한국과학기술연구원
Korea Institute of Science and Technology

Feature Separation and Recalibration for Adversarial Robustness

CVPR 2023 Highlight

[paper] [code]

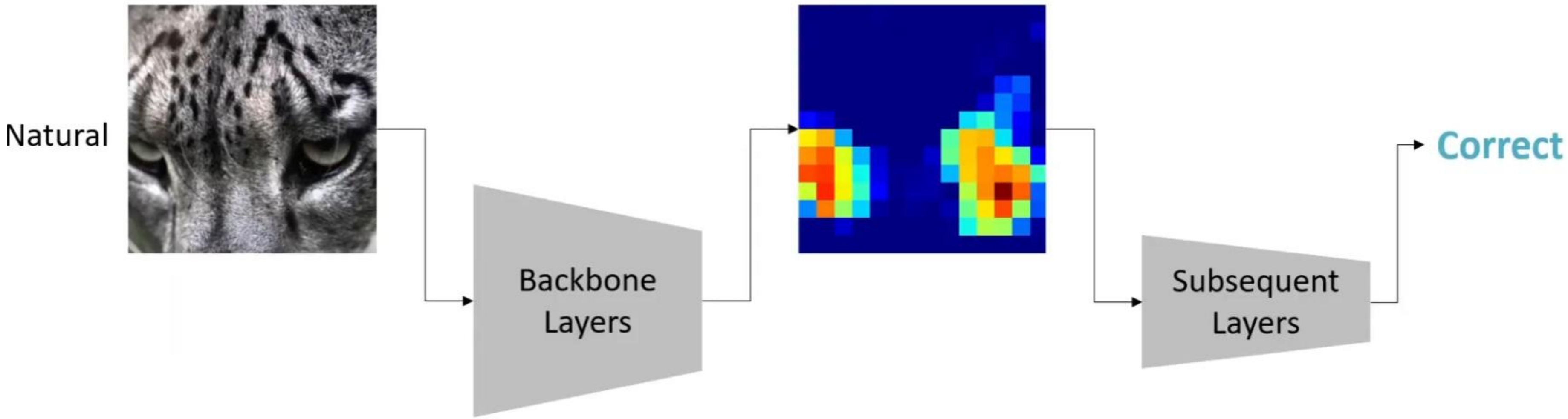
Woo Jae Kim, Yoonki Cho, Junsik Jung, Sung-Eui Yoon (KAIST)

2023.09.15

Mijin Koo

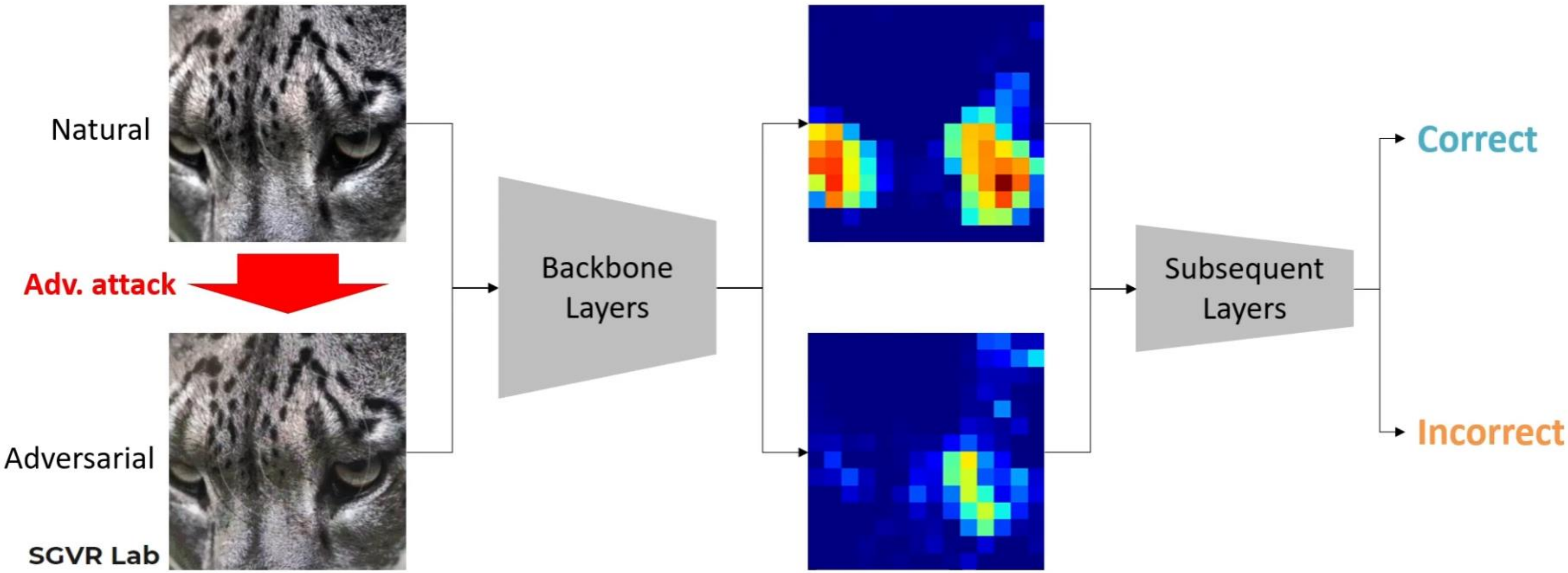
Summary

- **Feature Activation Disruption upon Adversarial Attack**
  - Feature-level disruptions lead to model mispredictions



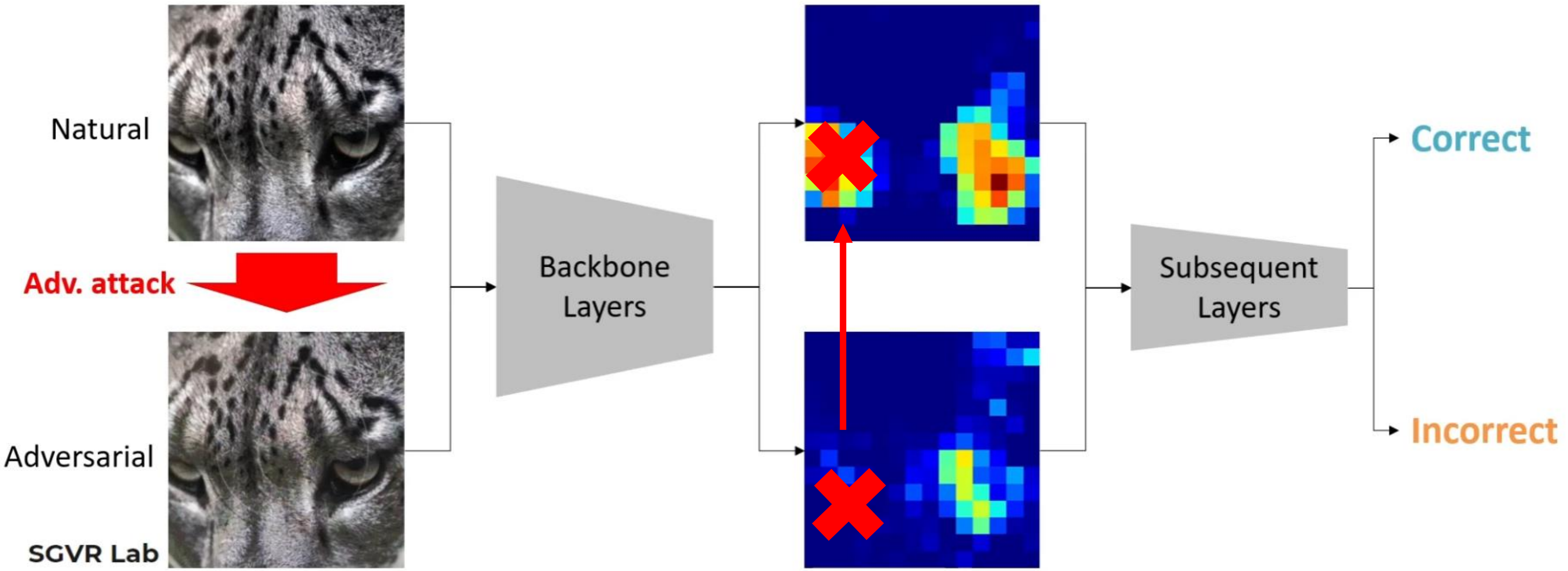
Summary

- **Feature Activation Disruption upon Adversarial Attack**
  - Feature-level disruptions lead to model mispredictions



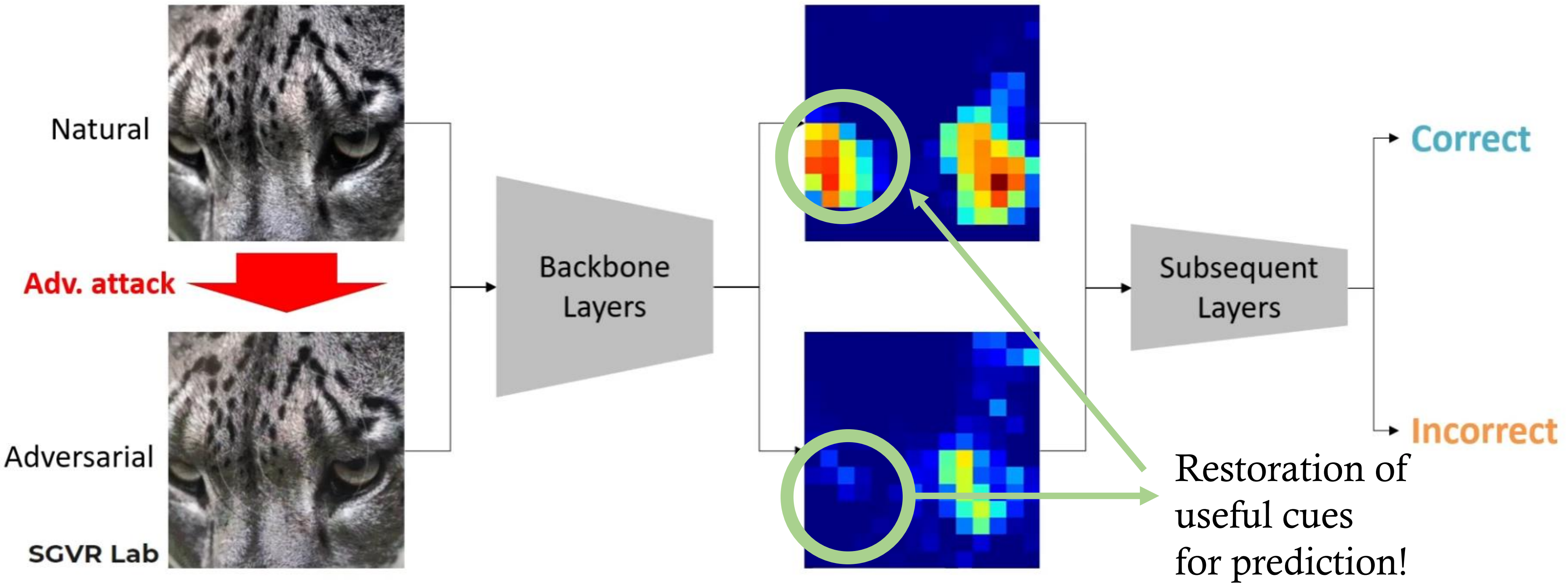
Summary

- **Limitations of Conventional Defense**
  - Conventional defense methods ***suppressed or deactivated*** disrupted activations
  - This approach lead to ***loss of potentially discriminative cues***



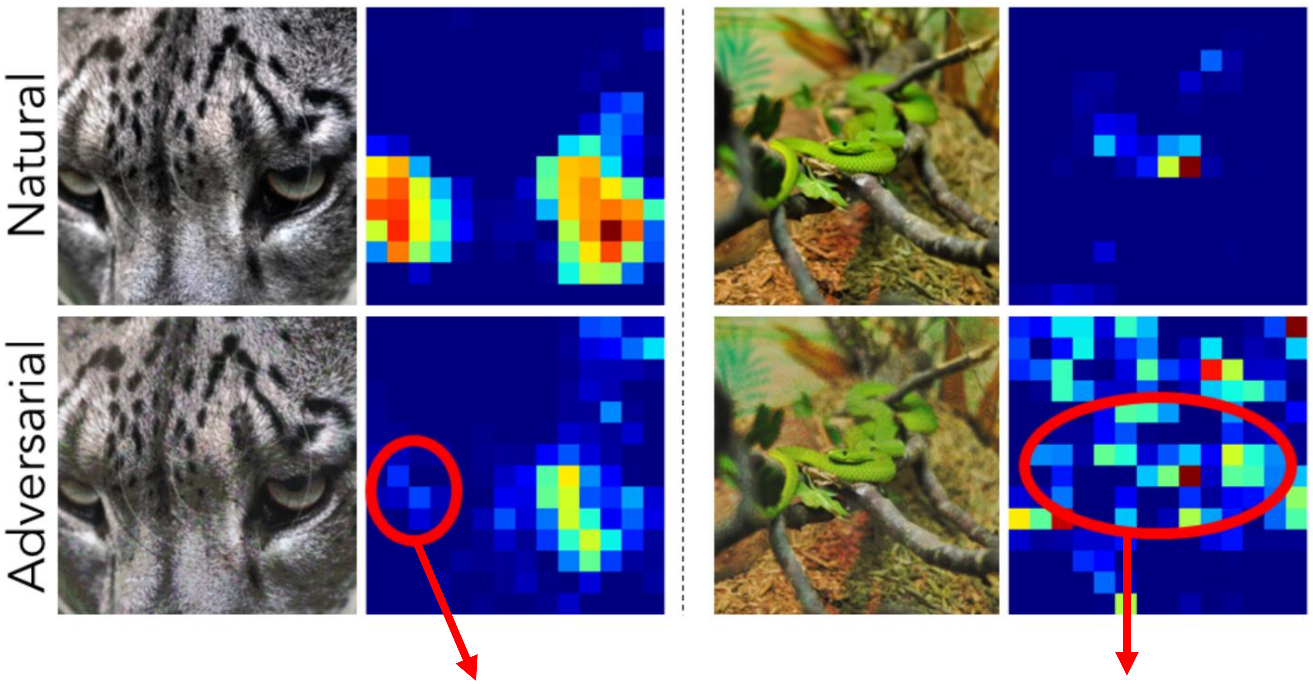
Summary

- **Proposed Approach**
  - Instead, we propose to ***restore useful cues*** from these disrupted activations
  - This additional useful cues ***enrich*** model's ability to make ***correct predictions***



Introduction

- **Feature Activation Disruption upon Adversarial Attack**
  - Adversarial attacks corrupt activations of feature maps
  - **Robust feature** with activations that help the model make correct predictions
  - **Non-robust feature** with activations that are responsible for model mispredictions upon adversarial attack.

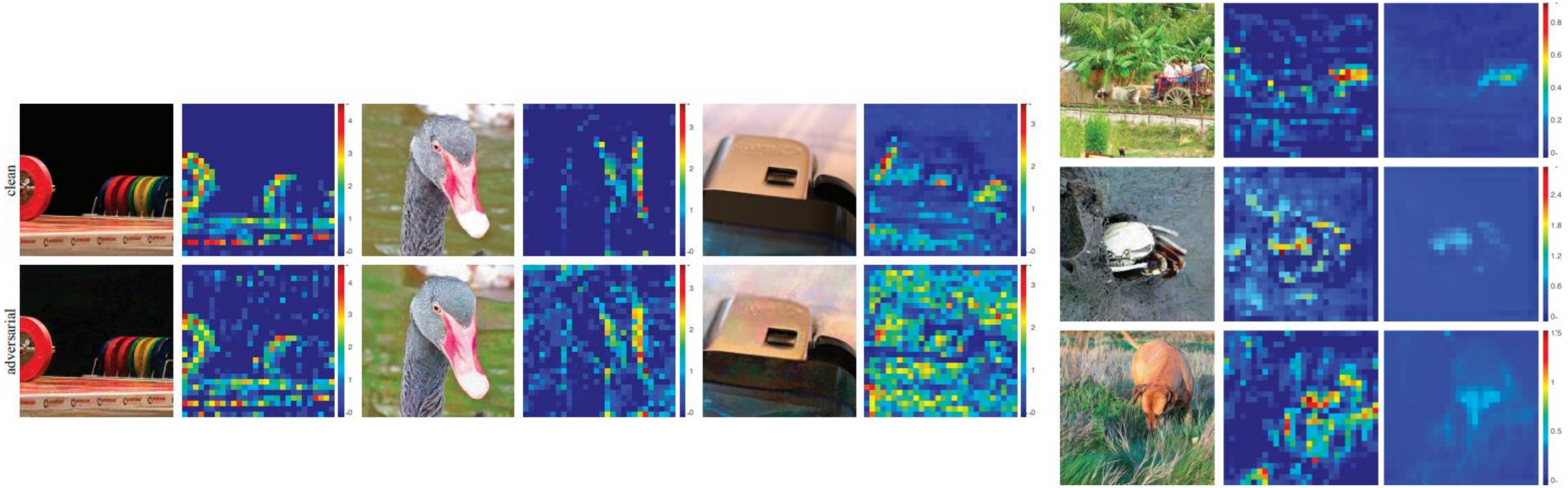


Loss of useful cues
→ Restoration

Gain of useless information
→ Deactivating

Introduction

- **Deactivating the non-robust feature activations (Previous)**
  - Adversarial perturbations on images lead to noise in the features.
  - Previous methods solve this problem by deactivating the non-robust feature activations that cause model mispredictions.
  - Increase adversarial robustness by performing ***feature denoising***
  - Our networks contain blocks that denoise the features using ***non-local means***

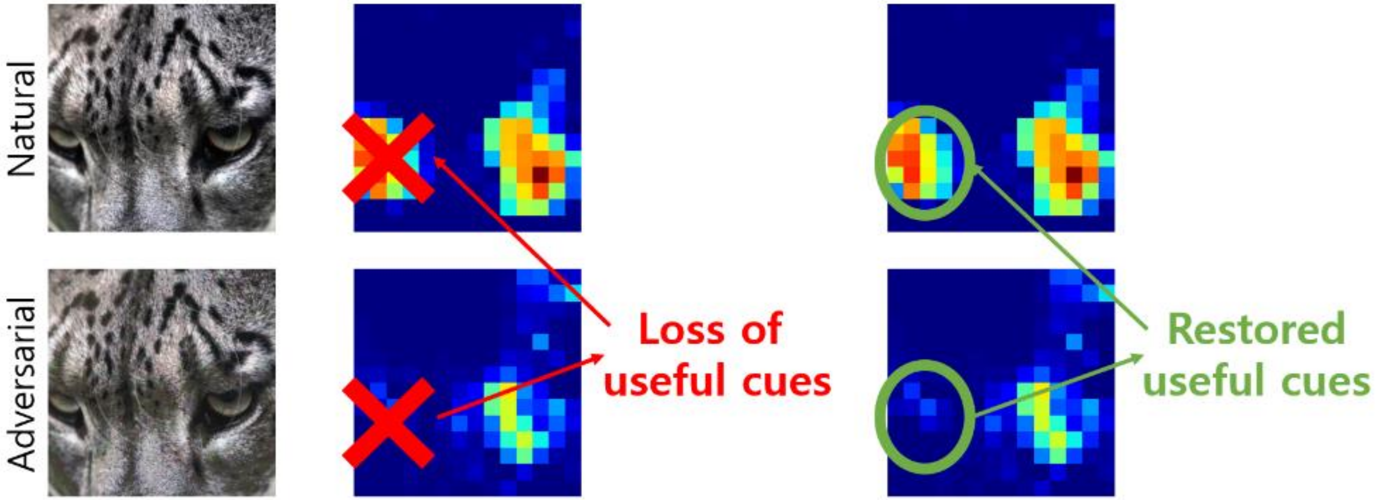


Feature denoising for improving adversarial robustness [2019 CVPR]

Introduction

- **Motivation: non robust feature에도 discriminative cue가 있다!**
  - We propose to ***restore useful cues from these disrupted activations*** that are otherwise neglected.

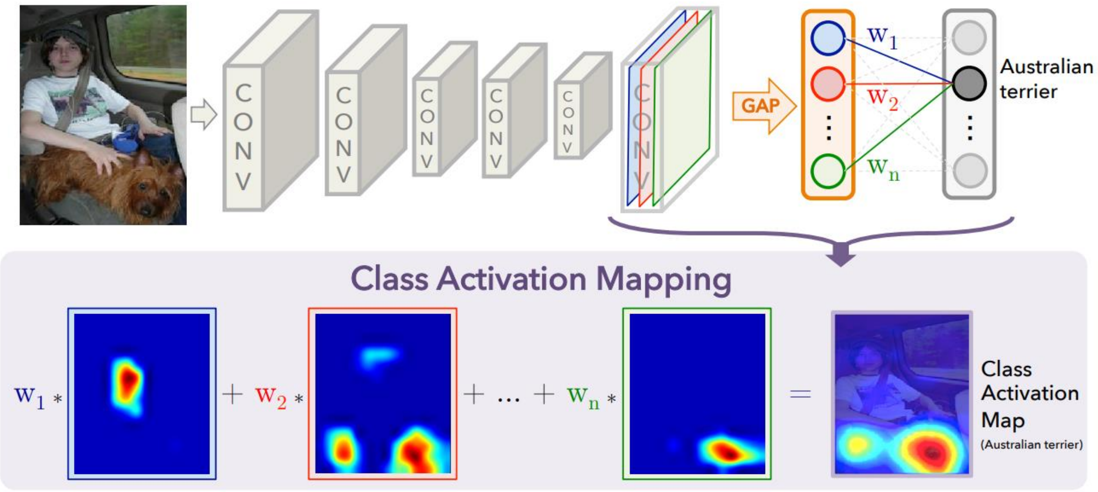
- **Contributions**
  - Novel approach of recalibrating deactivated activations to capture useful cues for correct model predictions
  - Easy to plug in Feature and Recalibration(FSR) module
  - Small overhead, successful experiments results



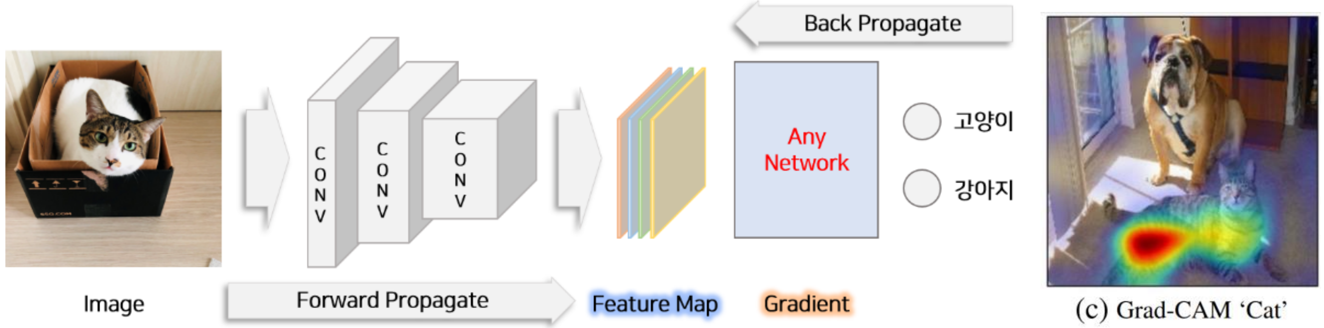
Conventional Approach **Our Approach**

Introduction

- **Visualize Explanation from Deep Networks**
  - **CAM** (Class Activation Mapping)
    - the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps
    - CAM highlights the class-specific discriminative regions
  - **Grad-CAM** (Gradient weighted CAM)
    - uses the gradients of any target flowing into the final convolutional layer



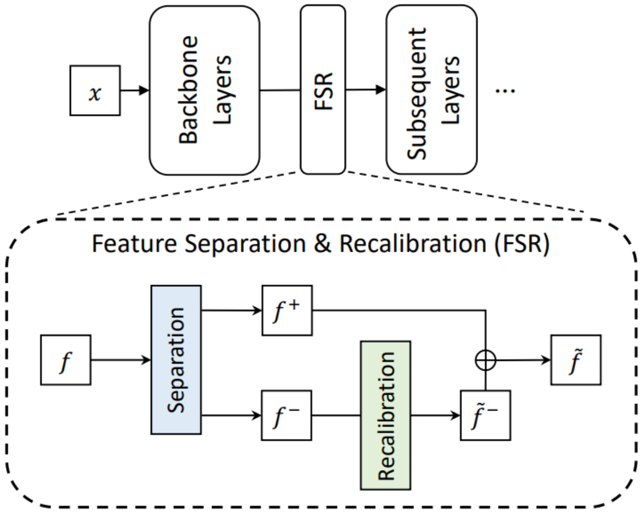
Learning Deep Features for Discriminative Localization [2016 CVPR]



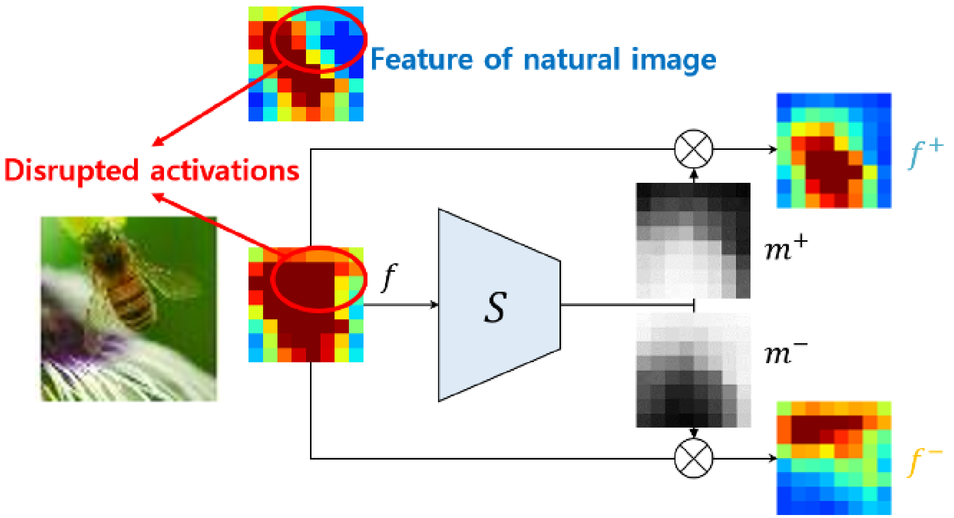
Grad CAM: Visual Explanations from Deep Networks via Gradient-based Localization [2017 ICCV]

Proposed Approach

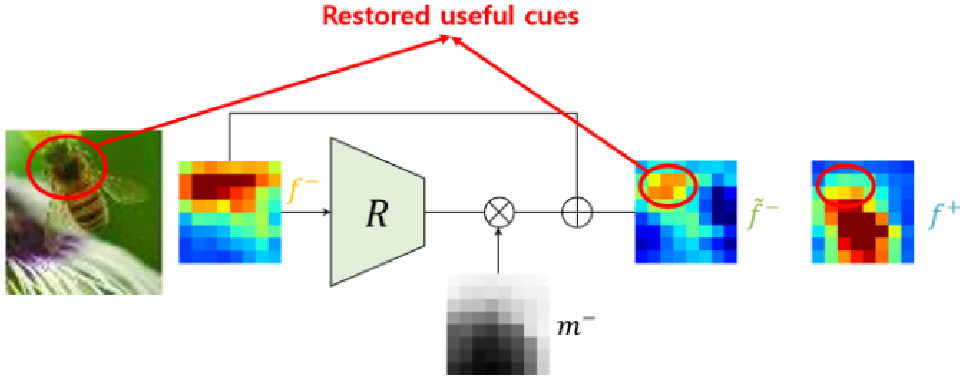
- **Feature Separation and Recalibration (FSR)**



Separation



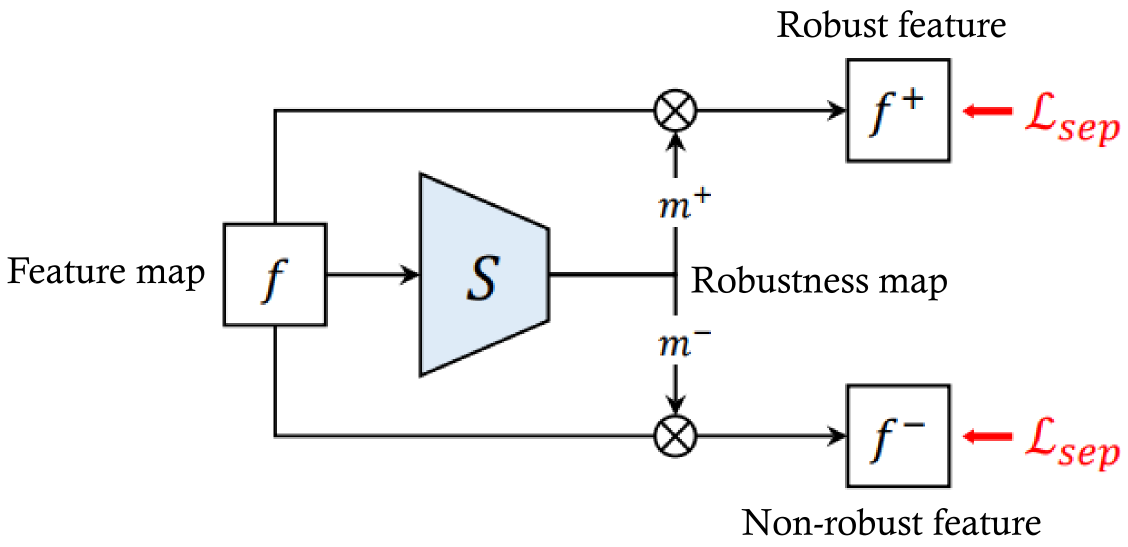
Recalibration



Proposed Approach

- **Separation stage**
  - Disentangle the feature map into the robust and non-robust features by masking out

- **Feature Separation**
  - Separation Net S
  - Input: feature map f
  - Output: robustness map r
  - Differentiable soft mask $m \in [0, 1]$; approximated by a binary mask $b \in \{0, 1\}$
    - $m^{+} = 1 - m^{-}$



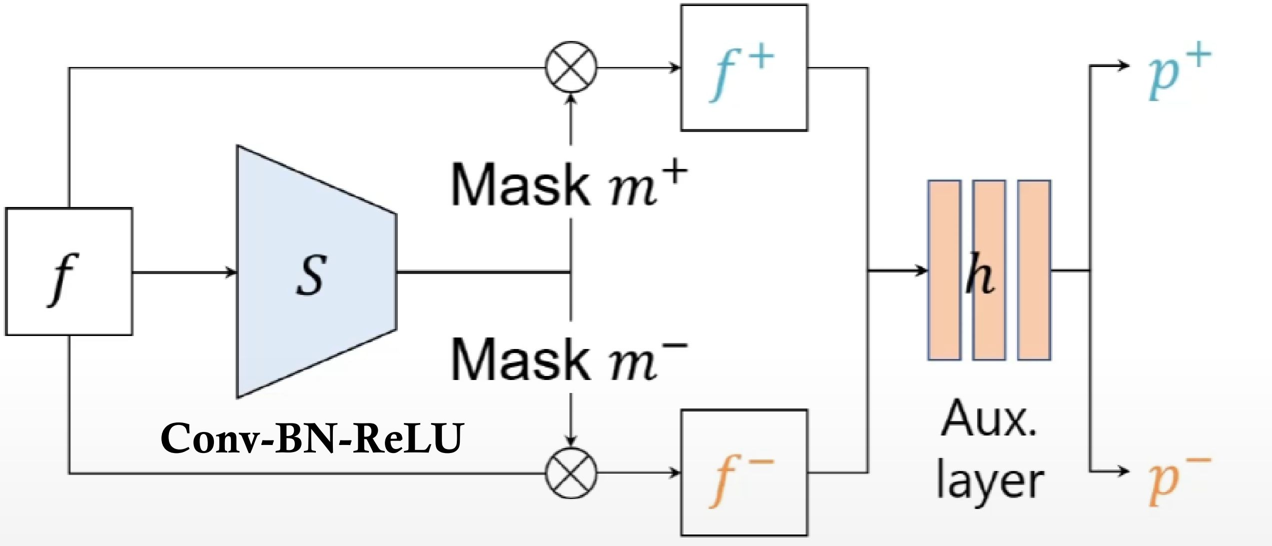
- Robust feature $f^{+} = m^{+} \otimes f$
- Non-robust feature $f^{-} = m^{-} \otimes f$

Proposed Approach

- **Feature Separation**
  - Separation Net S learns robustness score
  - L_{sep} guides the Separation Net to assign high robustness scores to units that help the auxiliary layer make correct predictions

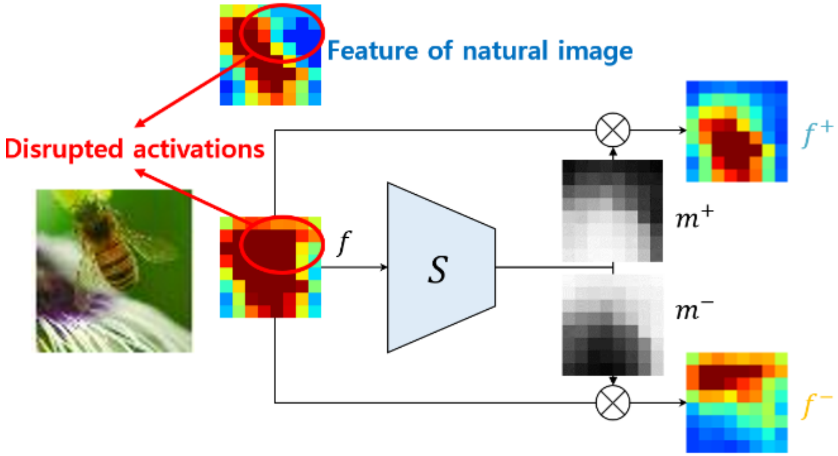
$$\mathcal{L}_{sep} = -\sum_{i=1}^{N}(y_i \cdot \log(p_i^+) + y_i' \cdot \log(p_i^-)), \qquad (3)$$

$$\mathcal{L}_{sep} = \underbrace{\mathcal{H}}_{\text{Cross-entropy loss}}(p^+, \underbrace{y}_{\text{GT label}}) + \mathcal{H}(\underbrace{p^-}_{\text{Pred. logit}}, \underbrace{y'}_{\text{Wrong label}})$$



Proposed Approach

- **Feature Separation**
  - **Soft mask M**
    - Positive mask emphasizes activations relevant to ***correct predictions***
    - Negative mask emphasizes activations relevant to ***mispredictions***



- $b \in \{0, 1\}$ with a differentiable soft mask $m \in [0, 1]$
  - By Gumbel softmax Approximate a binary mask
  - r: robustness map
  - g_1, g_2: samples from Gumbel distribution such that $g = -\log(-\log(u))\,, u \sim \text{Uniform}(0, 1)$

$$m = \frac{e^{((\log(\sigma(r))+g_1)/\tau)}}{e^{((\log(\sigma(r))+g_1)/\tau)} + e^{((\log(1-\sigma(r))+g_2)/\tau)}}, \qquad (2)$$

Proposed Approach

- **Feature Separation**
  - **Gumbel Softmax**
    - 이산 확률 분포에서의 샘플링을 연속적으로 다루기 위한 방법 → gradient 계산 가능

$$y_i = \frac{\exp(\boxed{(\log(\pi_i) + g_i)/\tau})}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)}$$

Gumbel softmax

$$p_i = \frac{e^{\boxed{z_i}}}{\sum_{j=1}^{k} e^{z_j}}$$

Softmax

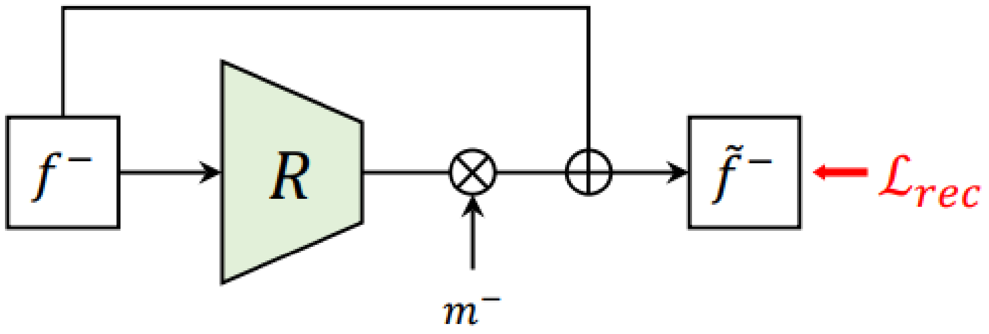
- r: robustness map
- σ: sigmoid function
- g: samples from Gumbel distribution such that g=−log(−log(u)), u ~ Uniform(0, 1)
- **τ**: temperature that controls the effect of g

$$m = \frac{e^{((\log(\sigma(r))+g_1)/\tau)}}{e^{((\log(\sigma(r))+g_1)/\tau)} + e^{((\log(1-\sigma(r))+g_2)/\tau)}}, \qquad (2)$$

Proposed Approach

- **Recalibration Stage**
  - Adjust the non-robust feature activations to capture the additional useful cues

- **Feature Recalibration**
  - Recalibration Net R
  - Input: non-robust feature map f^-
  - Output: recalibrated feature, $\tilde{f}^- = f^- + m^- \otimes R(f^-)$

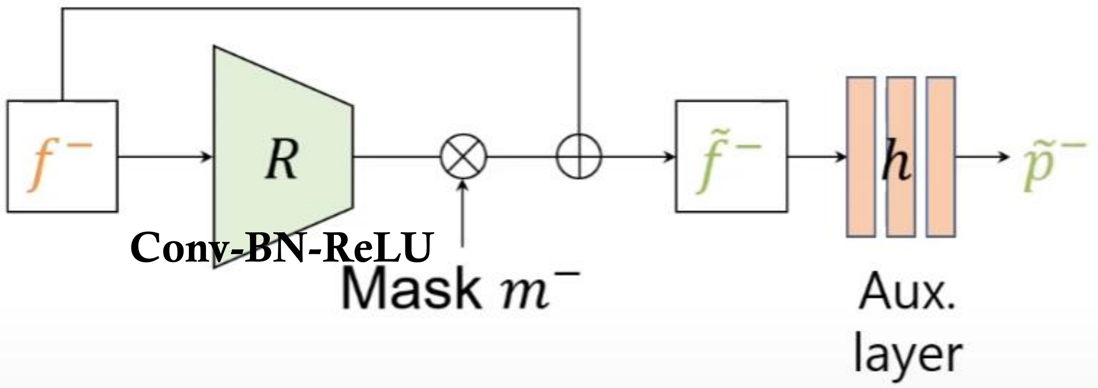


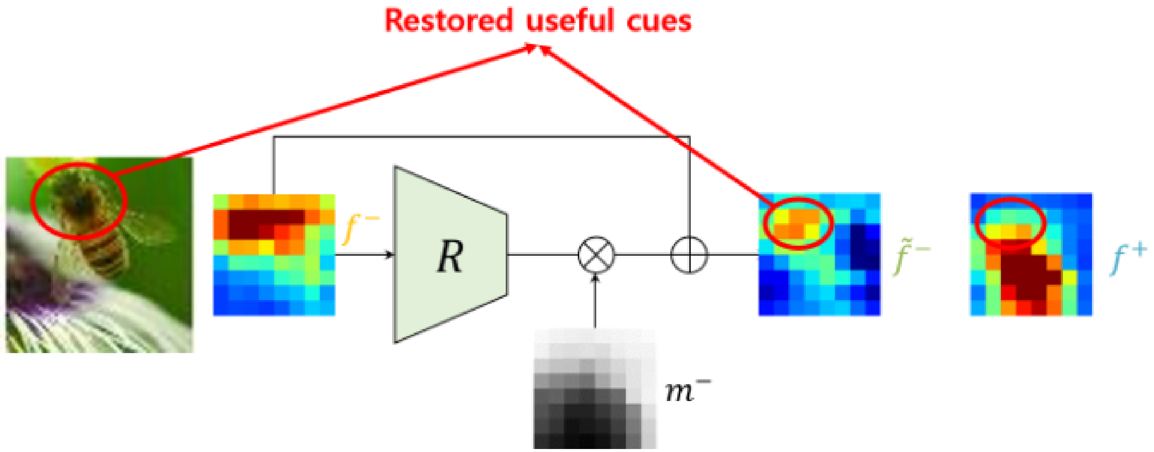
Proposed Approach

- **Feature Recalibration**
  - Guided network R to restore useful cues relevant to correct prediction
  - Input: non-robust feature map f^-
  - Output: recalibrated feature, $\tilde{f}^- = f^- + m^- \otimes R(f^-)$

$$\mathcal{L}_{rec} = -\sum_{i=1}^{N} y_i \cdot \log(\tilde{p}_i^-), \qquad (4)$$

$$\mathcal{L}_{rec} = \underbrace{\mathcal{H}}_{\text{Cross-entropy}}(\underbrace{\tilde{p}^-}_{\text{Pred.}}, \underbrace{y}_{\text{GT}})$$





Proposed Approach

- **Model Training**
  - Can be attached to any adversarial training(AT) technique with objective L_{cls}
  - FSR is highly modularized and easy to plug-in
  - Trained in an end-to-end manner

$$\mathcal{L} = \mathcal{L}_{cls} + \frac{1}{|L|} \sum_{l \in L} \left(\lambda_{sep} \cdot \mathcal{L}_{sep}^{l} + \lambda_{rec} \cdot \mathcal{L}_{rec}^{l} \right), \qquad (5)$$

  - L_{cls}: classification loss for adversarial training
  - L_{sep}: feature separation loss
  - L_{rec}: feature recalibration loss
  - λ_{sep}, λ_{rec}: hyperparameters that control weights

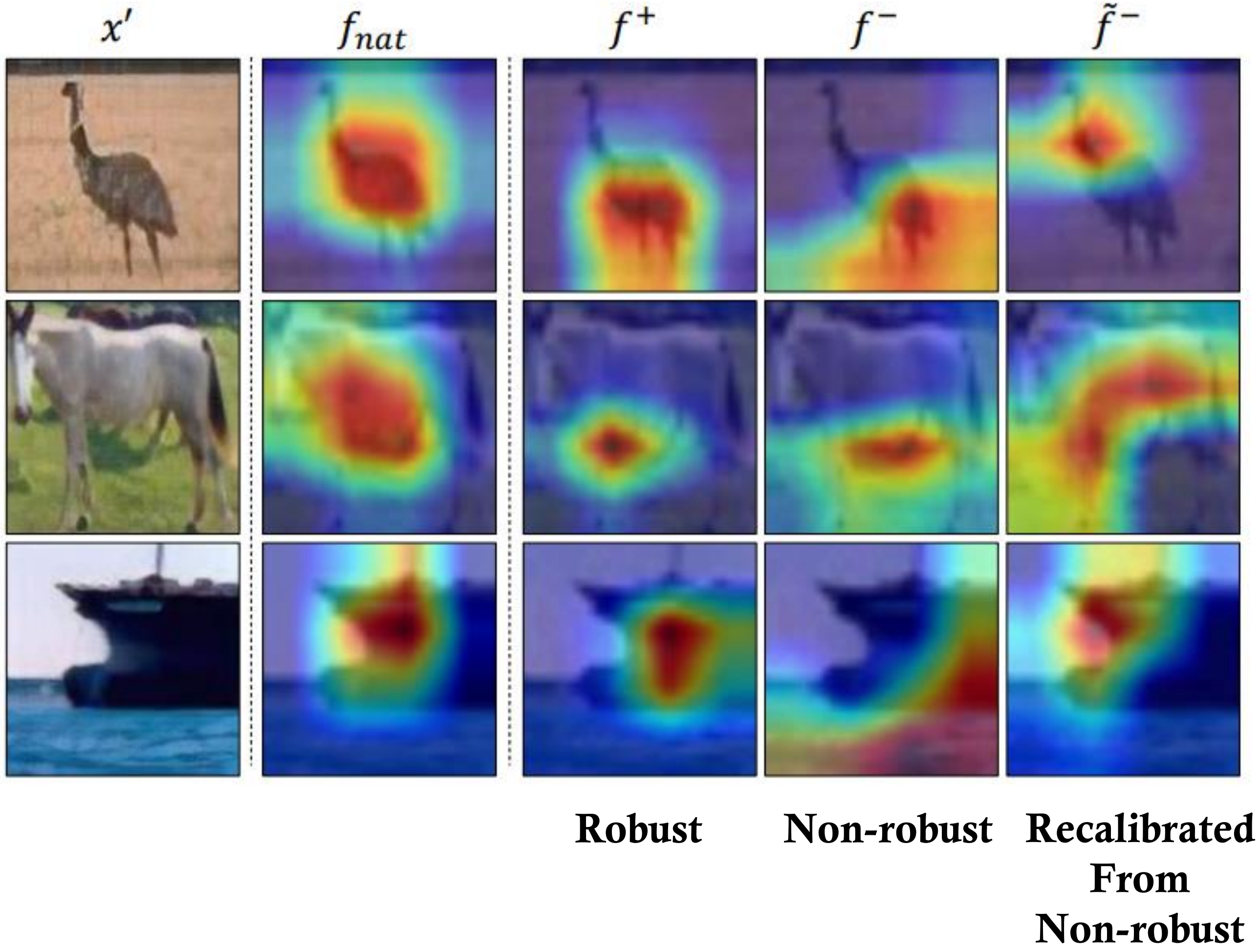
Experiments

- **Experimental Setups**
  - **Baselines**
    - PGD adversarial training (AT) [1]
    - TRADES [2]
    - MART [3]
  - **Datasets**
    - CIFAR-10/100
    - SVHN
    - Tiny ImageNet
  - **Models**
    - ResNet18
    - VGG16
    - WideResNet-34-10

[1] Madry et al., Towards deep learning models resistant to adversarial attacks. [ICLR 2018]
[2] Zhang et al., Theoretically principled trade-off between robustness and accuracy. [ICML 2019]
[3] Wang et al., Improving adversarial robustness via channel-wise activation suppressing. [ICLR 2021]

Experiments

- **Qualitative Results**



Experiments

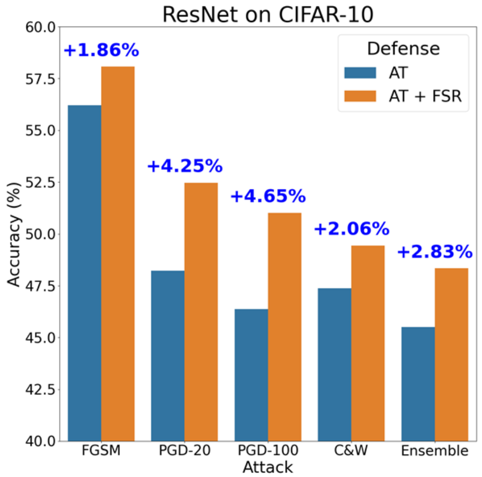
- **Application to Adversarial Training**

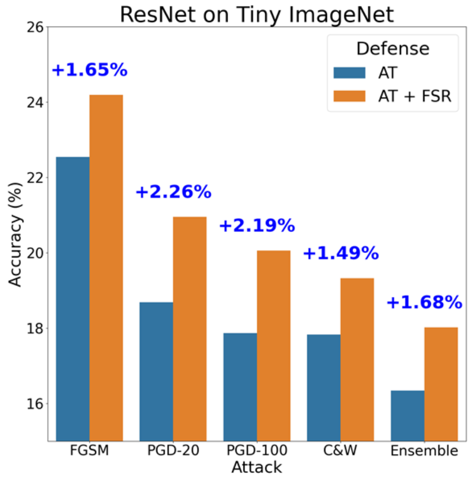
ResNet-18	CIFAR-10						SVHN					
Method	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble
AT	**85.02**	56.21	48.22	46.37	47.38	45.51	91.21	55.55	40.85	37.54	40.61	37.41
AT + FSR	81.46	**58.07**	**52.47**	**51.02**	**49.44**	**48.34**	**91.28**	**60.46**	**43.94**	**39.01**	**43.22**	**38.81**
TRADES	**86.31**	57.21	50.74	49.44	48.66	47.89	90.99	61.31	47.12	43.55	45.48	42.99
TRADES + FSR	84.49	**58.29**	**52.27**	**51.28**	**49.92**	**49.28**	**91.39**	**68.85**	**51.49**	**47.50**	**46.70**	**46.17**
MART	82.73	56.65	50.88	49.15	47.21	45.98	**90.50**	58.21	43.61	40.43	42.20	40.07
MART + FSR	**83.28**	**59.55**	**54.80**	**53.69**	**48.98**	**48.36**	89.87	**61.06**	**46.51**	**42.94**	**43.89**	**42.40**

Table 1. Robustness (accuracy (%)) of adversarial training strategies (AT, TRADES, MART) with (+ FSR) and without our FSR module against diverse white-box attacks on ResNet-18. Better results are marked in **bold**.

VGG16	CIFAR-10						SVHN					
Method	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble
AT	**80.56**	53.47	47.17	45.58	45.82	43.71	89.59	54.88	40.27	36.90	39.46	36.62
AT + FSR	80.06	**54.40**	**49.82**	**48.82**	**47.28**	**46.24**	**91.44**	**65.01**	**45.99**	**39.07**	**43.08**	**38.15**
TRADES	**82.44**	53.92	47.39	46.20	44.80	44.20	90.48	61.50	45.99	40.00	42.82	39.27
TRADES + FSR	80.78	**55.48**	**49.95**	**49.03**	**46.28**	**45.90**	**91.89**	**69.25**	**54.56**	**47.81**	**46.66**	**44.10**
MART	76.11	54.86	51.06	50.16	43.53	43.01	89.95	59.03	42.89	38.73	39.12	37.64
MART + FSR	**79.18**	**56.41**	**52.69**	**52.13**	**44.49**	**44.20**	**90.60**	**62.28**	**47.17**	**42.50**	**43.44**	**40.73**

Table 2. Robustness (accuracy (%)) of adversarial training strategies (AT, TRADES, MART) with (+ FSR) and without our FSR module against diverse white-box attacks on VGG16. Better results are marked in **bold**.





Experiments

- **Comparison with only FSR (w/o Adversarial Training)**

Feature deactivation or suppression

Method	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble	AutoAttack
AT [1]	85.02	56.21	48.22	46.37	47.38	45.51	44.11
FD [2]	85.14	56.81	48.54	46.70	47.72	45.82	44.57
CAS [3]	**85.78**	55.57	50.42	49.91	**53.47**	46.46	44.23
CIFS [4]	79.87	56.53	49.80	48.17	49.89	47.26	43.94
FSR (Ours)	81.46	**58.07**	**52.47**	**51.02**	49.44	**48.34**	**46.41**

Table 4. Comparison of robustness (accuracy (%)) between existing methods and our method. All models are trained using AT with ResNet-18 on CIFAR-10. The best results are marked in **bold**, and more comprehensive Ensemble and AutoAttack are highlighted in grey.

Method	FGSM	PGD-20	PGD-100	C&W	Ensemble	AutoAttack
AT	56.21	48.22	46.37	47.38	45.51	44.11
+ FSR	**58.07**	**52.47**	**51.02**	**49.44**	**48.34**	**46.41**
w/o Sep	57.51	50.71	48.98	49.32	47.60	45.47
w/o Rec	57.67	50.06	48.54	49.41	47.32	44.96

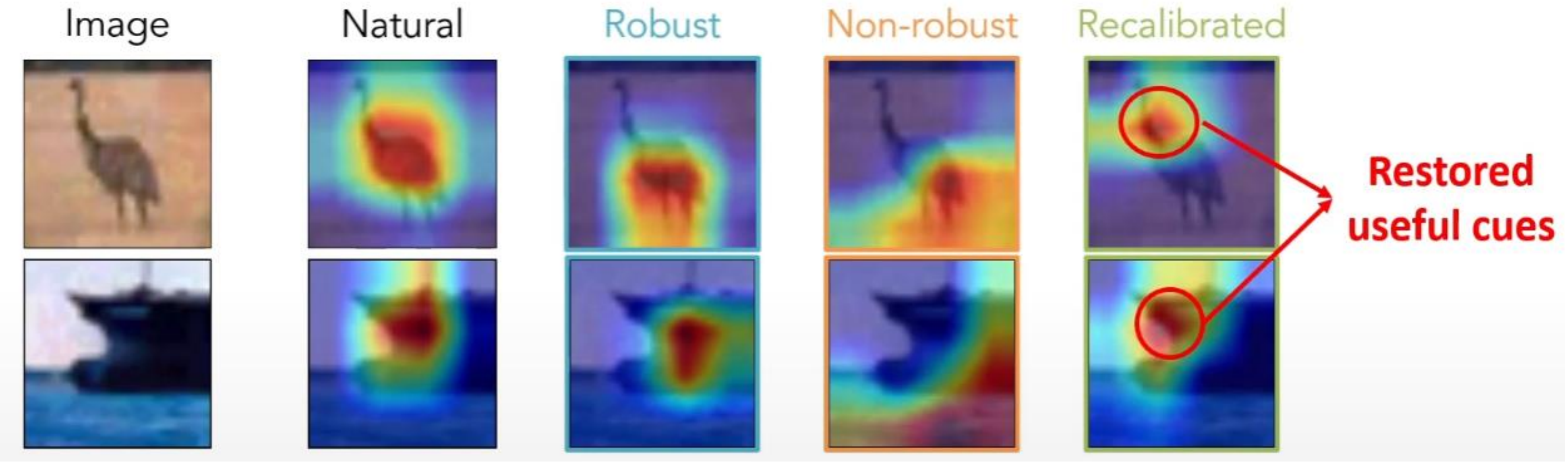
Table 6. Comparison of robustness (%) of FSR applied on AT upon removing the Separation or the Recalibration stage. Model and dataset used are ResNet-18 and CIFAR-10, respectively. Best results are marked in **bold**.

[1] Madry et al., Towards deep learning models resistant to adversarial attacks. [ICLR 2018]
[2] Xie et al., Feature Denoising for Improving Adversarial Robustness. [CVPR 2019]
[3] Wang et al., Improving adversarial robustness via channel-wise activation suppressing. [ICLR 2021]
[4] Zhang et al., CIFS: Improving Adversarial Robustness of CNNs via Channel-wise Importance-based Feature Selection [ICML 2021]

Experiments

- **Robustness of Recalibrated Feature**

	Method	(a) Classification		(b) Weighted k-NN	
		Ensemble	AutoAttack	5-NN	20-NN
Robust	f^+	47.89	45.82	66.21	61.58
Non-robust	f^-	33.11	28.39	54.69	53.89
Recalibrated	$\tilde{f}^-$	46.93	44.52	66.34	65.64
$\tilde{f}^- + f^+$	$\tilde{f}$ (Ours)	48.34	46.41	70.91	65.88



Experiments

▪ Computational Efficiency

Method	*VGG16*		*ResNet-18*	
	# Params (M)	FLOPs (G)	# Params (M)	FLOPs (G)
Vanilla	15.25	0.6299	11.17	1.1133
+ FSR	16.52	0.6701	12.43	1.1535

Table 7. Comparison of computational costs (# params and FLOPs) on a vanilla model and a model with our FSR module.

▪ Position of PSR module

	No attack	FGSM	PGD-20	PGD-100	C&W	Ensemble
`Block1`	**84.58**	56.41	48.29	46.28	46.96	44.89
`Block2`	83.76	56.34	48.86	47.03	47.32	45.28
`Block3`	82.60	56.62	50.43	49.11	47.84	46.33
`Block4`	81.46	**58.07**	**52.47**	**51.02**	**49.44**	**48.34**
`Block3 + Block4`	82.18	56.93	50.72	49.32	48.63	46.91

Table A4. Comparison of accuracy (%) as we insert our FSR module after different layers of ResNet-18.

▪ Effects of Gumbel Softmax

	FGSM	PGD-20	PGD-100	C&W	Ensemble	AutoAttack
Binary	55.78	49.21	47.79	48.74	46.91	44.26
Gumbel	**58.07**	**52.47**	**51.02**	**49.44**	**48.34**	**46.41**

Table A6. Comparison of accuracy (%) on using mask generated by discrete binary sampling or through Gumbel softmax.

Discussion

- **Limitations**
  - Assumption that the input images contain malicious perturbations designed to fool the model
  - So, FSR module occasionally decreases the natural accuracy by a small amout

ResNet-18	CIFAR-10						SVHN					
Method	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble
AT	**85.02**	56.21	48.22	46.37	47.38	45.51	91.21	55.55	40.85	37.54	40.61	37.41
AT + FSR	81.46	**58.07**	**52.47**	**51.02**	**49.44**	**48.34**	**91.28**	**60.46**	**43.94**	**39.01**	**43.22**	**38.81**
TRADES	**86.31**	57.21	50.74	49.44	48.66	47.89	90.99	61.31	47.12	43.55	45.48	42.99
TRADES + FSR	84.49	**58.29**	**52.27**	**51.28**	**49.92**	**49.28**	**91.39**	**68.85**	**51.49**	**47.50**	**46.70**	**46.17**
MART	82.73	56.65	50.88	49.15	47.21	45.98	**90.50**	58.21	43.61	40.43	42.20	40.07
MART + FSR	**83.28**	**59.55**	**54.80**	**53.69**	**48.98**	**48.36**	89.87	**61.06**	**46.51**	**42.94**	**43.89**	**42.40**

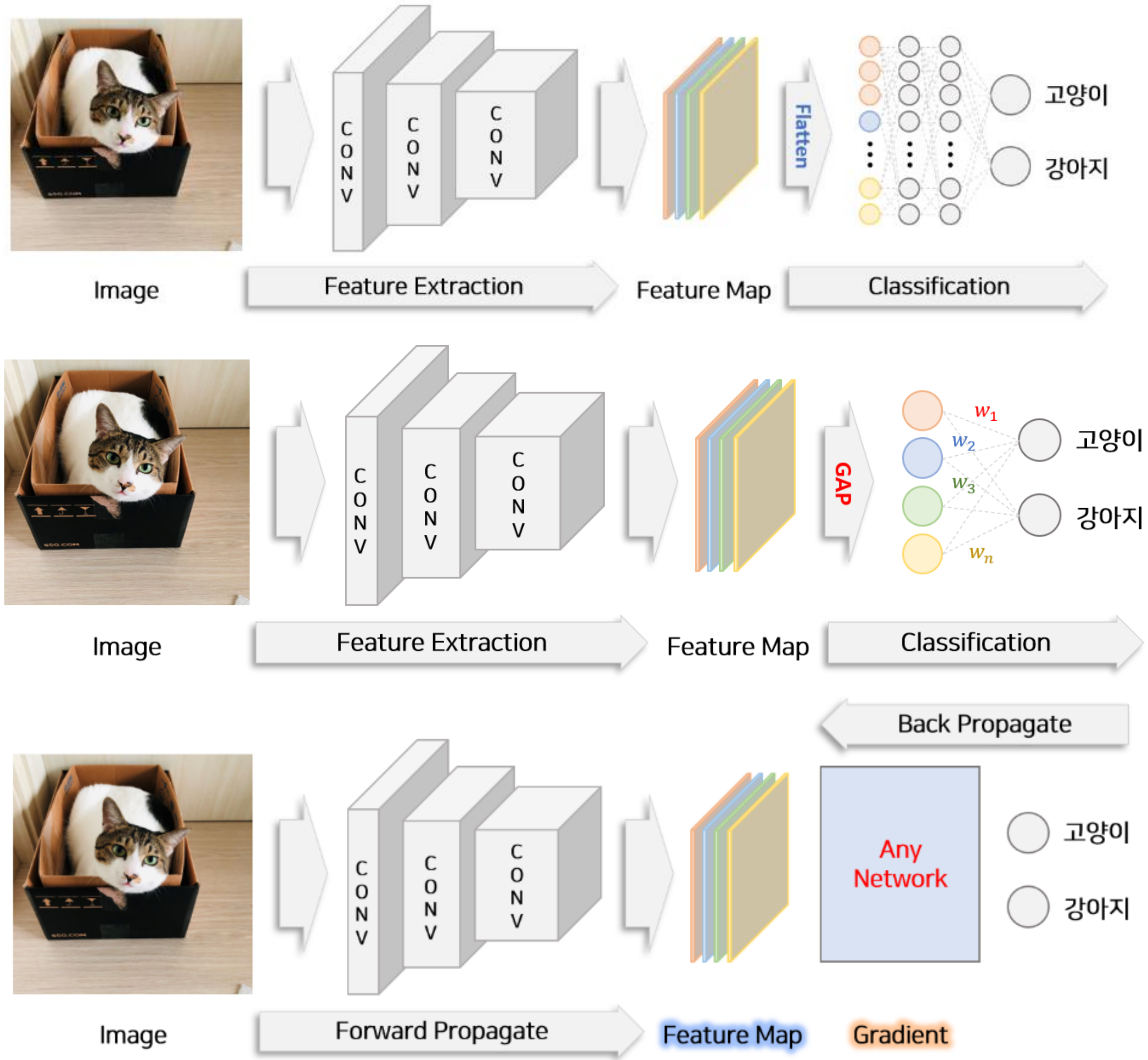
Table 1. Robustness (accuracy (%)) of adversarial training strategies (AT, TRADES, MART) with (+ FSR) and without our FSR module against diverse white-box attacks on ResNet-18. Better results are marked in **bold**.

VGG16	CIFAR-10						SVHN					
Method	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble	Natural	FGSM	PGD-20	PGD-100	C&W	Ensemble
AT	**80.56**	53.47	47.17	45.58	45.82	43.71	89.59	54.88	40.27	36.90	39.46	36.62
AT + FSR	80.06	**54.40**	**49.82**	**48.82**	**47.28**	**46.24**	**91.44**	**65.01**	**45.99**	**39.07**	**43.08**	**38.15**
TRADES	**82.44**	53.92	47.39	46.20	44.80	44.20	90.48	61.50	45.99	40.00	42.82	39.27
TRADES + FSR	80.78	**55.48**	**49.95**	**49.03**	**46.28**	**45.90**	**91.89**	**69.25**	**54.56**	**47.81**	**46.66**	**44.10**
MART	76.11	54.86	51.06	50.16	43.53	43.01	89.95	59.03	42.89	38.73	39.12	37.64
MART + FSR	**79.18**	**56.41**	**52.69**	**52.13**	**44.49**	**44.20**	**90.60**	**62.28**	**47.17**	**42.50**	**43.44**	**40.73**

Table 2. Robustness (accuracy (%)) of adversarial training strategies (AT, TRADES, MART) with (+ FSR) and without our FSR module against diverse white-box attacks on VGG16. Better results are marked in **bold**.

Thank you!

Grad CAM


CONV
CONV
CONV
Flatten
고양이
강아지
Image
Feature Extraction
Feature Map
Classification
CONV
CONV
CONV
GAP
w_1
w_2
w_3
w_n
고양이
강아지
Image
Feature Extraction
Feature Map
Classification
Back Propagate
CONV
CONV
CONV
Any Network
고양이
강아지
Image
Forward Propagate
Feature Map
Gradient

Gumbel Softmax

Gumbel-Max trick

Let $\pi_1, \pi_2, \ldots, \pi_n$ be probabilities, i.e., $\sum_k \pi_k = 1$

We define $Z = \underset{k}{\operatorname{argmax}}\{\log\pi_k + G_k\}$ where $G_1, \ldots, G_n$ i.i.d. $\sim Gumbel(0,1)$

Then, $\mathbb{P}(Z = k) = \pi_k$

Proof. Let $u_k = \log\pi_k + G_k$

$$
\begin{aligned}
\mathbb{P}(Z = k) &= \mathbb{P}(u_k \geq u_j, \forall j \neq k) \\
&= \int_{-\infty}^{\infty} \mathbb{P}(u_k \geq u_j, \forall j \neq k \mid u_k)\mathbb{P}(u_k)du_k \\
&= \int_{-\infty}^{\infty} \prod_{j \neq k} \mathbb{P}(u_k \geq u_j \mid u_k)\mathbb{P}(u_k)du_k \\
&= \int_{-\infty}^{\infty} \prod_{j \neq k} e^{-e^{-u_k+\log\pi_j}} e^{-(u_k-\log\pi_k+e^{-(u_k-\log\pi_k)})} du_k \\
&= \int_{-\infty}^{\infty} e^{-\sum_{j\neq k}\pi_j e^{-u_k}} \pi_k e^{-(u_k+\pi_k e^{-u_k})} du_k \\
&= \pi_k \int_{-\infty}^{\infty} e^{-u_k-(\pi_k+\sum_{j\neq k}\pi_j)e^{-u_k}} du_k = \pi_k
\end{aligned}
$$

$Gumbel(0,1)$

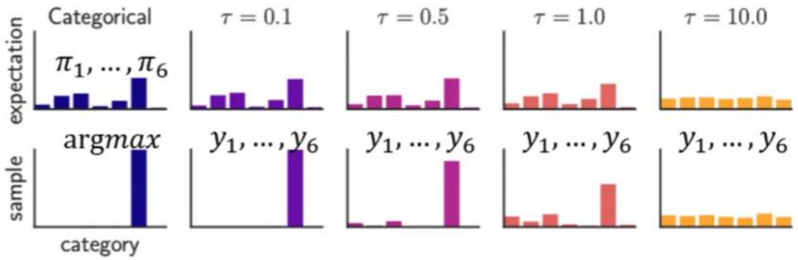
PDF $f(x) = e^{-x+e^{-x}}$

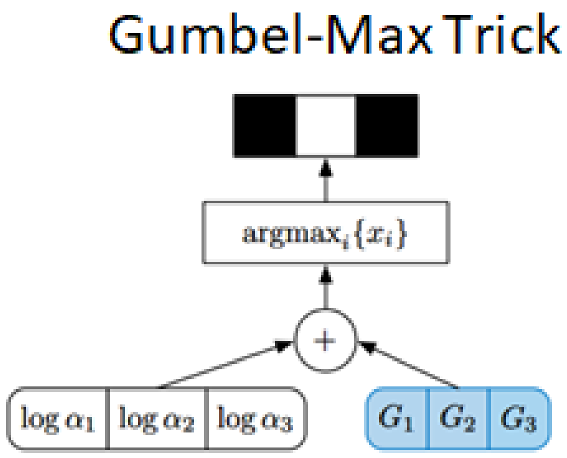
CDF $F(x) = e^{-e^{-x}}$

Gumbel-Softmax trick

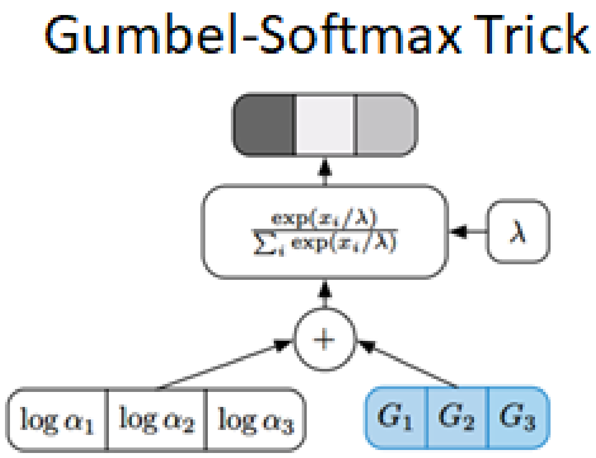
$$\underset{k}{\operatorname{argmax}}\{\log\pi_k + G_k\} \rightarrow y_i = \frac{e^{(\log\pi_i + G_i)/\tau}}{\sum_{k=1}^{n} e^{(\log\pi_k + G_k)/\tau}}$$

for $i = 1, \ldots, n$ where $\tau > 0$ is softmax temperature





(a) Discrete(α)



(b) Concrete(α, λ)