Attacking Adversarial Attacks as A Defense

박경찬

고려대학교 산업경영공학과

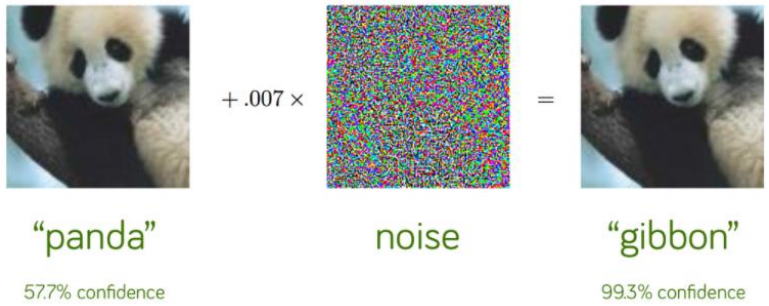
DSBA 연구실

INDEX

Adversarial Defense: Modeling & Denosing

❖ Adversarial Attack & Defense

- Adversarial Attack은 이미지에 아주 작은 noise를 추가하여 모델의 결과를 심각하게 훼손시키는 방법론
- 특히, White Box Attack은 모델의 weight에 기반해 noise를 생성하기 때문에 아주 강력한 공격 방법론임

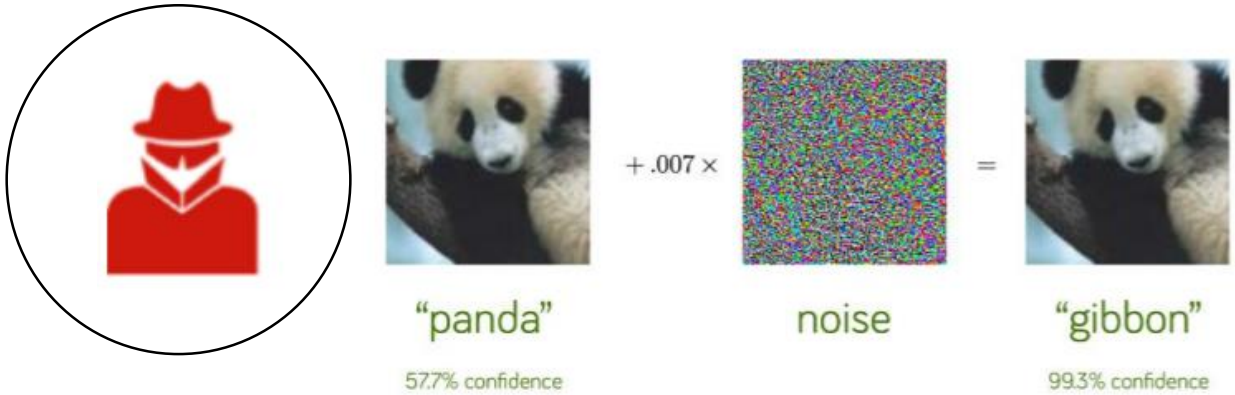


- Adversarial Attack은 해당 noise를 잘 생성하는 것이 목적임
  - $\mathbf{max}_{\delta}\, \mathbf{l}(\mathbf{x}+\boldsymbol{\delta}, \mathbf{y}, \boldsymbol{\theta}), \mathbf{subject\ to}\ ||\boldsymbol{\delta}||_{\mathbf{p}} \leq \boldsymbol{\epsilon}$
- Adversarial Defense는 가장 강한 noise를 최대한 방어하는 것이 목적임
  - $\mathbf{min}_{\boldsymbol{\theta}}[\mathbf{max}_{\boldsymbol{\delta}}\, \mathbf{l}(\mathbf{x}+\boldsymbol{\delta}, \mathbf{y}, \boldsymbol{\theta}), \mathbf{subject\ to}\ \left|\left|\boldsymbol{\delta}\right|\right|_{\mathbf{p}} \leq \boldsymbol{\epsilon}]$

Adversarial Defense: Modeling & Denosing

❖ Value of Adversarial Defense

- White Box Adversarial Attack은 attacker가 모델의 모든 정보를 알고 있다는 매우 극단적인 가정을 전제로 함
- 따라서 그런 White Box Attack을 막고자 하는 Adversarial Defense 연구가 과연 실용성이 있을까 하는 의문이 있을 수 있음



Adversarial Defense: Modeling & Denosing

❖ Value of Adversarial Defense

- ❖ Adversarial Attack은 딥러닝 모델이 작은 noise에도 쉽게 망가질 수 있다는 것을 보여주는 장치
- ▪ 즉 Adversarial Defense의 궁극적인 목적은 모델이 noise에 강건하게 만드는 것, 즉 신뢰성 높은 모델을 개발하는 것임
  - ➢ $\mathbf{min}_{\theta}[\mathbf{max}_{\delta}\, \mathbf{l}(\mathbf{x}+\boldsymbol{\delta}, \mathbf{y}, \boldsymbol{\theta}), \mathbf{subject\ to}\ \left|\left|\boldsymbol{\delta}\right|\right|_{p} \leq \boldsymbol{\epsilon}]$

- ▪ 특히 noise가 발생할 확률이 높으면서 오작동시 큰 손해를 발생시킬 수 있는 분야에서는 더욱 중요한 연구 과제임 (Ex: 자율주행..)
- ▪ 따라서 연구와 실용의 관점 모두에서 중요한 연구 분야라고 할 수 있음

Adversarial Defense: Modeling & Denosing

❖ Value of Adversarial Defense

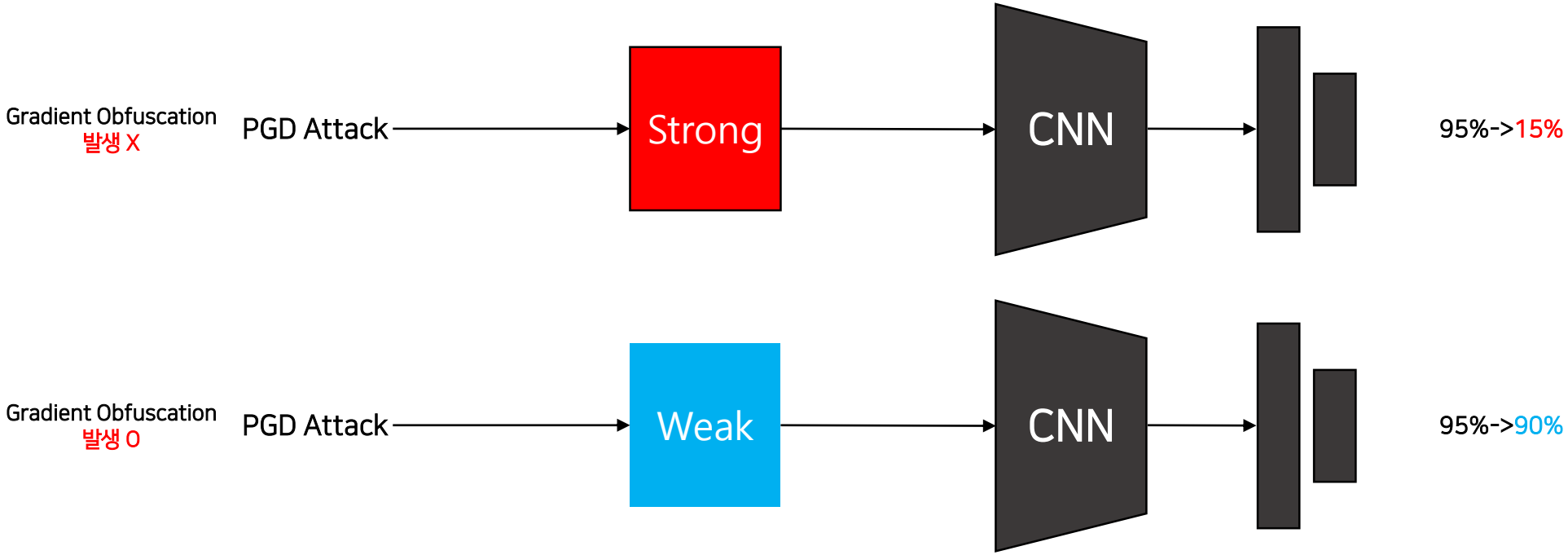
- Adversarial Attack은 딥러닝 모델이 작은 noise에도 쉽게 망가질 수 있다는 것을 보여주는 장치
- 즉 Adversarial Defense의 궁극적인 목적은 모델이 noise에 강건하게 만드는 것, 즉 신뢰성 높은 모델을 개발하는 것임
  - ➢ $\mathbf{min}_{\theta}[\mathbf{max}_{\delta}\, \mathbf{l}(\mathbf{x}+\boldsymbol{\delta}, \mathbf{y}, \boldsymbol{\theta}), \mathbf{subject\ to}\ \left|\left|\boldsymbol{\delta}\right|\right|_{p} \leq \boldsymbol{\epsilon}]$

- 특히 noise가 발생할 확률이 높으면서 오작동시 큰 손해를 발생시킬 수 있는 분야에서는 더욱 중요한 연구 과제임 (Ex: 자율주행..)
- 따라서 연구와 실용의 관점 모두에서 중요한 연구 분야라고 할 수 있음
- **그럼에도 불구하고 매우 극단적인 가정을 가지고 있고, 따라서 어렵다.**

Adversarial Defense: Modeling & Denosing

❖ Adversarial Training

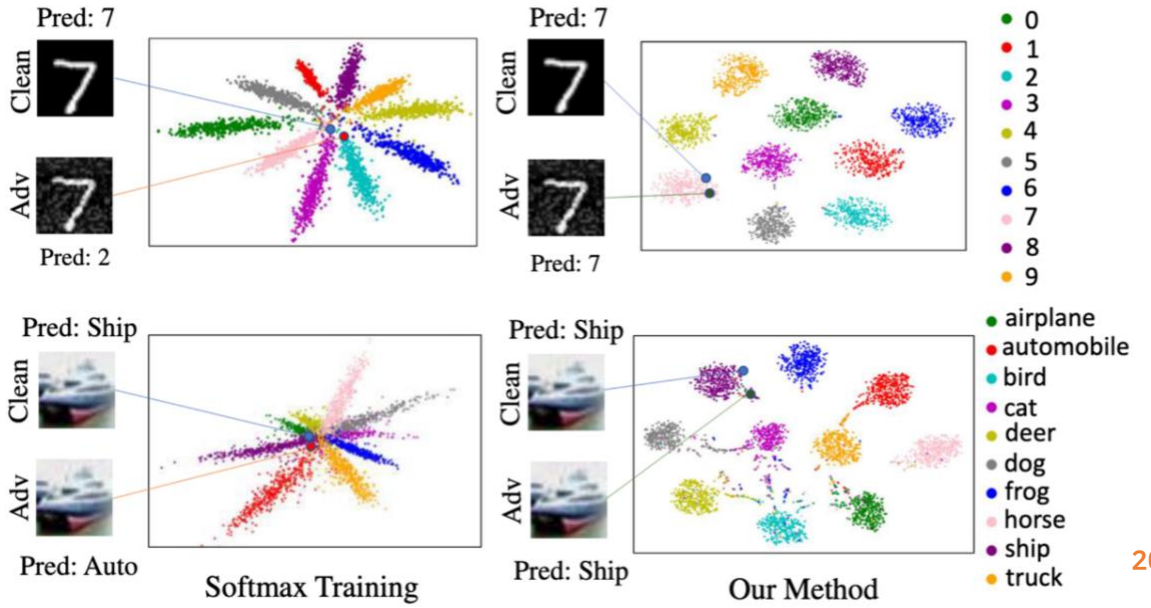
- Adversarial Training은 현재 Adversarial Defesnse의 유일한 방법론
- 그 밖의 다른 방법론들은 Gradient Obfuscation 문제가 발생함
  - ➢ Gradient Obfuscation => [Robust Model X, Weak Noise O]
  - ➢ 즉 기존의 Attack 방법론들이 잘 작동하지 않았을 뿐 minmax 문제를 푼 것이 아님



Adversarial Defense: Modeling & Denosing

❖ Adversarial Training

- Adversarial Training은 현재 Adversarial Defesnse의 유일한 방법론
- 그 밖의 다른 방법론들은 Gradient Obfuscation 문제가 발생함
  - ➢ Gradient Obfuscation => [Robust Model X, Weak Noise O]
  - ➢ 즉 기존의 Attack 방법론들이 잘 작동하지 않았을 뿐 minmax 문제를 푼 것이 아님



2020/08/14 서승완 박사 과정 세미나 참고

Adversarial Defense: Modeling & Denosing

❖ Adversarial Training

- Auto Attack이라는 보다 강력한 기법이 등장하여 결국 Adversarial Training 기반의 방법론들만 효과가 있음을 실험을 통해 증명함

Table 2. **Robustness evaluation of adversarial defenses by** *AutoAttack*. We report clean test accuracy, the robust accuracy of the individual attacks as well as the combined one of *AutoAttack* (AA column). We also provide the robust accuracy reported in the original papers and compute the difference to the one of *AutoAttack*. If negative (in red) *AutoAttack* provides lower (better) robust accuracy.

#	paper	clean	$APGD_{CE}$	$APGD^{T}_{DLR}$	FAB^{T}	Square	AA	reported	reduct.
CIFAR-10 - l_∞ - $\epsilon = 8/255$									
1	(Carmon et al., 2019)	89.69	61.74	**59.54**	60.12	66.63	59.53	62.5	-2.97
2	(Alayrac et al., 2019)	86.46	60.17	**56.27**	56.81	66.37	56.03	56.30	-0.27
3	(Hendrycks et al., 2019)	87.11	57.23	**54.94**	55.27	61.99	54.92	57.4	-2.48
4	(Rice et al., 2020)	85.34	57.00	**53.43**	53.83	61.37	53.42	58	-4.58
5	(Qin et al., 2019)	86.28	55.70	52.85	53.28	60.01	52.84	52.81	0.03
6	(Engstrom et al., 2019)	87.03	51.72	**49.32**	49.81	58.12	49.25	53.29	-4.04
7	(Kumari et al., 2019)	87.80	51.80	**49.15**	49.54	58.20	49.12	53.04	-3.92
8	(Mao et al., 2019)	86.21	49.65	**47.44**	47.91	56.98	47.41	50.03	-2.62
9	(Zhang et al., 2019a)	87.20	46.15	**44.85**	45.39	55.08	44.83	47.98	-3.15
10	(Madry et al., 2018)	87.14	44.75	**44.28**	44.75	53.10	44.04	47.04	-3.00
11	(Pang et al., 2020)	80.89	57.07	**43.50**	44.06	49.73	43.48	55.0	-11.52
12	(Wong et al., 2020)	83.34	45.90	**43.22**	43.74	53.32	43.21	46.06	-2.85
13	(Shafahi et al., 2019)	86.11	43.66	**41.64**	43.44	51.95	41.47	46.19	-4.72
14	(Ding et al., 2020)	84.36	50.12	**41.74**	42.47	55.53	41.44	47.18	-5.74
15	(Moosavi-Dezfooli et al., 2019)	83.11	41.72	**38.50**	38.97	47.69	38.50	41.4	-2.90
16	(Zhang & Wang, 2019)	89.98	64.42	**37.29**	38.48	59.12	36.64	60.6	-23.96
17	(Zhang & Xu, 2020)	90.25	71.40	**37.54**	38.99	66.88	36.45	68.7	-32.25
18	(Jang et al., 2019)	78.91	37.76	**34.96**	35.50	44.33	34.95	37.40	-2.45
19	(Kim & Wang, 2020)	91.51	56.64	35.93	**35.41**	61.30	34.22	57.23	-23.01
20	(Moosavi-Dezfooli et al., 2019)	80.41	36.65	**33.70**	34.08	43.46	33.70	36.3	-2.60
21	(Wang & Zhang, 2019)	92.80	59.09	33.61	**31.19**	64.22	29.35	58.6	-29.25
22	(Wang & Zhang, 2019)	92.82	69.62	29.73	**29.10**	66.77	26.93	66.9	-39.97
23	(Mustafa et al., 2019)	89.16	8.16	1.13	**0.71**	33.91	0.28	32.32	-32.04
24	(Chan et al., 2020)	93.79	2.06	**0.53**	58.13	71.43	0.26	15.5	-15.24
25	(Pang et al., 2020)	93.52	89.48	**0.00**	**0.00**	35.82	0.00	31.4	-31.40

Adversarial Defense: Modeling & Denosing

❖ Adversarial Training

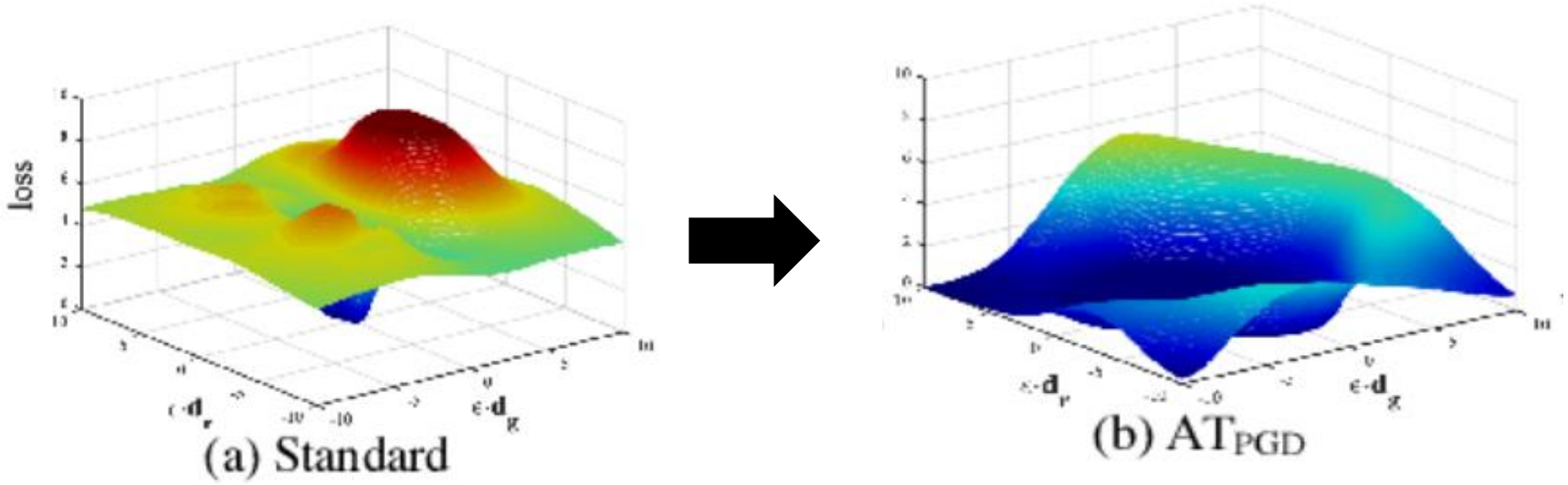
- Auto Attack이라는 보다 강력한 기법이 등장하여 결국 Adversarial Training 기반의 방법론들만 효과가 있음을 실험을 통해 증명함

#	paper	model	architecture	clean	report.	AA
1	(Gowal et al., 2020)‡	*available*	WRN-70-16	91.10	65.87	65.88
2	(Gowal et al., 2020)‡	*available*	WRN-28-10	89.48	62.76	62.80
3	(Wu et al., 2020a)‡	*available*	WRN-34-15	87.67	60.65	60.65
4	(Wu et al., 2020b)‡	*available*	WRN-28-10	88.25	60.04	60.04
5	(Carmon et al., 2019)‡	*available*	WRN-28-10	89.69	62.5	59.53
6	(Gowal et al., 2020)	*available*	WRN-70-16	85.29	57.14	57.20
7	(Sehwag et al., 2020)‡	*available*	WRN-28-10	88.98	-	57.14
8	(Gowal et al., 2020)	*available*	WRN-34-20	85.64	56.82	56.86
9	(Wang et al., 2020)‡	*available*	WRN-28-10	87.50	65.04	56.29
10	(Wu et al., 2020b)	*available*	WRN-34-10	85.36	56.17	56.17

Adversarial Defense: Modeling & Denosing

❖ Adversarial Training

- Adversarial Defesne를 한다는 것은 minmax를 문제를 푸는 것이고, 이는 input loss surface를 평평하게 한다는 의미와 같음
- Adversarial Training은 학습 단계에서 각 batch마다 adversarial example을 생성하여 학습 데이터에 추가함으로써 이를 달성함
  - ➢ 따라서 Adversarial Training도 효과적이기 위해서는 충분히 strong한 attack을 통해 adversarial example을 생성해야 함
  - ➢ PGD-10 Attack이면 충분한 것이 실험적으로 증명됨 (FGSM과 같은 week attack을 사용하면 효과가 없음)



Adversarial Defense: Modeling & Denosing

❖ Adversarial Defense: Modeling

- Input loss surface가 가장 평평하도록 하는 모델(or 함수) f는 상수 함수임
  - ➢ 만약 모델이 상수 함수라면 adversarial attack으로 인해 output의 결과가 바뀔 수 없음
  - ➢ 하지만 classification 성능 또한 극단적으로 제한됨
- 즉 모델(함수)이 adversarial attack에 영향을 받지 않기 위해서는 복잡성이 낮아야 하는데, Classification 문제를 풀기 위해서는 복잡성이 높아야 하므로 모순이 발생함
- 따라서 모델링 관점에서 adversarial defense 문제를 해결하는 것은 매우 어려움

#	paper	model	architecture	clean	report.	AA
1	(Gowal et al., 2020)‡	*available*	WRN-70-16	91.10	65.87	65.88
2	(Gowal et al., 2020)‡	*available*	WRN-28-10	89.48	62.76	62.80
3	(Wu et al., 2020a)‡	*available*	WRN-34-15	87.67	60.65	60.65
4	(Wu et al., 2020b)‡	*available*	WRN-28-10	88.25	60.04	60.04
5	(Carmon et al., 2019)‡	*available*	WRN-28-10	89.69	62.5	59.53
6	(Gowal et al., 2020)	*available*	WRN-70-16	85.29	57.14	57.20

Adversarial Defense: Modeling & Denosing

❖ Adversarial Defense: Denosing

- 따라서 다른 관점에서의 접근이 필요한데 가장 이상적인 것은 Denosing 기반의 방법론임
- 즉 Denosing을 통해 Adversarial Noise를 제거 혹은 그 효과를 완화하도록 하는 것
  - $\mathbf{min}_{\delta_2}[\mathbf{max}_{\delta}\, \mathbf{l}(\mathbf{x}+\boldsymbol{\delta}+\boldsymbol{\delta}_2, \mathbf{y}, \boldsymbol{\theta}), \mathbf{subject\ to}\ \left|\left|\boldsymbol{\delta}\right|\right|_{\mathbf{p}} \leq \boldsymbol{\epsilon}]$
- 모델링 관점에서의 접근법에서 발생하는 모순에서 완전히 자유롭다는 장점이 있지만, 이 또한 어려움
  - Adversrial Example에 적용됬을 때 Adversarial Attack의 효과를 지울 수 있어야함
  - Original image에 적용됬을 때 아무 효과도 없어야 함
- 따라서 목적에 부합하는 noise를 어떻게 구할 것인지에 대한 논리가 필요함

❖ Attacking Adversarial Attack as A Defense

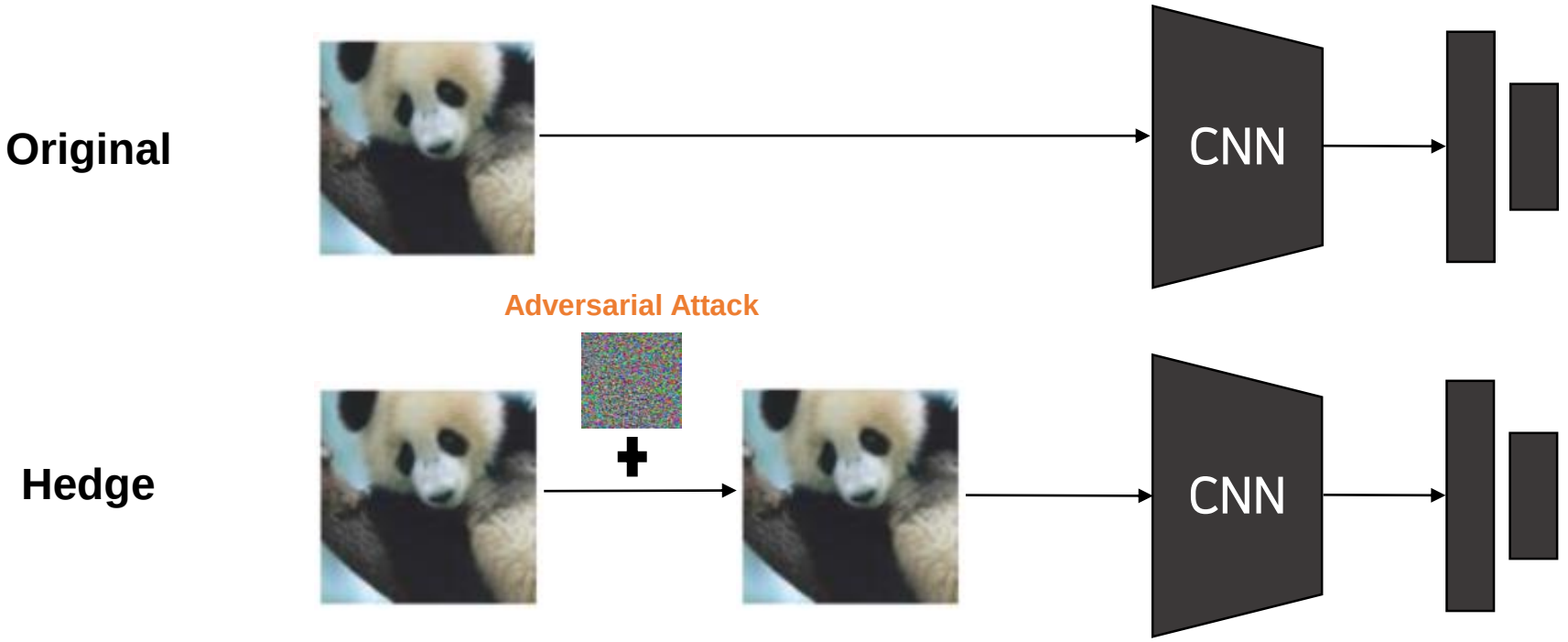
- 본 논문은 Adversarial Training+Denosing 방법론을 제안함
- Adversarial noise가 강건하지 못하다는 것을 지적하며, 실제로 random noise도 denosing 효과가 있음을 실험을 통해 보임
- Adversarial Attack 기법을 통해 adversarial noise를 공격하는 Hedge Defense 방법론을 제안함

Table 1: Robust accuracy (%) before perturbation (-), after random noise (Random), and after Hedge Defense (Hedge) on CIFAR10. Eight robust models are showed here. Evaluations are repeated three times with the mean values presented. Extra models and variances are provided in Appendix B.4.

Model	Method	Nat-Acc.	PGD	C&W	Deep Fool	APGD CE	APGD T	FAB	Square	RayS	Auto Attack	Worst Case
WA+ SiLU Gowal et al. (2020)	-	91.10	69.16	67.48	71.23	68.24	66.17	66.70	71.76	72.03	66.16	66.15
	Random	90.82	69.34	71.35	75.65	69.46	67.45	78.82	77.58	79.23	67.76	66.24
	Hedge	90.62	**71.43**	**78.42**	**79.64**	**73.86**	**73.00**	**82.28**	**83.30**	**82.03**	**72.66**	**68.61**

❖ Attacking Adversarial Attack as A Defense

- 본 논문은 Adversarial Training+Denosing 방법론을 제안함
- Adversarial noise가 강건하지 못하다는 것을 지적하며, 실제로 random noise도 denosing 효과가 있음을 실험을 통해 보임
- Adversarial Attack 기법을 통해 adversarial noise를 공격하는 Hedge Defense 방법론을 제안함



Paper

❖ Attacking Adversarial Attack as A Defense

- 기존 Adversarial Attack

  1) CE Loss: $-\log(\mathrm{p}(y_c))$

  2) KL Loss: $\sum_{i=1}^{i=C} \mathrm{p}(y_c) * \log(\frac{\mathrm{p}(y_c)}{\mathrm{q}(y_c)})$

- Attacking Adversarial Attack

  1) CE Loss => 라벨 정보가 필요하므로 불가

  2) KL Loss

❖ Attacking Adversarial Attack as A Defense

- 기존 Adversarial Attack

  1) CE Loss: $-\log(\mathrm{p}(y_c))$

  2) KL Loss: $\sum_{i=1}^{i=C} \mathrm{p}(y_c) * \log(\frac{\mathrm{p}(y_c)}{\mathrm{q}(y_c)})$

- Attacking Adversarial Attack

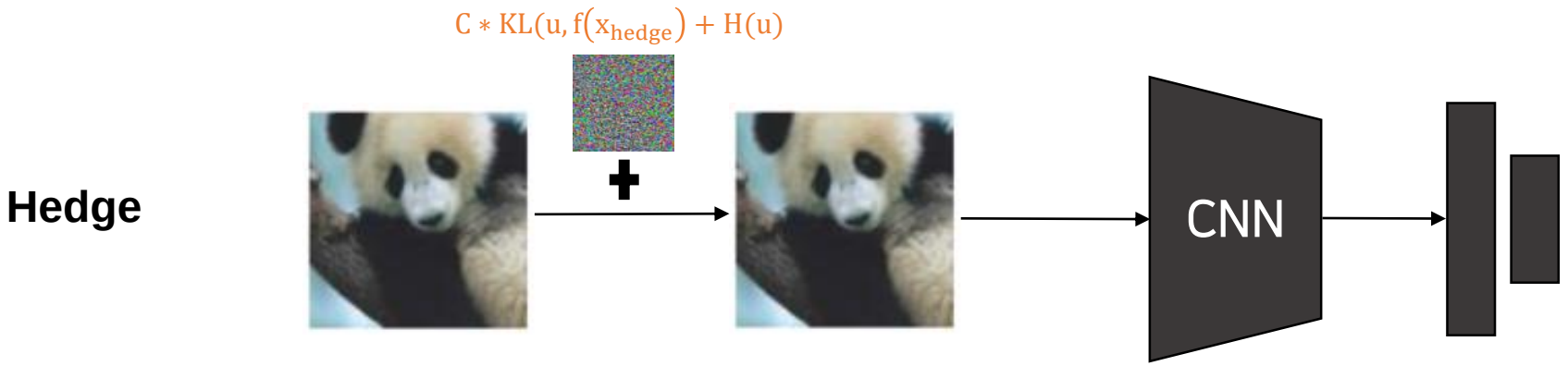
  1) CE Loss => 라벨 정보가 필요하므로 불가

  2) KL Loss => Adversaial Example에는 효과적이지만, Original 이미지를 매우 망가뜨림

WRN-34-20	Original	PGD-100
Only Adversarial Training	87.34%	53.49%
+Denosing [FGSM $\epsilon = 0.03$]	61.19%	84.25%

❖ Attacking Adversarial Attack as A Defense

- 본 논문에서는 Adversarial Example에만 효과적으로 적용되는 Hedge Defense라는 방법론을 제안함
- Hedge Defense는 모든 class에 대해 adversarial attack을 실시함
  - ➢ Hedge Defense: $C * \sum_{i=1}^{i=C} \frac{1}{C} * -\log(p(y_i)) - \log(\frac{1}{C})$ ➔ $C * KL(u, f(x_{hedge}) + H(u)$
- 즉 Hedge Defense는 균등 분포와의 KL Divergence가 커지도록 noise를 계산하여 input에 추가함



❖ Attacking Adversarial Attack as A Defense

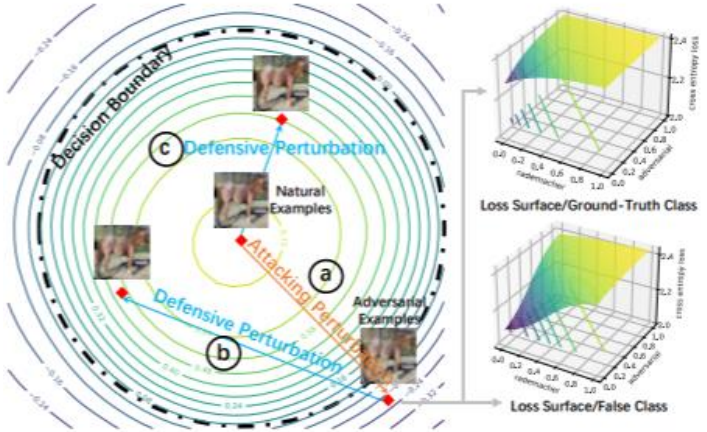
- 본 논문에서는 Adversarial Example에만 효과적으로 적용되는 Hedge Defense라는 방법론을 제안함
- Hedge Defense가 사용하는 loss는 아래와 같음
  - ➢ Hedge Defense: $C * \sum_{i=1}^{i=C} \frac{1}{C} * -\log(p(y_i)) - \log(\frac{1}{C})$ ➔ $C * KL(u, f(x_{hedge}) + H(u)$
- 즉 Hedge Defense는 균등 분포와의 KL Divergence가 커지도록 noise를 계산하여 input에 추가함
- Hedge Defense의 noise는 adversarial example에만 영향을 미침

Table 1: Robust accuracy (%) before perturbation (-), after random noise (Random), and after Hedge Defense (Hedge) on CIFAR10. Eight robust models are showed here. Evaluations are repeated three times with the mean values presented. Extra models and variances are provided in Appendix B.4.

Model	Method	Nat-Acc.	PGD	C&W	Deep Fool	APGD CE	APGD T	FAB	Square	RayS	Auto Attack	Worst Case
WA+ SiLU Gowal et al. (2020)	-	91.10	69.16	67.48	71.23	68.24	66.17	66.70	71.76	72.03	66.16	66.15
	Random	90.82	69.34	71.35	75.65	69.46	67.45	78.82	77.58	79.23	67.76	66.24
	Hedge	90.62	**71.43**	**78.42**	**79.64**	**73.86**	**73.00**	**82.28**	**83.30**	**82.03**	**72.66**	**68.61**

❖ Attacking Adversarial Attack as A Defense

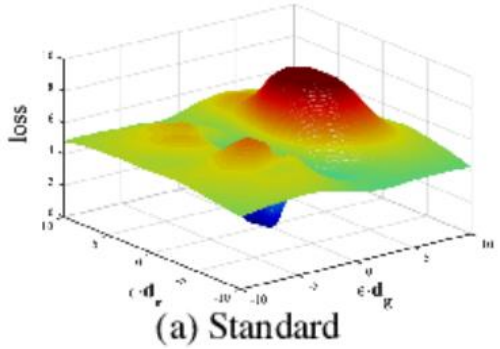
- Hedge Defense가 작동하는 이유는 ground-truth class 주변의 loss surface가 다른 class에 비해 smoother하기 때문임



- 즉 모든 class에 대해 adversarial attack을 가하면 상대적으로 덜 민감한 ground-truth class가 영향을 가장 적게 받기 때문임
- 따라서 original image의 output에는 영향을 미치지 않고, adversarial image에만 효과적으로 작동함
- 단, 해당 방법론은 adversarial training 방법과 함께 사용해야 함

❖ Attacking Adversarial Attack as A Defense

- Adversarial Training을 해야만 ground-truth class의 loss surface가 상대적으로 smooth함



Standard Training

0.99	0.01	0	0	0	0	0	0	0	0	0	0.99	0.01	0	0	0	0	0	0	0	0

Adversarial Training

0.79	0.15	0.06	0	0	0	0	0	0	0	0.40	0.50	0.10	0	0	0	0	0	0	0	0

❖ Experiment

- 본 논문에서는 다양한 Adversarial Training 기법에 Hedge Defense를 추가로 적용하여 실험을 진행함
- Hedge Defense를 하기 위한 adversarial attack으로 PGD-20 Attack을 사용함 $\epsilon = 8/255$

Table 1: Robust accuracy (%) before perturbation (-), after random noise (Random), and after Hedge Defense (Hedge) on CIFAR10. Eight robust models are showed here. Evaluations are repeated three times with the mean values presented. Extra models and variances are provided in Appendix B.4.

Model	Method	Nat-Acc.	PGD	C&W	Deep Fool	APGD CE	APGD T	FAB	Square	RayS	Auto Attack	Worst Case
WA+ SiLU Gowal et al. (2020)	-	91.10	69.16	67.48	71.23	68.24	66.17	66.70	71.76	72.03	66.16	66.15
	Random	90.82	69.34	71.35	75.65	69.46	67.45	78.82	77.58	79.23	67.76	66.24
	Hedge	90.62	**71.43**	**78.42**	**79.64**	**73.86**	**73.00**	**82.28**	**83.30**	**82.03**	**72.66**	**68.61**
AWP Wu et al. (2020b)	-	88.25	64.14	61.44	64.81	63.56	60.49	60.97	66.15	66.93	60.50	60.46
	Random	87.91	64.18	65.01	70.41	64.53	61.54	73.37	72.47	74.89	62.00	60.42
	Hedge	86.98	**66.16**	**71.94**	**73.83**	**68.18**	**66.47**	**76.37**	**76.69**	**76.80**	**66.68**	**62.72**
RST Carmon et al. (2019)	-	89.69	63.17	61.74	66.69	62.01	60.14	60.66	66.91	67.61	60.13	60.08
	Random	89.50	63.47	65.90	71.16	63.42	61.49	75.48	74.51	77.02	61.89	60.09
	Hedge	88.64	**66.38**	**73.89**	**75.25**	**68.37**	**68.09**	**78.48**	**79.24**	**78.98**	**67.46**	**63.10**
Pre-Training Hendrycks et al. (2019a)	-	87.11	58.19	57.09	59.77	57.52	55.32	55.68	62.38	63.32	55.30	55.28
	Random	86.93	58.52	61.79	70.4	58.59	56.45	71.99	71.16	74.56	56.79	55.32
	Hedge	87.43	**62.01**	**73.12**	**75.50**	**65.63**	**65.38**	**76.62**	**78.32**	**78.65**	**64.45**	**59.17**
MART Wang et al. (2020)	-	87.50	63.49	59.62	63.14	61.83	56.73	57.39	64.88	65.66	56.74	56.70
	Random	87.08	63.59	63.82	72.26	63.02	58.10	73.04	72.11	74.71	58.75	56.99
	Hedge	86.18	**64.56**	**71.71**	**73.86**	**67.00**	**63.30**	**74.35**	**76.09**	**75.60**	**63.42**	**59.35**
HYDRA Sehwag et al. (2020)	-	88.98	60.95	62.19	64.66	59.86	57.67	58.41	65.02	65.61	57.67	57.63
	Random	88.80	61.42	66.08	69.50	61.25	59.03	74.60	73.26	76.20	59.55	57.85
	Hedge	87.82	**64.19**	**73.05**	**73.43**	**66.15**	**65.56**	**76.94**	**77.58**	**77.36**	**65.07**	**61.13**
TRADES Zhang et al. (2019b)	-	84.92	55.83	54.47	58.15	55.08	53.09	53.56	59.45	59.69	53.09	53.06
	Random	84.69	56.03	58.65	65.48	56.06	54.22	69.33	67.79	71.29	54.51	53.02
	Hedge	83.99	**58.99**	**69.98**	**71.30**	**62.24**	**62.22**	**74.31**	**74.71**	**73.94**	**61.28**	**56.21**
AT Madry et al. (2018)	-	83.23	46.97	57.51	55.84	47.97	50.99	72.68	77.71	60.93	47.29	38.81
	Random	82.99	47.06	57.77	56.86	47.92	51.23	72.73	77.66	82.94	47.27	40.55
	Hedge	81.03	**48.31**	**63.22**	**63.52**	**52.38**	**57.67**	**74.89**	**77.93**	**80.47**	**51.82**	**42.87**
Standard	-	94.78	00.00	00.00	00.83	00.00	00.00	00.04	00.39	00.04	00.00	00.00
	Random	91.32	00.00	**15.97**	74.57	01.16	01.16	**84.37**	**56.41**	79.15	01.20	00.00
	Hedge	91.50	00.00	15.66	**74.71**	**01.23**	**01.21**	84.29	56.37	**79.43**	**01.25**	00.00

Paper

❖ **Experiment**

Table 1: Robust accuracy (%) before perturbation (-), after random noise (Random), and after Hedge Defense (Hedge) on CIFAR10. Eight robust models are showed here. Evaluations are repeated three times with the mean values presented. Extra models and variances are provided in Appendix B.4.

Model	Method	Nat-Acc.	PGD	C&W	Deep Fool	APGD CE	APGD T	FAB	Square	RayS	Auto Attack	Worst Case
WA+ SiLU Gowal et al. (2020)	-	91.10	69.16	67.48	71.23	68.24	66.17	66.70	71.76	72.03	66.16	66.15
	Random	90.82	69.34	71.35	75.65	69.46	67.45	78.82	77.58	79.23	67.76	66.24
	Hedge	90.62	**71.43**	**78.42**	**79.64**	**73.86**	**73.00**	**82.28**	**83.30**	**82.03**	**72.66**	**68.61**
AWP Wu et al. (2020b)	-	88.25	64.14	61.44	64.81	63.56	60.49	60.97	66.15	66.93	60.50	60.46
	Random	87.91	64.18	65.01	70.41	64.53	61.54	73.37	72.47	74.89	62.00	60.42
	Hedge	86.98	**66.16**	**71.94**	**73.83**	**68.18**	**66.47**	**76.37**	**76.69**	**76.80**	**66.68**	**62.72**
RST Carmon et al. (2019)	-	89.69	63.17	61.74	66.69	62.01	60.14	60.66	66.91	67.61	60.13	60.08
	Random	89.50	63.47	65.90	71.16	63.42	61.49	75.48	74.51	77.02	61.89	60.09
	Hedge	88.64	**66.38**	**73.89**	**75.25**	**68.37**	**68.09**	**78.48**	**79.24**	**78.98**	**67.46**	**63.10**
Pre-Training Hendrycks et al. (2019a)	-	87.11	58.19	57.09	59.77	57.52	55.32	55.68	62.38	63.32	55.30	55.28
	Random	86.93	58.52	61.79	70.4	58.59	56.45	71.99	71.16	74.56	56.79	55.32
	Hedge	87.43	**62.01**	**73.12**	**75.50**	**65.63**	**65.38**	**76.62**	**78.32**	**78.65**	**64.45**	**59.17**
MART Wang et al. (2020)	-	87.50	63.49	59.62	63.14	61.83	56.73	57.39	64.88	65.66	56.74	56.70
	Random	87.08	63.59	63.82	72.26	63.02	58.10	73.04	72.11	74.71	58.75	56.99
	Hedge	86.18	**64.56**	**71.71**	**73.86**	**67.00**	**63.30**	**74.35**	**76.09**	**75.60**	**63.42**	**59.35**
HYDRA Sehwag et al. (2020)	-	88.98	60.95	62.19	64.66	59.86	57.67	58.41	65.02	65.61	57.67	57.63
	Random	88.80	61.42	66.08	69.50	61.25	59.03	74.60	73.26	76.20	59.55	57.85
	Hedge	87.82	**64.19**	**73.05**	**73.43**	**66.15**	**65.56**	**76.94**	**77.58**	**77.36**	**65.07**	**61.13**
TRADES Zhang et al. (2019b)	-	84.92	55.83	54.47	58.15	55.08	53.09	53.56	59.45	59.69	53.09	53.06
	Random	84.69	56.03	58.65	65.48	56.06	54.22	69.33	67.79	71.29	54.51	53.02
	Hedge	83.99	**58.99**	**69.98**	**71.30**	**62.24**	**62.22**	**74.31**	**74.71**	**73.94**	**61.28**	**56.21**
AT Madry et al. (2018)	-	83.23	46.97	57.51	55.84	47.97	50.99	72.68	77.71	60.93	47.29	38.81
	Random	82.99	47.06	57.77	56.86	47.92	51.23	72.73	77.66	82.94	47.27	40.55
	Hedge	81.03	**48.31**	**63.22**	**63.52**	**52.38**	**57.67**	**74.89**	**77.93**	**80.47**	**51.82**	**42.87**
Standard	-	94.78	00.00	00.00	00.83	00.00	00.00	00.04	00.39	00.04	00.00	00.00
	Random	91.32	00.00	**15.97**	74.57	01.16	01.16	**84.37**	**56.41**	79.15	01.20	00.00
	Hedge	91.50	00.00	15.66	**74.71**	**01.23**	**01.21**	84.29	56.37	**79.43**	**01.25**	00.00

Paper

❖ **Experiment**

Table 2: Robust accuracy (%) on CIFAR100. All settings align with CIFAR10 in Table 1.

Model	Method	Nat-Acc.	PGD	C&W	Deep Fool	APGD CE	APGD T	FAB	Square	RayS	Auto Attack	Worst Case
WA+ SiLU Gowal et al. (2020)	-	69.15	40.84	39.46	40.74	40.32	37.33	37.65	42.97	43.15	37.29	37.26
	Random	69.02	40.97	43.61	52.22	41.35	38.75	54.59	51.08	55.09	38.79	37.33
	Hedge	68.67	**43.25**	**55.94**	**57.82**	**46.75**	**46.47**	**59.69**	**59.81**	**59.75**	**45.00**	**39.78**
AWP Wu et al. (2020b)	-	60.38	34.13	31.41	31.33	33.37	29.18	29.47	34.57	34.79	29.16	29.15
	Random	60.45	34.21	35.24	44.42	34.53	30.56	45.59	42.94	47.82	30.75	29.25
	Hedge	59.37	**36.69**	**46.01**	**48.13**	**40.18**	**38.10**	**49.49**	**49.82**	**50.34**	**37.75**	**32.29**
TRADES Zhang et al. (2019b)	-	57.34	25.19	32.83	31.97	34.51	37.24	52.74	**54.06**	30.01	33.94	19.61
	Random	55.60	25.30	33.34	33.97	34.11	36.73	51.41	52.36	55.59	33.78	22.00
	Hedge	56.04	**29.75**	**44.09**	**45.33**	**39.25**	**42.04**	**53.27**	54.01	**55.79**	**38.72**	**26.59**
AT Madry et al. (2018)	-	58.61	25.34	35.01	31.93	32.05	35.56	52.62	55.11	32.31	31.70	19.72
	Random	58.49	25.43	35.53	35.54	32.09	35.57	52.58	**55.00**	**58.72**	31.74	21.61
	Hedge	57.66	**28.23**	**43.73**	**45.02**	**35.79**	**40.08**	**53.24**	54.99	57.22	**35.54**	**24.80**

Table 3: Top-1 robust accuracy (%) for Fast Adversarial Training on ImageNet.

Method	Model	Hedge	$\epsilon_a = 2/255$			$\epsilon_a = 4/255$		
			PGD-10	PGD-50	PGD-100	PGD-10	PGD-50	PGD-100
Fast AT Wong et al. (2020)	ResNet-50	-	43.44	43.40	43.38	30.81	30.17	30.13
		✓	**45.38**	**45.44**	**45.43**	**34.42**	**34.63**	**34.71**
	ResNet-101	-	44.69	44.62	44.60	34.02	33.26	33.18
		✓	**47.00**	**47.07**	**47.08**	**38.36**	**38.50**	**38.54**

❖ Experiment

- Ablation Study를 통해 다양한 조건에서 Hedge Defense가 어떻게 작동하는지 실험을 진행함
- Denoising 방법론은 input image에 noise를 추가하는데 시간이 소요되는 것이 단점
- Defense를 위한 noise의 크기는 너무 크면 오히려 역효과가 발생함

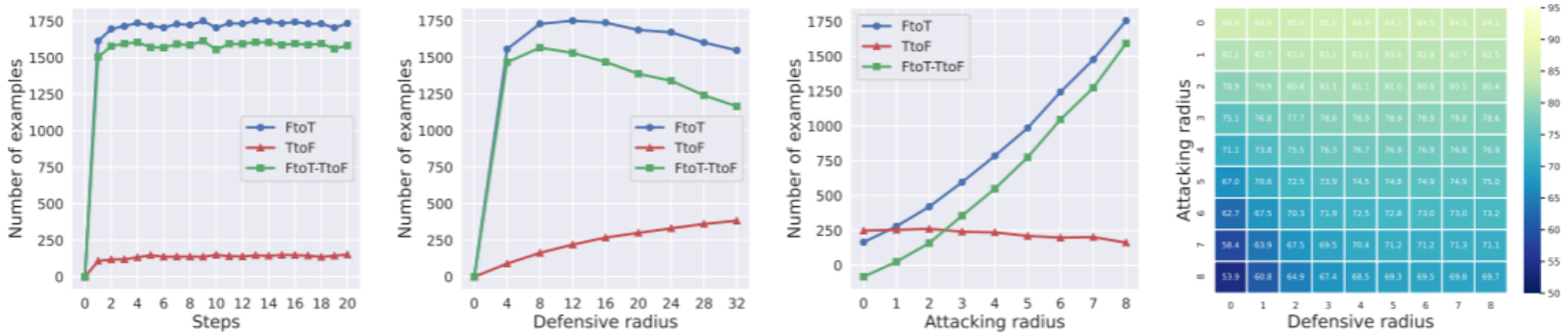
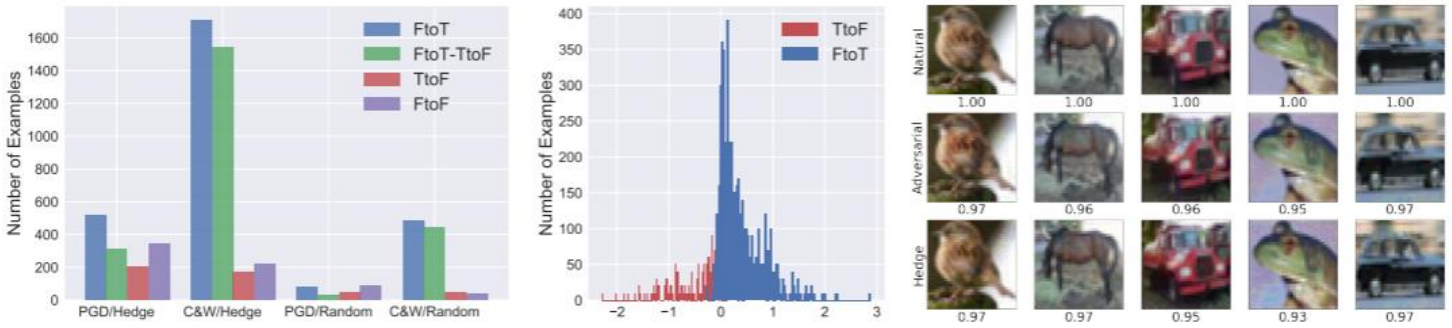


Figure 3: (a) Number of examples for FtoT, TtoF, and FtoT-TtoF against Hedge Defense iteration steps (Section 4.3.1). (b,c,d) Ablation study on the perturbation radii (/255) of ϵ_a and ϵ_d (Section 4.3.2).

❖ Experiment

- Ablation Study를 통해 다양한 조건에서 Hedge Defense가 어떻게 작동하는지 실험을 진행함
- Loss surface의 smooth 정도를 나타내는 Lipschitzness 값을 계산함
- 그 결과, F-To-T의 경우가 T-To-F에 비해 Lipschitzness Difference가 높았음



(a) The Trade-off of Defensive Perturbation (b) Lipschitzness Difference (c) Visualization of Examples

Figure 2: Analytical Experiments: (a) The number of examples for FtoT, TtoF, FtoT-TtoF, and FtoF (Section 4.2.1). (b) The distribution of Lipschitzness difference for FtoT and TtoF (Section 4.2.2). (c) The visualization for natural, adversarial, and hedge examples with their SSIM scores (Section 4.2.3).

❖ Experiment

- 모델 자체가 강건해지는 것이 아닌 Denosing을 통한 방법론은 공격자가 어떤 Denosing 방법을 쓰는지 알고 있는 경우를 가정할 수 있음
- Hedge Defense는 Defense-aware attack에도 큰 영향을 받지 않음

Algorithm 6 Attack Hedge Defense

Input: the coming input $\mathbf{x}$, the number of defensive iterations K, the number of attacking iterations T, the step size η, the deep network $f(\cdot)$, the attacking radius ϵ_{a}, and the defensive radius ϵ_{d}.
// $\mathcal{U}(-\mathbf{1},\mathbf{1})$ generates a uniform noise
Initialization: $\mathbf{x}'_0 \leftarrow \mathbf{x} + \epsilon_{\mathrm{a}}\,\mathcal{U}(-\mathbf{1},\mathbf{1})$.
for $t = 1 \dots T$ **do**
  $\mathbf{x}''_{(t-1),0} \leftarrow \mathbf{x}'_{t-1} + \epsilon_{\mathrm{d}}\,\mathcal{U}(-\mathbf{1},\mathbf{1})$
  for $k = 1 \dots K$ **do**
    $\mathbf{x}''_{(t-1),k} \leftarrow \mathbf{x}''_{(t-1),(k-1)} +$
      $\eta \cdot \mathrm{sign}(\nabla_{\mathbf{x}''_{(t-1),(k-1)}} \sum_{c=1}^{C} \mathcal{L}(f(\mathbf{x}''_{(t-1),(k-1)}), c))$;
    // Π is the projection operator.
    $\mathbf{x}''_{(t-1),k} \leftarrow \Pi_{\mathbb{B}(\mathbf{x}',\epsilon_{\mathrm{d}})}(\mathbf{x}''_{(t-1),k})$;
  end for
  $\mathbf{x}'_t \leftarrow \mathbf{x}'_{t-1} + \eta \cdot \mathrm{sign}(\nabla_{\mathbf{x}''_{(t-1),K}} \mathcal{L}(f(\mathbf{x}''_{(t-1),K}), y))$;
  $\mathbf{x}'_t \leftarrow \Pi_{\mathbb{B}(\mathbf{x},\epsilon_{\mathrm{a}})}(\mathbf{x}'_t)$;
end for
Output: the adversarial example $\mathbf{x}'_T$.

❖ Experiment

- 모델 자체가 강건해지는 것이 아닌 Denosing을 통한 방법론은 공격자가 어떤 Denosing 방법을 쓰는지 알고 있는 경우를 가정할 수 있음
- Hedge Defense는 Defense-aware attack에도 큰 영향을 받지 않음
- Robust accuracy에서 약 0.6% 정도의 근소한 차이를 보임

Table 8: Defense-aware attack on Hedge Defense.

Model	Attack	Direct Prediction (%)	Prediction of Hedge Defense (%)
RST Carmon et al. (2019)	Attack the Model	63.17	66.38
	Attack Hedge Defense	67.08	65.78

Thank You