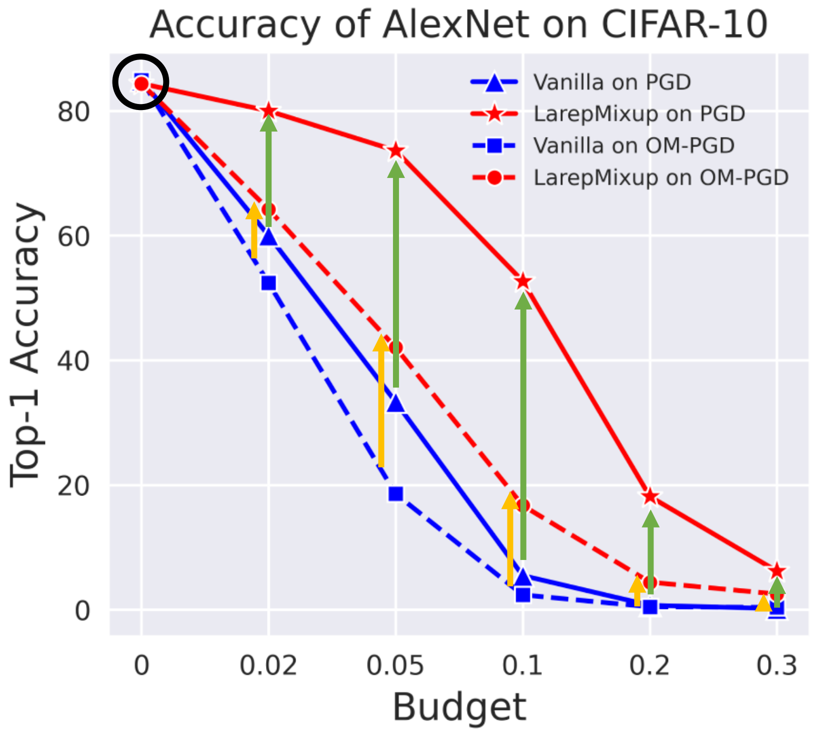
Exp 1: Robustness against Different L_p Adversarial Attack Budgets: $||\delta||_p \leq \epsilon$ and $||\zeta||_p \leq \eta$

- Exp Setup: Off-manifold perturbation δ budget $\epsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.02. On-manifold perturbation ζ budget $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.005.

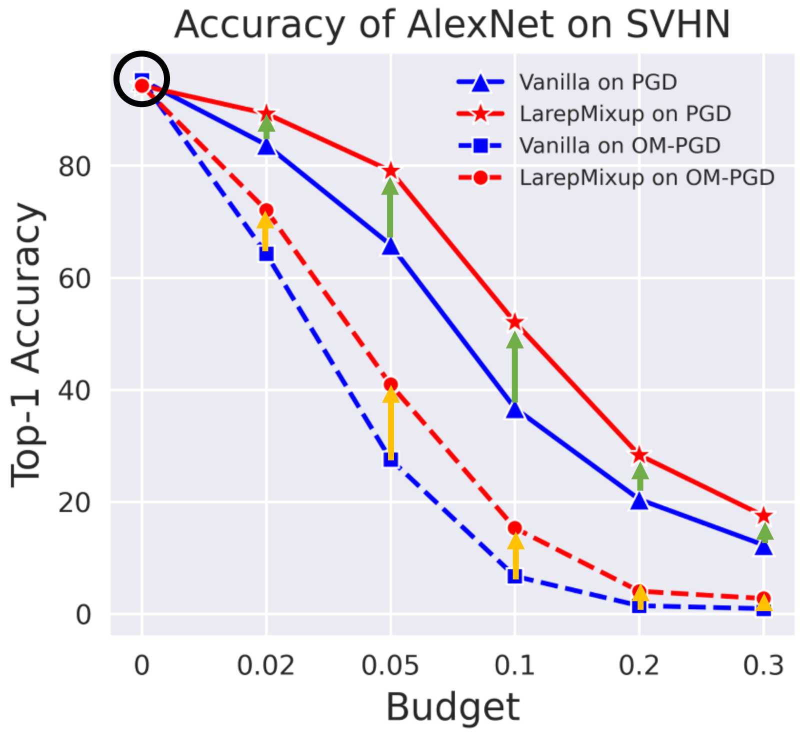
- Finding 3: The model after LarepMixup training have very similar accuracy performance on clean examples to that before training.



CIFAR-10



SVHN

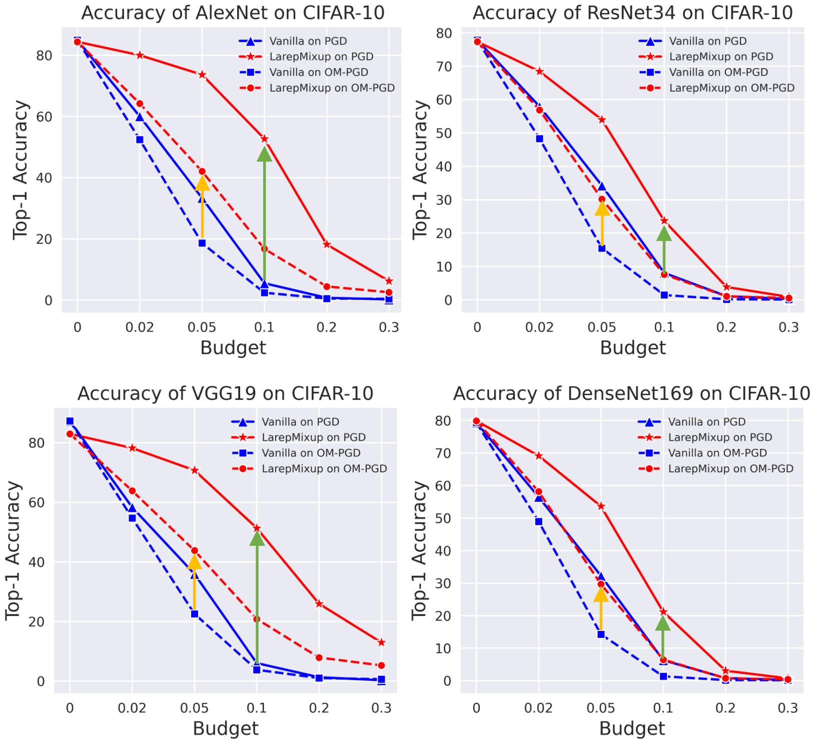


Exp 1: Robustness against Different L_p Adversarial Attack Budgets: $||\delta||_p \le \epsilon$ and $||\zeta||_p \le \eta$

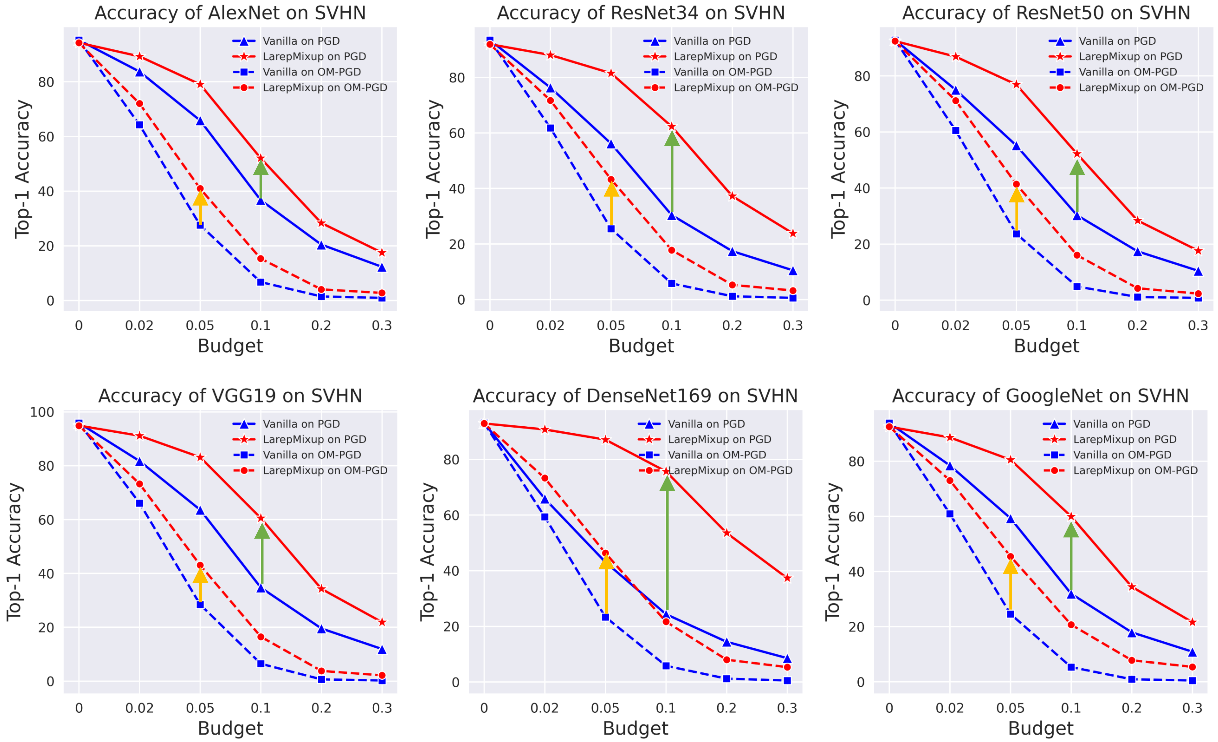
❖ Exp Setup: Off-manifold perturbation δ budget $\epsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.02. On-manifold perturbation ζ budget $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.3\}$, single step budget is 0.005.

- Finding 4 : On other models (VGG19, ResNet34, DenseNet169, ResNet50, GoogleNet), conclusions from observations 1/2/3 hold true.

CIFAR-10



SVHN



Exp 2: Comparison with Existing Mixup Training

- Exp Setup: Run six times, mean and standard deviation, 40 epochs, α from $Beta(\beta = (1.0, 1.0))$, budget 0.05.
- Finding 1: Against off-manifold attacks on CIFAR-10, LarepMixup also perform better than others on robust accuracy and clean accuracy.

Table 2: Accuracy (%) of CIFAR-10 classification models on off/on-manifold adversarial examples

PreActResNet18

Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	Known Attacker	Modify Network
Vanilla	**87.37±0.00**	32.07±0.00	28.93±0.00	7.59±0.00	10.36±0.00	2.60±0.00		
InputMixup[56]	84.48±1.45	63.58±3.36	68.12±3.46	56.63±10.20	37.97±2.58	41.11±2.10	✗	✗
CutMix[54]	82.14±3.00	65.51±1.03	69.67±1.34	<u>64.41±3.55</u>	36.79±2.60	39.74±3.10	✗	✗
PuzzleMixup[29]	83.11±1.64	65.73±2.46	70.35±2.60	64.03±6.06	38.86±1.53	41.83±1.74	✗	✗
ManifoldMixup[52]	71.10±4.17	49.26±1.34	52.49±1.91	44.08±1.60	25.33±2.76	27.19±2.53	✗	✓
PatchUp[14]	72.02±4.10	51.35±2.13	55.91±2.29	44.61±2.56	28.81±3.35	30.94±3.13	✗	✓
Ours-Convex	84.02±1.77	**68.86±2.88**	**72.65±3.59**	**66.98±5.93**	<u>39.03±2.16</u>	<u>42.03±2.31</u>	✗	✗
Ours-Mask	<u>84.60±1.27</u>	<u>66.56±1.50</u>	<u>71.22±1.93</u>	63.69±4.61	**39.27±2.97**	**42.54±2.74**	✗	✗

PreActResNet34

Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	Known Attacker	Modify Network
Vanilla	**83.57±0.00**	31.37±0.00	25.71±0.00	5.27±0.00	12.27±0.00	1.89±0.00		
InputMixup[56]	68.42±7.38	62.19±4.22	63.84±4.98	63.79±4.99	26.36±4.07	29.77±4.16	✗	✗
CutMix[54]	71.21±6.16	62.45±2.71	64.61±3.50	64.30±3.16	28.88±2.07	32.12±2.38	✗	✗
PuzzleMixup[29]	67.06±7.62	60.89±4.99	62.55±5.76	62.66±5.84	25.89±2.98	28.96±3.37	✗	✗
ManifoldMixup[52]	73.69±1.78	49.65±1.94	52.24±2.08	43.75±2.04	31.09±3.13	32.81±3.18	✗	✓
PatchUp[14]	72.71±2.96	49.53±1.44	52.76±2.80	42.31±1.80	32.35±3.66	34.10±3.45	✗	✓
Ours-Convex	<u>78.44±1.60</u>	**67.81±1.04**	**71.12±1.08**	**70.60±1.30**	**33.98±1.04**	**37.42±1.03**	✗	✗
Ours-Mask	77.13±3.17	<u>66.16±1.58</u>	<u>68.90±1.62</u>	<u>68.40±2.16</u>	<u>32.95±2.26</u>	<u>36.38±2.23</u>	✗	✗

Convex combination
Binary Mask combination

For each column: **champion** <u>runner up</u>

Exp 2: Comparison with Existing Mixup Training

- Exp Setup: Run six times, mean and standard deviation, 40 epochs, α from $Beta(\beta = (1.0,1.0))$, budget 0.05. None of mixup schemes reported on-manifold robustness. We conduct a fair evaluation under the same setting.

- Finding 2: Against on-manifold attacks on CIFAR-10, LarepMixup always occupied champions and runners-up.

Table 2: Accuracy (%) of CIFAR-10 classification models on off/on-manifold adversarial examples

PreActResNet18

Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	OM-FGSM	OM-PGD	Known Attacker	Modify Network
Vanilla	**87.37±0.00**	32.07±0.00	28.93±0.00	7.59±0.00	10.36±0.00	2.60±0.00	51.02±0.00	21.68±0.00		
InputMixup[56]	84.48±1.45	63.58±3.36	68.12±3.46	56.63±10.20	37.97±2.58	41.11±2.10	58.53±0.43	44.11±1.34	✗	✗
CutMix[54]	82.14±3.00	65.51±1.03	69.67±1.34	64.41±3.55	36.79±2.60	39.74±3.10	57.59±0.31	43.50±1.71	✗	✗
PuzzleMixup[29]	83.11±1.64	65.73±2.46	70.35±2.60	64.03±6.06	38.86±1.53	41.83±1.74	57.80±0.77	43.68±2.19	✗	✗
ManifoldMixup[52]	71.10±4.17	49.26±1.34	52.49±1.91	44.08±1.60	25.33±2.76	27.19±2.53	50.16±1.66	38.64±0.80	✗	✓
PatchUp[14]	72.02±4.10	51.35±2.13	55.91±2.29	44.61±2.56	28.81±3.35	30.94±3.13	52.22±2.32	41.33±1.24	✗	✓
Ours-Convex	84.02±1.77	**68.86±2.88**	**72.65±3.59**	**66.98±5.93**	39.03±2.16	42.03±2.31	**60.02±0.91**	**46.72±1.52**	✗	✗
Ours-Mask	84.60±1.27	66.56±1.50	71.22±1.93	63.69±4.61	**39.27±2.97**	**42.54±2.74**	58.36±0.60	44.80±0.73	✗	✗

PreActResNet34

Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	OM-FGSM	OM-PGD	Known Attacker	Modify Network
Vanilla	**83.57±0.00**	31.37±0.00	25.71±0.00	5.27±0.00	12.27±0.00	1.89±0.00	49.23±0.00	17.05±0.00		
InputMixup[56]	68.42±7.38	62.19±4.22	63.84±4.98	63.79±4.99	26.36±4.07	29.77±4.16	54.68±3.84	47.18±2.29	✗	✗
CutMix[54]	71.21±6.16	62.45±2.71	64.61±3.50	64.30±3.16	28.88±2.07	32.12±2.38	55.65±2.56	46.40±0.99	✗	✗
PuzzleMixup[29]	67.06±7.62	60.89±4.99	62.55±5.76	62.66±5.84	25.89±2.98	28.96±3.37	54.04±3.87	46.31±2.05	✗	✗
ManifoldMixup[52]	73.69±1.78	49.65±1.94	52.24±2.08	43.75±2.04	31.09±3.13	32.81±3.18	52.99±0.24	39.47±1.34	✗	✓
PatchUp[14]	72.71±2.96	49.53±1.44	52.76±2.80	42.31±1.80	32.35±3.66	34.10±3.45	53.03±2.37	39.38±1.63	✗	✓
Ours-Convex	78.44±1.60	**67.81±1.04**	**71.12±1.08**	**70.60±1.30**	**33.98±1.04**	**37.42±1.03**	**58.96±0.67**	**47.99±1.16**	✗	✗
Ours-Mask	77.13±3.17	66.16±1.58	68.90±1.62	68.40±2.16	32.95±2.26	36.38±2.23	58.31±0.96	47.30±1.06	✗	✗

For each column: **champion** runner up

Exp 2: Comparison with Existing Mixup Training

❖ Exp Setup: Run six times, mean and standard deviation, 40 epochs, α from $Beta(\beta = (1.0, 1.0))$, budget 0.05. None of mixup schemes reported on-manifold robustness. We conduct a fair evaluation under the same setting.

● Finding 3: On SVHN, LarepMixup most frequently occupied champions and runners-up.

Table 3: Accuracy (%) of SVHN classification models on off/on-manifold adversarial examples

PreActResNet18

Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	OM-FGSM	OM-PGD	Known Attacker	Modify Network
Vanilla	**95.97±0.00**	57.29±0.00	34.57±0.00	29.21±0.00	22.51±0.00	21.54±0.00	41.04±0.00	6.78±0.00		
InputMixup[56]	94.39±0.79	68.77±2.03	58.81±2.34	51.25±2.22	**60.50±3.33**	<u>64.42±2.16</u>	44.58±0.86	18.48±1.04	✗	✗
CutMix[54]	94.19±1.07	68.78±2.01	59.52±3.28	52.50±3.64	57.45±3.26	63.62±1.52	44.31±1.02	17.87±0.91	✗	✗
PuzzleMixup[29]	<u>94.54±0.66</u>	67.55±1.79	58.79±3.34	51.65±3.48	55.87±2.22	63.42±1.51	43.63±0.62	16.00±1.15	✗	✗
ManifoldMixup[52]	89.15±4.22	67.21±1.85	<u>60.32±1.94</u>	<u>53.60±3.21</u>	52.95±3.15	60.57±1.97	43.32±1.52	**22.19±2.01**	✗	✓
PatchUp[14]	89.87±1.78	66.44±0.78	58.96±1.90	52.36±2.82	54.68±2.69	61.54±1.68	43.40±0.91	<u>21.51±1.05</u>	✗	✓
Ours-Convex	94.38±0.61	**70.62±1.35**	**63.35±0.67**	**56.66±1.22**	58.14±0.75	**64.45±0.54**	<u>45.24±0.44</u>	19.59±0.57	✗	✗
Ours-Mask	94.42±0.93	<u>70.22±1.30</u>	60.02±1.72	53.34±2.02	57.98±2.44	64.36±1.08	**45.26±0.54**	19.90±0.71	✗	✗

PreActResNet34

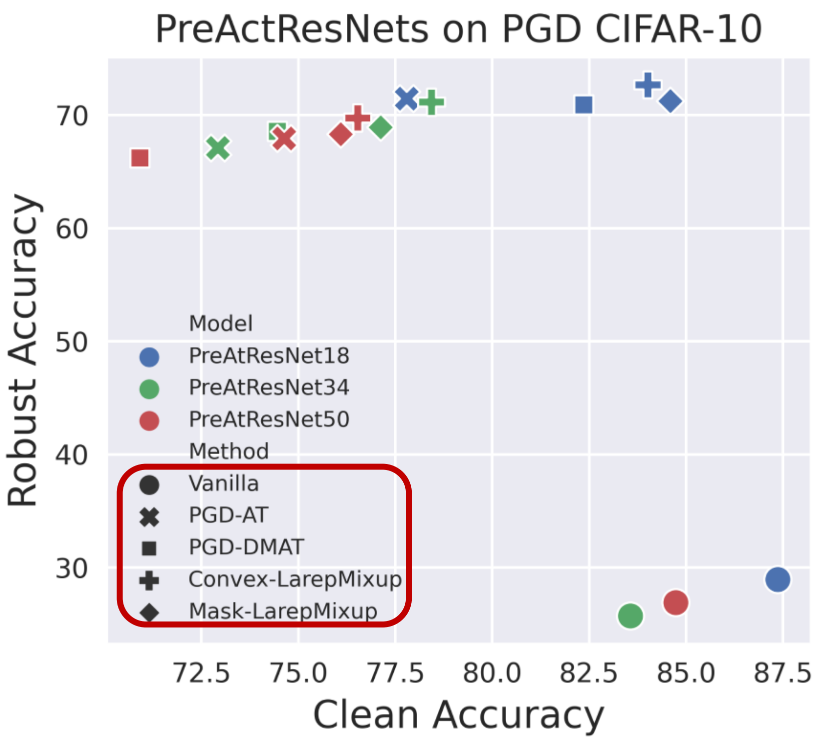
Method	Clean	FGSM	PGD	AutoAttack	DeepFool	CW	OM-FGSM	OM-PGD	Known Attacker	Modify Network
Vanilla	**95.75±0.00**	57.11±0.00	35.57±0.00	29.80±0.00	19.94±0.00	25.62±0.00	36.62±0.00	5.01±0.00		
InputMixup[56]	93.41±1.85	66.14±0.85	60.42±6.52	52.82±7.44	49.76±3.32	62.47±1.10	39.97±0.97	17.07±0.85	✗	✗
CutMix[54]	93.36±2.74	65.71±0.56	60.09±7.25	53.39±8.66	49.26±2.00	61.83±1.35	39.81±1.09	16.25±0.88	✗	✗
PuzzleMixup[29]	92.53±4.79	65.12±0.82	61.06±7.05	54.17±8.54	48.65±3.22	61.63±2.37	39.24±1.89	15.89±2.15	✗	✗
ManifoldMixup[52]	81.27±2.68	61.63±2.07	**63.61±3.10**	**59.19±1.94**	44.88±4.40	56.29±3.92	36.11±1.07	<u>21.68±1.26</u>	✗	✓
PatchUp[14]	68.39±9.86	51.94±4.91	55.01±6.31	52.17±5.91	36.07±2.41	47.47±5.47	31.81±2.20	**22.19±2.72**	✗	✓
Ours-Convex	<u>94.94±0.31</u>	**68.37±0.76**	61.75±3.65	53.55±4.05	**52.21±1.67**	**64.61±1.27**	**41.13±0.41**	16.88±0.38	✗	✗
Ours-Mask	93.63±1.13	<u>67.69±0.52</u>	<u>63.21±5.39</u>	<u>55.74±5.69</u>	<u>52.10±2.75</u>	<u>64.27±1.30</u>	<u>40.70±0.60</u>	17.01±0.47	✗	✗

For each column: **champion** <u>runner up</u>

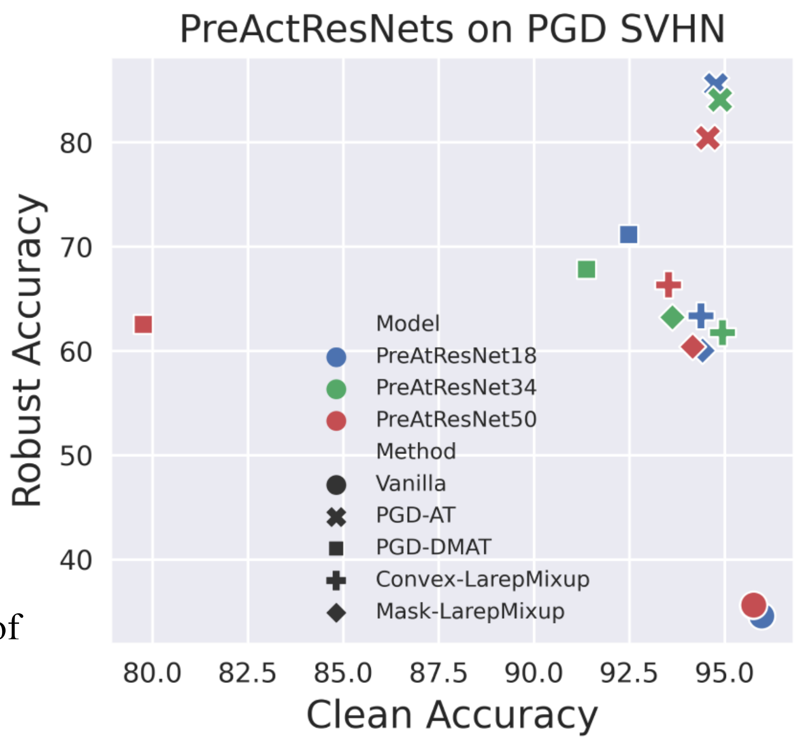
Exp 3: Comparison with Existing Adversarial Training

- ❖ Exp Setup: budget $\epsilon = 0.05$, single step budget is 0.02. budget $\eta = 0.05$, single step budget is 0.005. The number of augmented adversarial examples is the same as the number of augmented mixed examples.
- ● There is a strong assumption in AT, that is, the defender needs to construct adversarial examples during the training phase.

CIFAR-10



SVHN



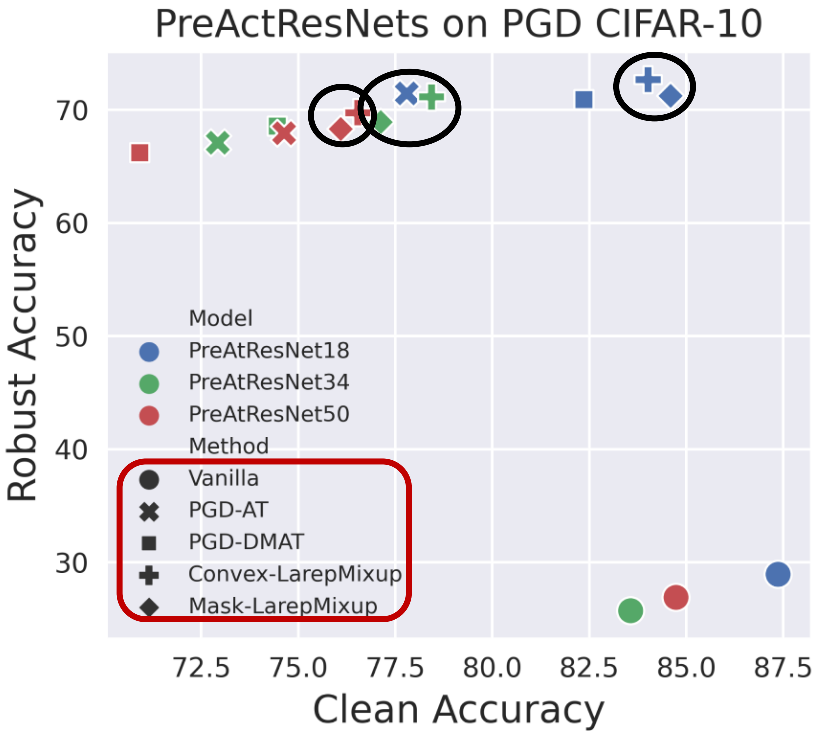
- ✓ larger x-axis means higher clean accuracy
- ✓ larger y-axis means higher adversarial accuracy
- ✓ The same color is a group of comparison results.

Exp 3: Comparison with Existing Adversarial Training

❖ Exp Setup: budget $\epsilon = 0.05$, single step budget is 0.02. budget $\eta = 0.05$, single step budget is 0.005. The number of augmented adversarial examples is the same as the number of augmented mixed examples.

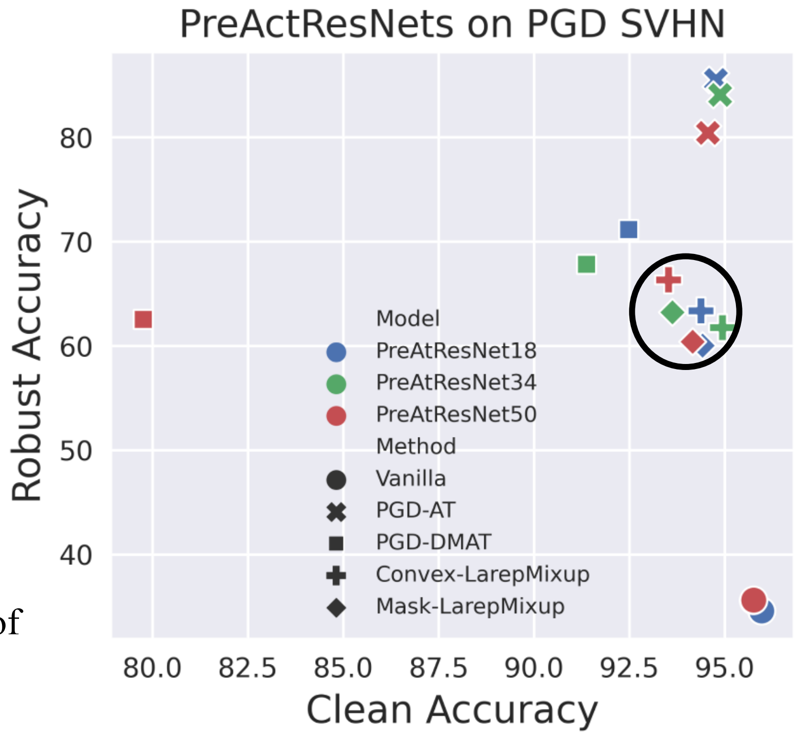
- Finding 1: Against PGD on CIFAR-10, LarepMixup are better than AT and DMAT in both aspects.
- Finding 2: Against PGD on SVHN, LarepMixup outperforms in clean accuracy.

CIFAR-10



✓ larger x-axis means higher clean accuracy
✓ larger y-axis means higher adversarial accuracy
✓ The same color is a group of comparison results.

SVHN

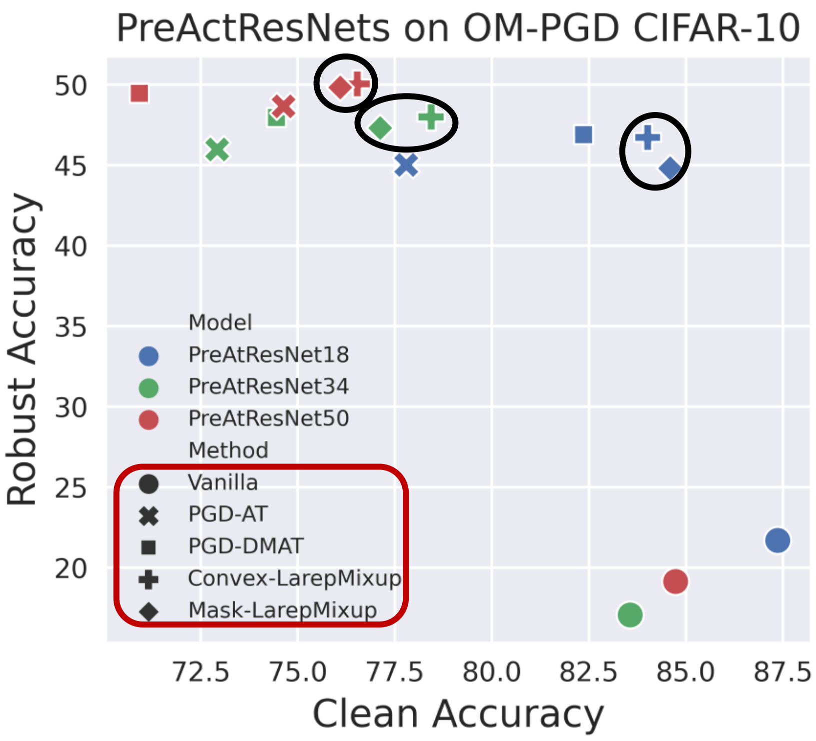


Exp 3: Comparison with Existing Adversarial Training

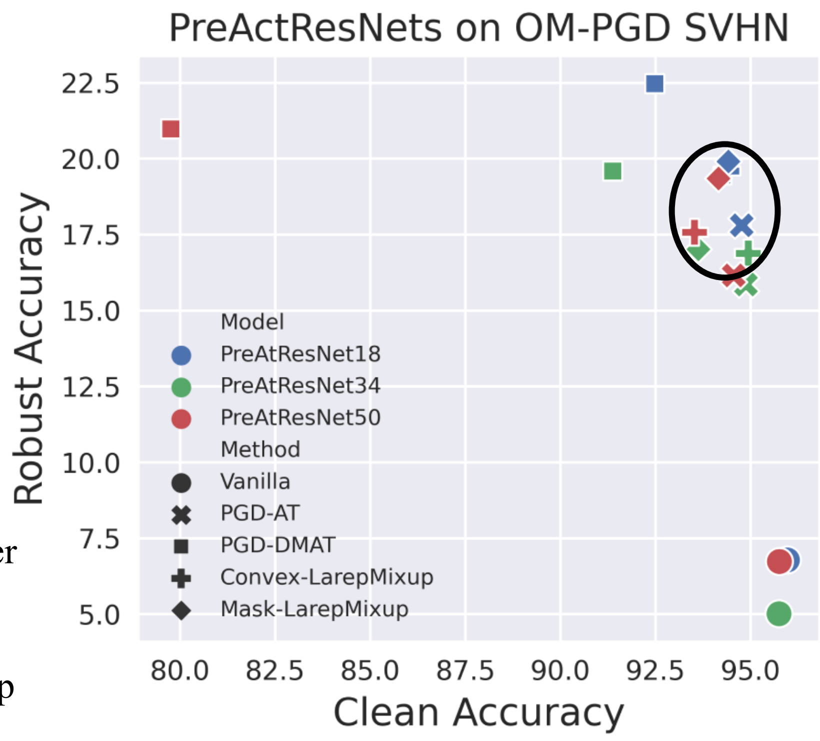
- ❖ Exp Setup: budget $\epsilon = 0.05$, single step budget is 0.02. budget $\eta = 0.05$, single step budget is 0.005. The number of augmented adversarial examples is the same as the number of augmented mixed examples.

- Finding 3: Against OM-PGD on CIFAR-10 and SVHN, conclusions from observation 1/2 hold true.
- Finding 4: Robustness advantage between PGD-AT and PGD-DMAT is reversed.

CIFAR-10



SVHN

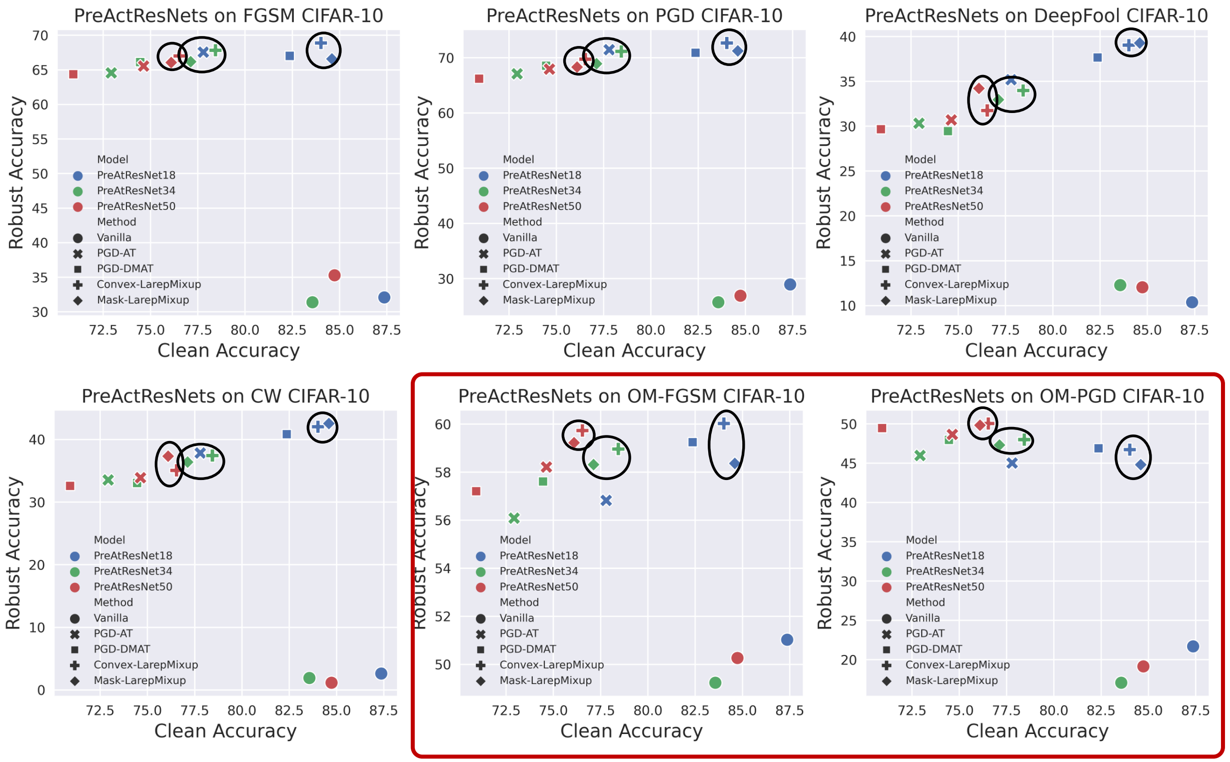


- ✓ Right points mean better accuracy on clean examples.
- ✓ Higher points mean better accuracy on adversarial examples.
- ✓ The same color is a group of comparison results.

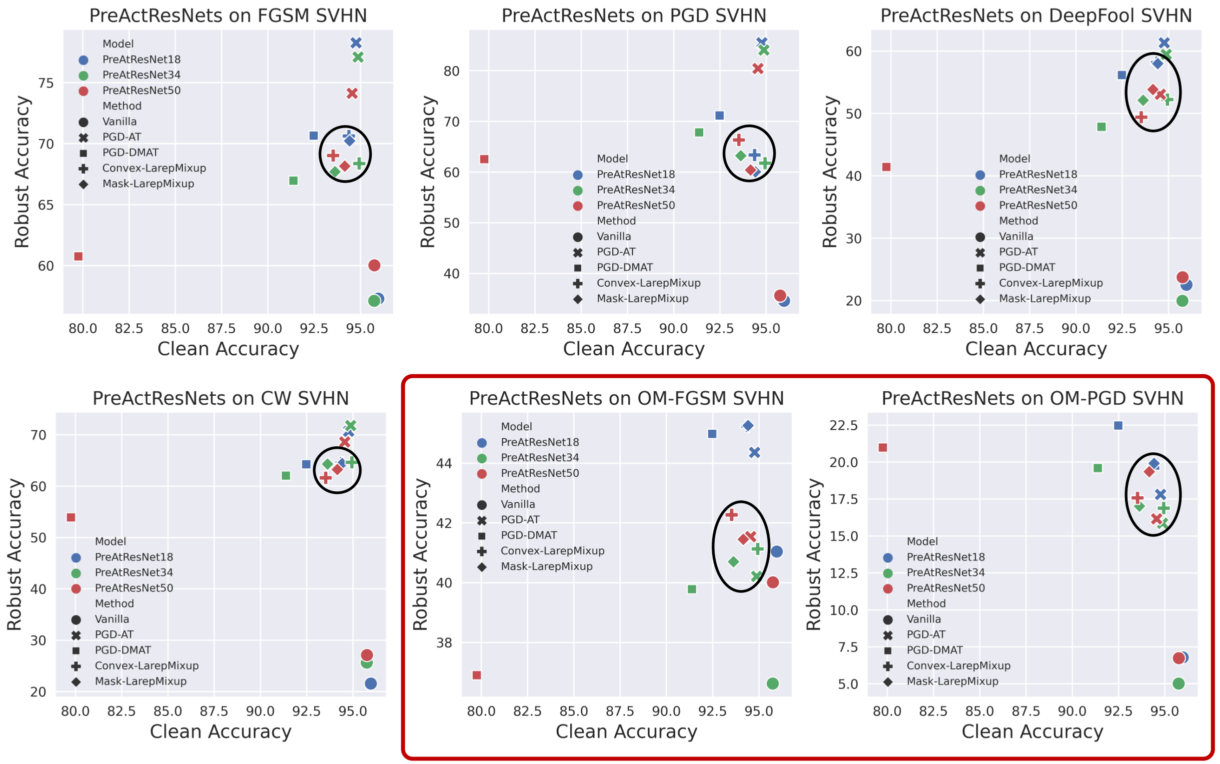
Exp 3: Comparison with Existing Adversarial Training

- ❖ Exp Setup: budget $\epsilon = 0.05$, single step budget is 0.02. budget $\eta = 0.05$, single step budget is 0.005. The number of augmented adversarial examples is the same as the number of augmented mixed examples.
- Finding 5: On other attacks (FGSM, OM-FGSM, DeepFool, CW), previous conclusions hold true.
- Finding 6: PGD-AT and PGD-DMAT have decreased robustness improvement against non-PGD related attacks.

CIFAR-10



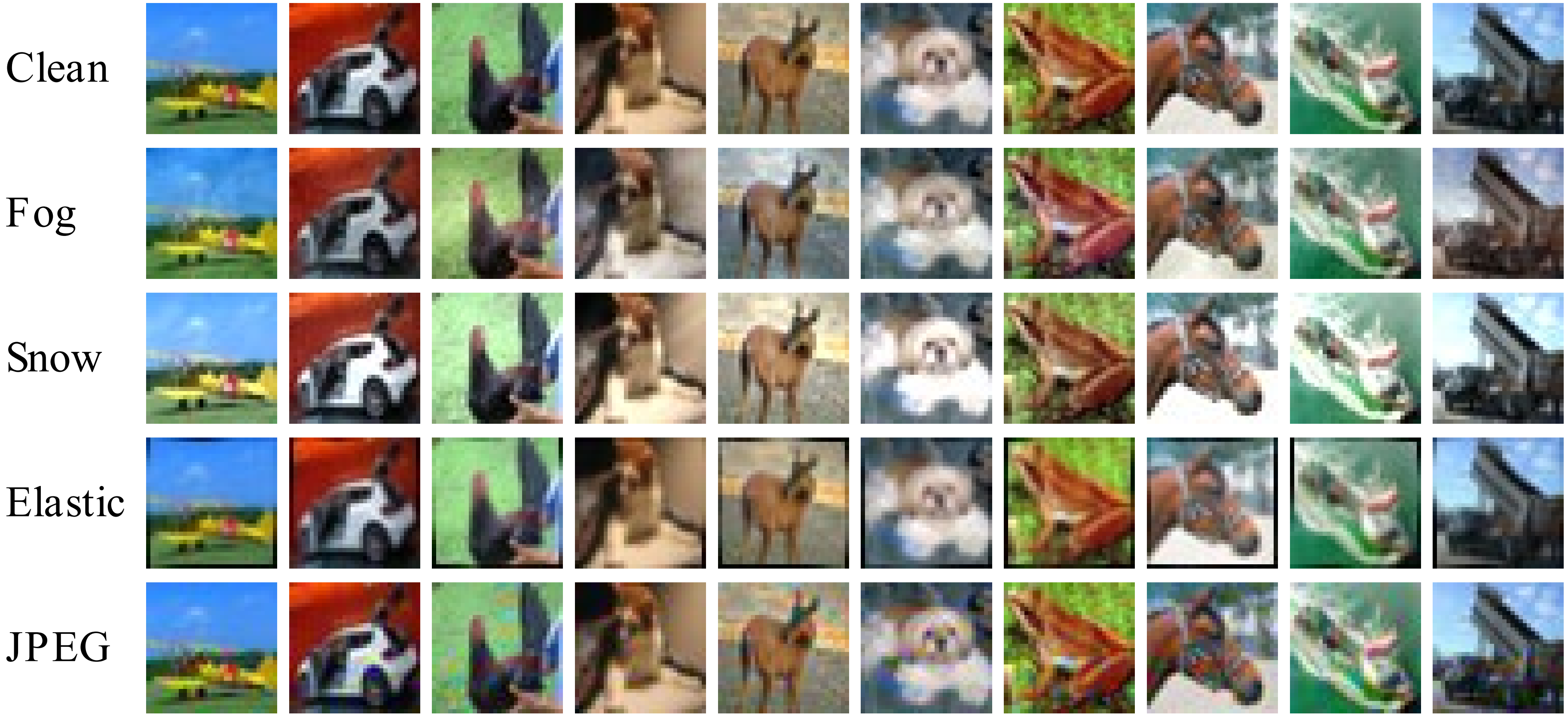
SVHN



Exp 4: Robustness against Non-L_p Constrained Perturbations

- Exp Setup: 4 perceptual attacks Fog, Snow, Elastic, JPEG. Run three times and take the average.
- Simulate natural environmental noise, compression distortion, etc.

CIFAR-10

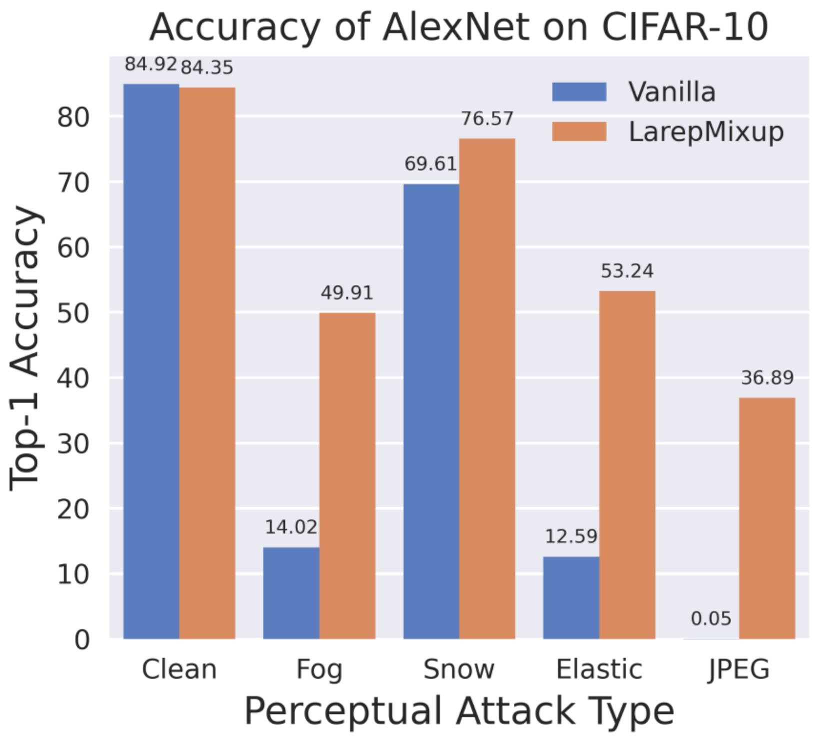


Exp 4: Robustness against Non-L_p Constrained Perturbations

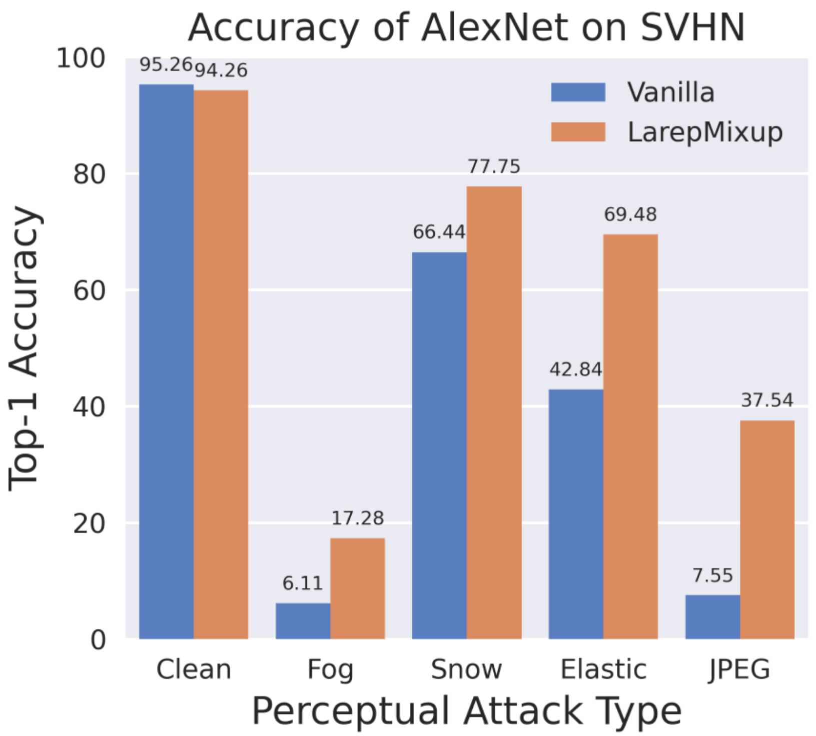
❖ Exp Setup: 4 perceptual attacks Fog, Snow, Elastic, JPEG. Run three times and take the average.

- Finding 1: The robust accuracy of AlexNet against perceptual attacks shows significant increase.
- Finding 2: The clean accuracy of AlexNet is not much different before and after LarepMixup training.

CIFAR-10



SVHN

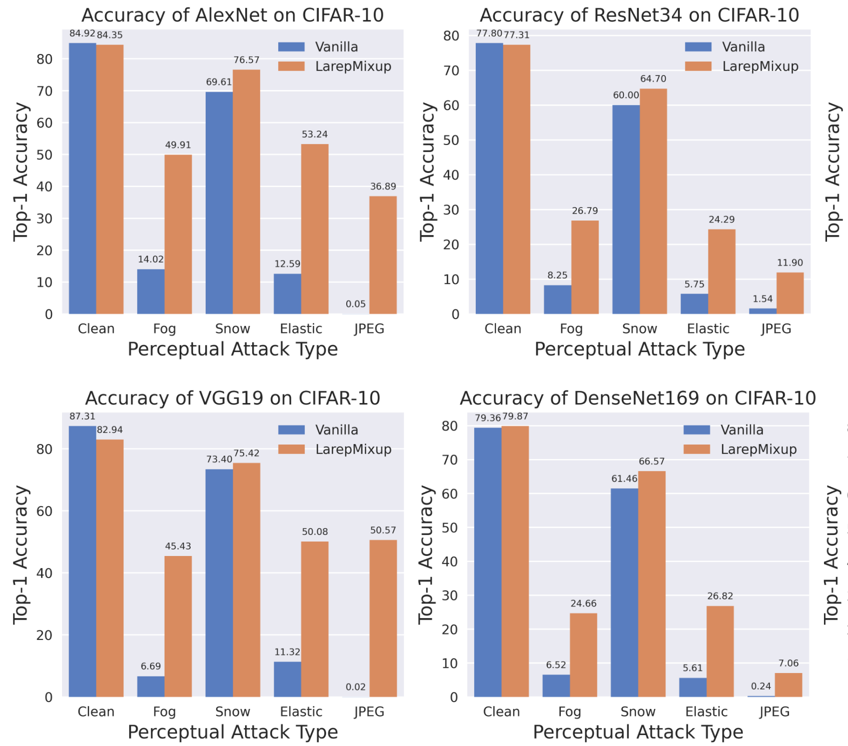


Exp 4: Robustness against Non-L_p Constrained Perturbations

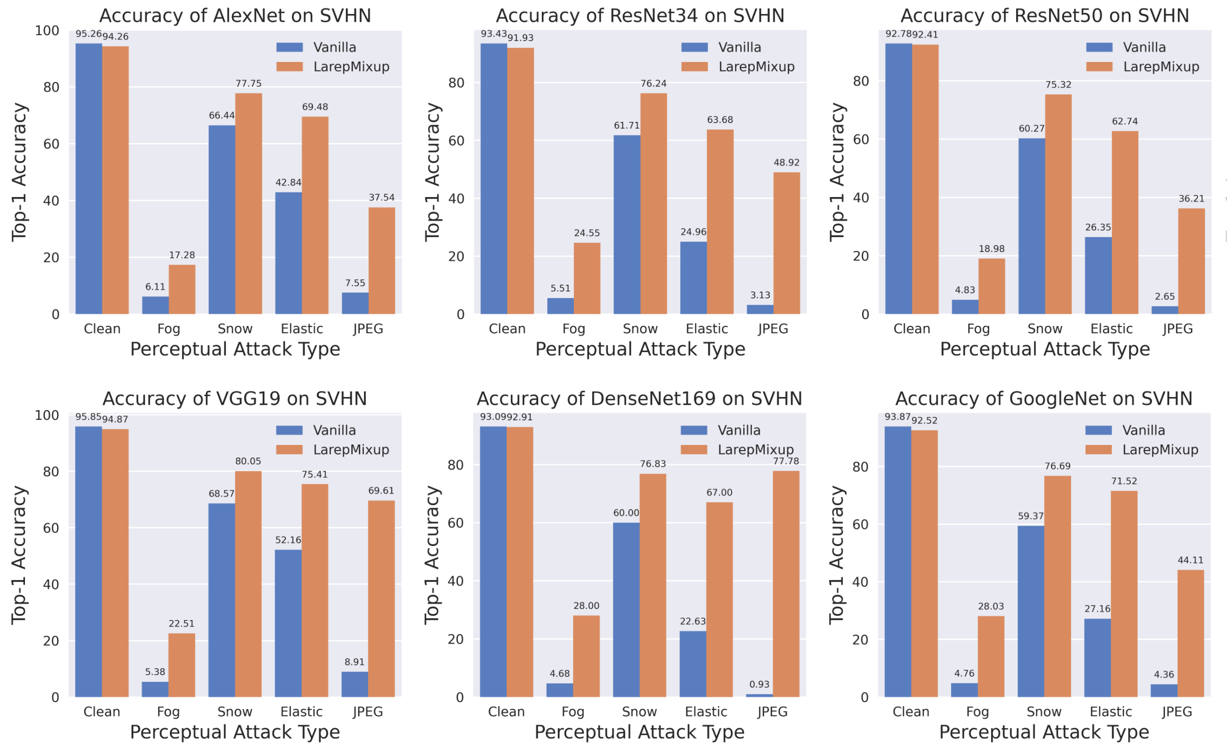
❖ Exp Setup: 4 perceptual attacks Fog, Snow, Elastic, JPEG. Run three times and take the average.

● Finding 3: On other models (VGG19, ResNet34, DenseNet169, ResNet50, GoogleNet), conclusions from observation 1/2 hold true.

CIFAR-10



SVHN



Exp 5: Effect of Mixing Modes

❖ Exp Setup: High dimensional ImageNet-Mixed 10 (256×256 pixels). Run three times and take average.

- Finding 1: For off-manifold attacks, the robustness improvement from four mixing modes is not much different.
- Finding 2: For on-manifold attacks, the advantage of convex mixing is obsivious. For source samples on the same object manifold, linear combination is more likely to produce interpolation points lying on the manifold.
- Finding 3: There is little difference in accuracy improvement in terms of the number of mixed source samples.

Table 4: Robust accuracy (%) of PreActResNet18 under different mixing modes (ImageNet-Mixed10)

	Method	Vanilla	Dual-LarepMixup Convex	Dual-LarepMixup Mask	Ternary-LarepMixup Convex	Ternary-LarepMixup Mask
	Clean	90.47	90.57±0.55	**90.89±0.35**	90.67±0.21	90.24±1.25
off-manifold	FGSM	13.93	17.09±0.29	16.21±0.14	16.71±0.34	**17.29±0.94**
	PGD	2.00	5.38±0.81	4.68±0.45	4.73±0.69	**5.81±1.32**
	AutoAttack	0.00	**3.74±0.19**	3.68±0.29	3.60±0.18	3.66±0.04
	DeepFool	8.87	**85.38±0.19**	83.98±0.42	84.89±0.18	83.93±1.00
	CW	0.10	**84.61±0.30**	83.16±0.52	84.19±0.47	83.28±0.62
on-manifold	OM-FGSM	26.90	**59.91±1.30**	28.61±5.58	57.36±1.89	28.21±0.98
	OM-PGD	20.43	**58.76±1.30**	27.99±5.92	56.59±1.87	27.47±1.44

Summary

- We propose LarepMixup, a mixup-based training framework towards addressing the threats from off/on-manifold adversarial attacks at the same time.
- We design a flexible data augmentation strategy, dual-mode manifold interpolation, for generating mixed examples using convex or binary mask mixing modes.
- To our knowledge, we are the first to focus on the performance of the mixup trainied model on on-manifold L_p attacks and off-manifold non-L_p attacks.

Future Work

- While mixup training was originally proposed for image classification tasks, it can be extended to other input domains, such as natural language processing, network intrusion detection.
  - Text Classification
  - Network Traffic Classification
- Help improve DNN's capability to handle variations in language syntax or traffic patterns and increases the model's robustness to unseen adversarial evasion attacks.

THANK YOU!

Mengdie Huang[1], Yi Xie[1], Xiaofeng Chen[1], Jin Li[2], Changyu Dong[3], Zheli Liu[4], Willy Susilo[5]
[1] Xidian University
[2] Guangzhou University
[3] Newcastle University
[4] Nankai University
[5] University of Wollongong

Q&A

Mengdie Huang (Maggie)
mdhuang1@stu.xidian.edu.cn