Adversarial Neural Pruning with Latent Vulnerability Suppression

Divyam Madaan[1], Jinwoo Shin[2,3] and Sung Ju Hwang[1,3,4]

[1]*School of Computing, KAIST, Daejeon, South Korea*

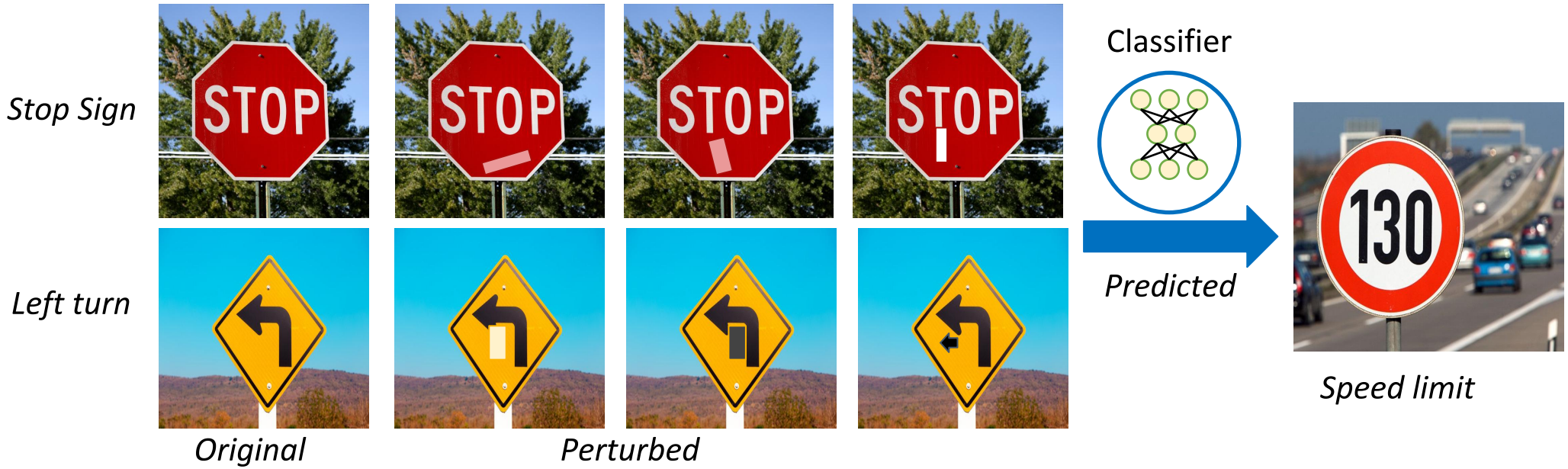
[2]*School of Electrical Engineering, KAIST, Daejeon, South Korea*

[3]*Graduate School of AI, KAIST, Daejeon, South Korea*

[4]*AITRICS, Seoul, South Korea*

Motivation

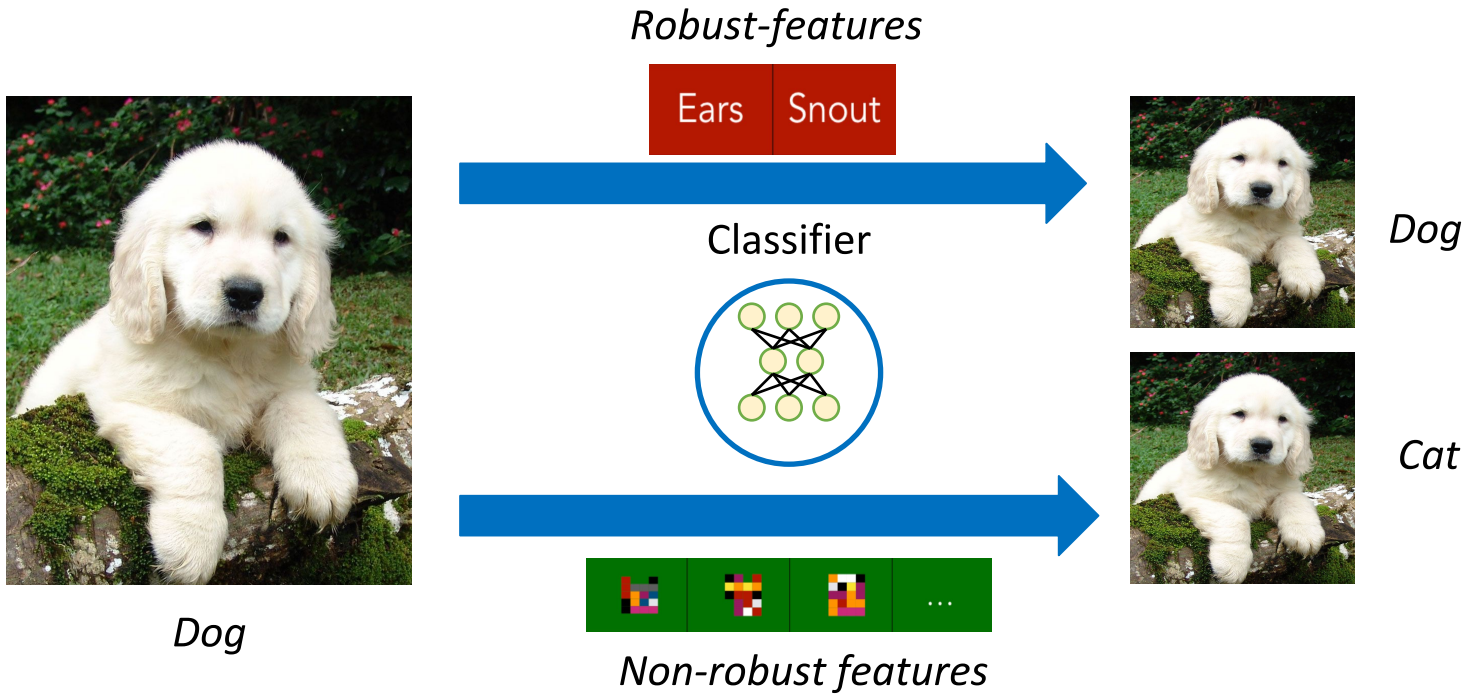
Deep neural networks are extremely brittle to ***adversarial perturbed inputs.***



Robustness and accuracy of these networks is critical for their deployment in ***safety and reliability critical applications***.

Motivation

There exists a set of ***robust*** and ***non-robust*** features in the input space [Ilyas et al., 2019].

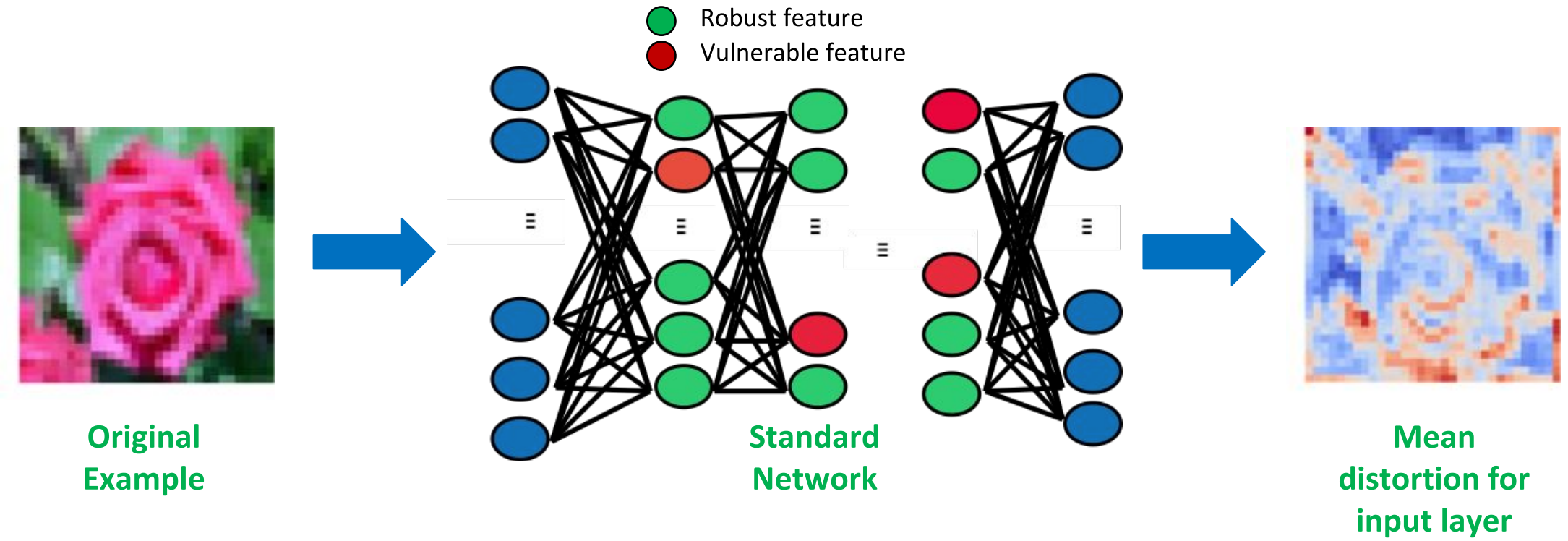


Deep neural networks rely on ***non-robust features*** for generalization to test set.
In this work, we investigate the vulnerability of the ***latent-features of a network***.

[Ilyas et al., 2019] Adversarial Examples are not Bugs, They are Features. NeurIPS 2019

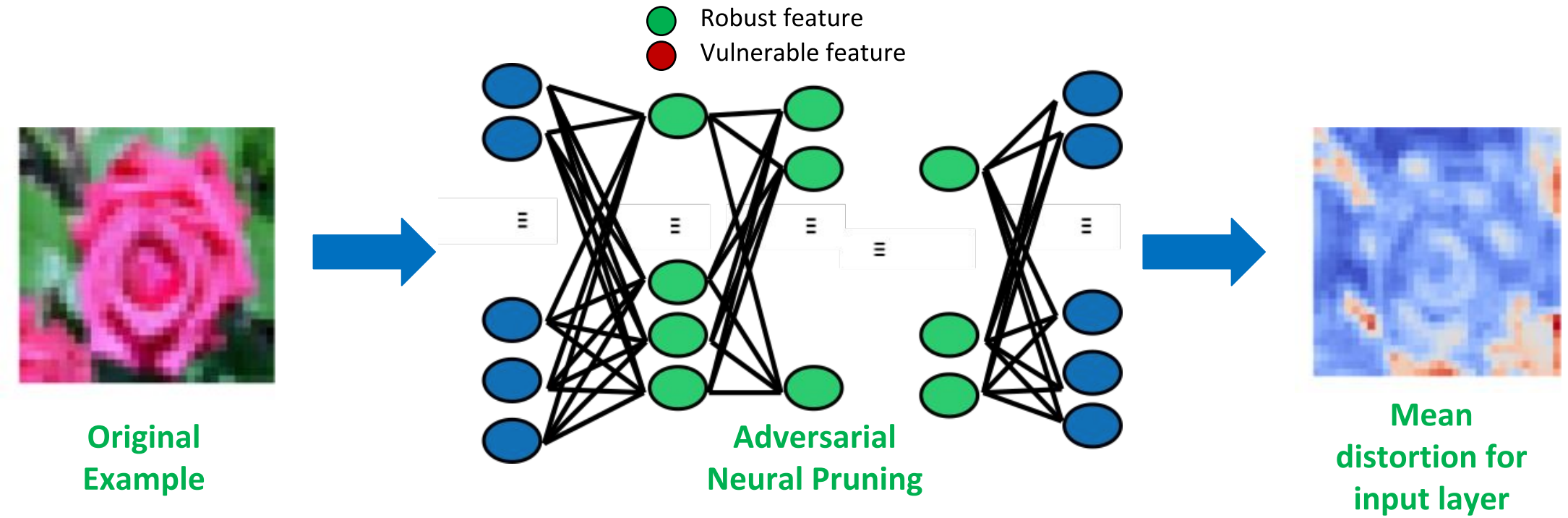
Adversarial Neural Pruning

We found that ***pruning the vulnerable features*** in a model improves adversarial robustness as well as computational efficiency.



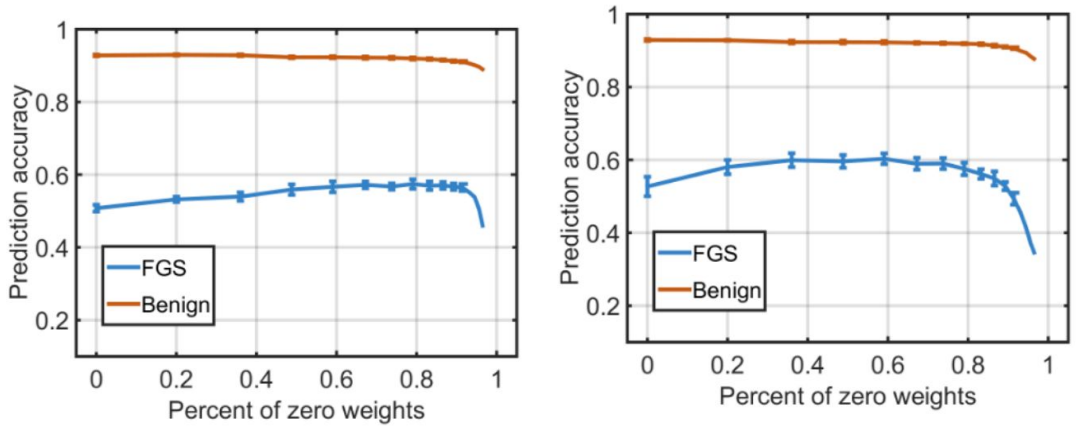
Adversarial Neural Pruning

We found that ***pruning the vulnerable features*** in a model improves adversarial robustness as well as computational efficiency.



Related Work: Robustness and Sparsity

Guo et al. (2019) and Ye et al. (2018) demonstrated that ***higher sparsity leads to more robust model***.
In contrast, Wang et al. (2018) illustrated that ***higher sparsity decreases robustness.***



Robustness of VGG-like network (left) and ResNet-32 (right) with varying weight sparsity [Guo et al., 2019]

Parameters pruned	Natural images	FGSM $\epsilon = 0.1$	FGSM $\epsilon = 0.2$	FGSM $\epsilon = 0.3$	PGD	Papernot's black-box	Trade-off
FGSM Training							
0%	99.2%	97.9%	94.0%	84.7%	0.5%	89.2%	-
weight - 96%	99.0%	94.8%	83.5%	59.0%	2.2%	79.6%	high compression
weight - 80%	99.2%	98.2%	94.7%	85.9%	0.2%	89.6%	high robustness
filter - 70%	98.9%	94.1%	82.3%	60.1%	1.7%	82.5%	high compression
filter - 60%	99.0%	97.8%	93.6%	83.0%	0.4%	85.7%	high robustness
PGD Training							
0%	99.0%	97.3%	95.6%	93.5%	92.5%	96.8%	-
weight - 94%	98.8%	95.6%	94.2%	91.9%	90.6%	95.6%	high compression
weight - 85%	99.0%	96.9%	95.3%	93.3%	92.0%	96.0%	high robustness
filter - 65%	98.9%	89.8%	86.9%	82.3%	75.4%	87.5%	high compression
filter - 40%	99.0%	94.9%	93.1%	90.8%	87.3%	94.1%	high robustness

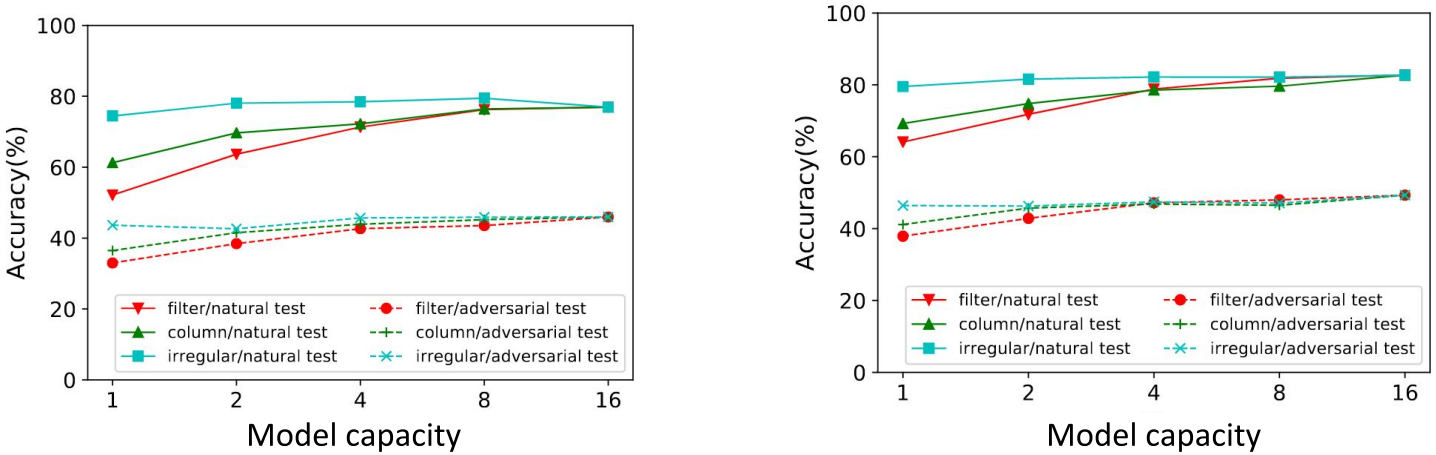
Robustness evaluation of pruned networks by weight or filter pruning on MNIST dataset [Wang et al., 2018]

However, all these works test their hypothesis ***with heuristic pruning techniques***.

[Ye et al., 2018] Defending DNN adversarial attacks with pruning and logits augmentation. ICLR Workshop Submission, 2018
[Wang et al., 2018] Adversarial Robustness of Pruned Neural Networks. ICLR Workshop Submission, 2018.
[Guo et al., 2019] Sparse DNNs with Improved Adversarial Robustness. Neurips 2018

Related Work: Robustness and Sparsity

Ye et al. (2019) proposed ***concurrent adversarial training and weight pruning*** to achieve robust and sparse networks.



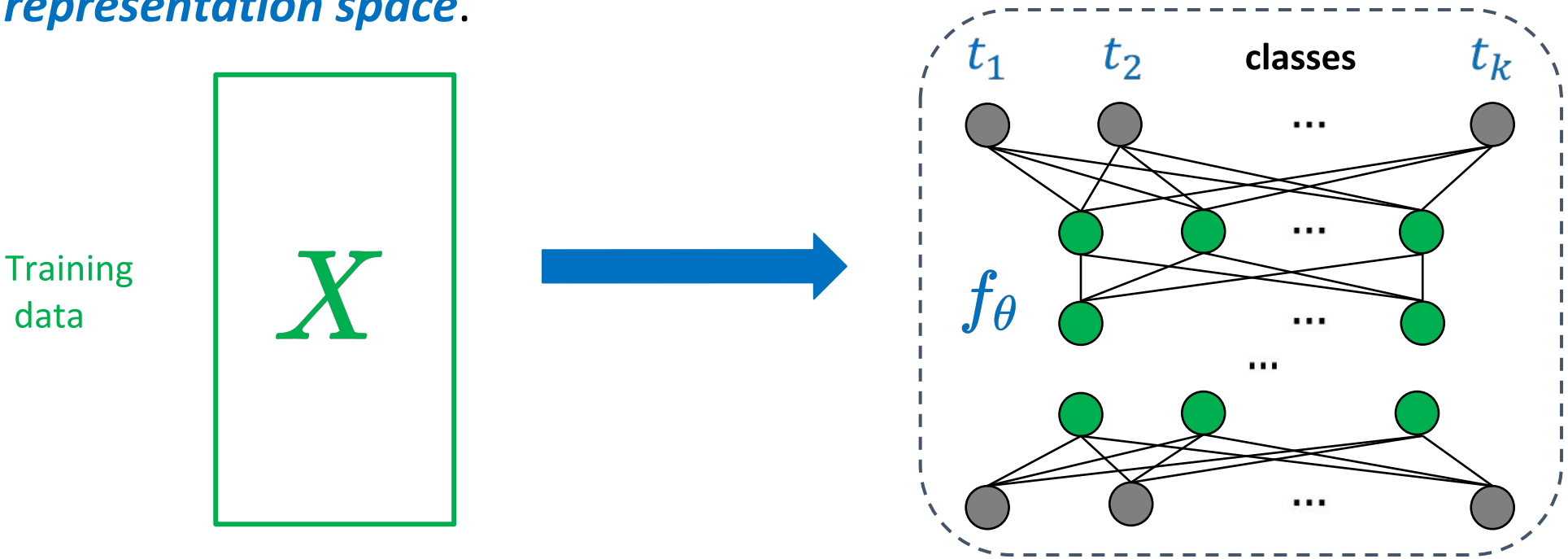
ADMM Robustness with VGG-16 (left) and ResNet-32 (right) with varying weight sparsity on CIFAR-10 [Ye et al., 2019]

However, it requires a ***pre-trained adversarial defense model*** and still does not take into account the ***robustness of a latent-feature***.

[Ye et al., 2019] Adversarial Robustness vs. Model Compression, or Both? ICCV 2019

Vulnerability of a latent-feature

Inspired by the motivation, we introduce the concept of ***vulnerability in the deep latent representation space***.

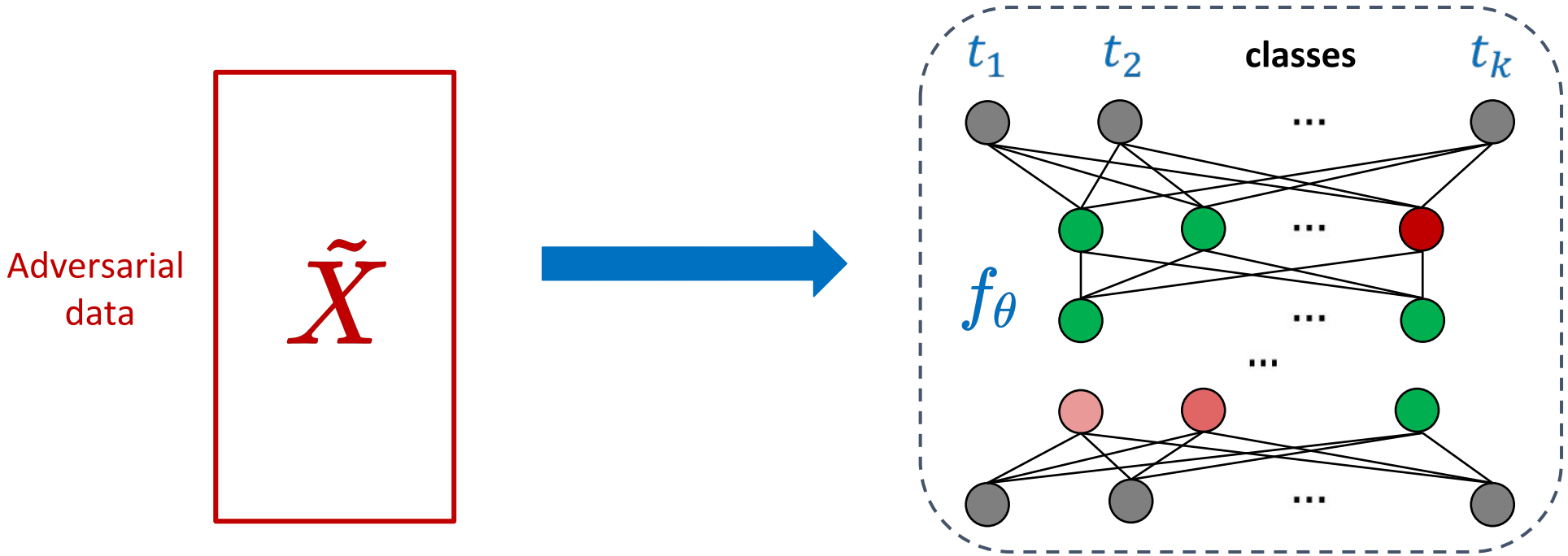


Feature Representation: $z_{l+1} = f_l(z_l) = \max\{W_l z_l + b_l, 0\}, \quad \forall l \in \{1, 2, \ldots, L-1\},$

where, $\theta = \{W_1, \ldots, W_{L-1}, b_1, \ldots, b_{L-1}\}$

Vulnerability of a latent-feature

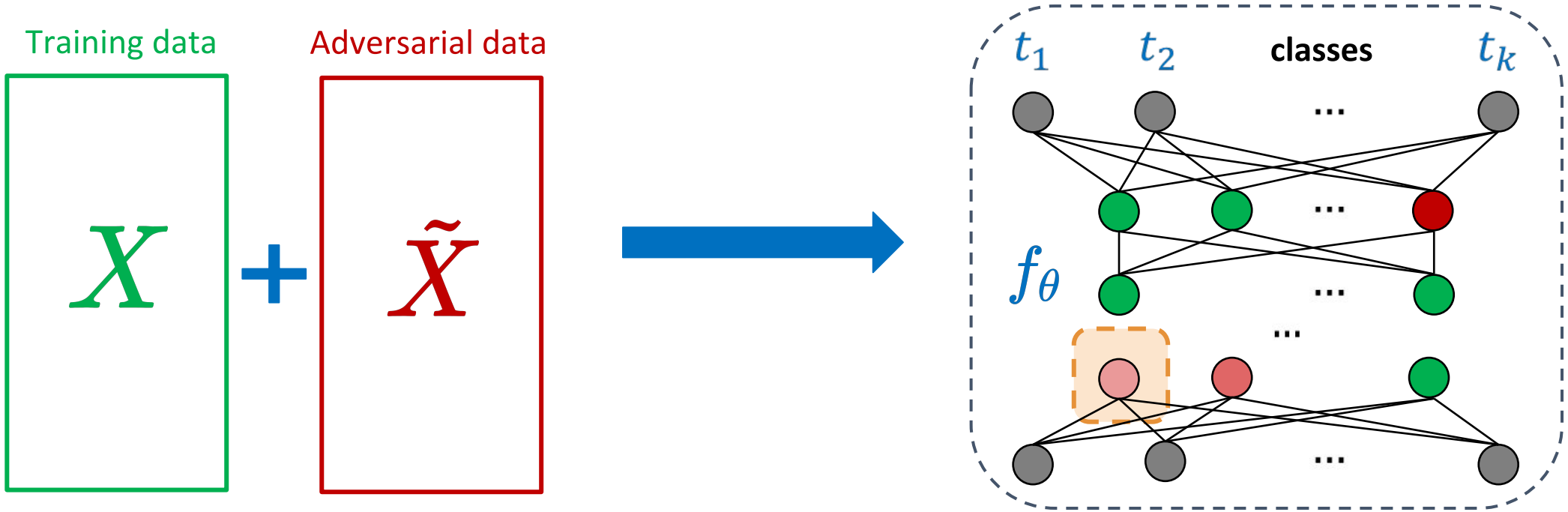
Deep neural networks rely on ***vulnerable features*** for generalization to test set.



Adversarial examples ***distort*** the ***vulnerable features*** to cause ***misclassification***.

Vulnerability of a latent-feature

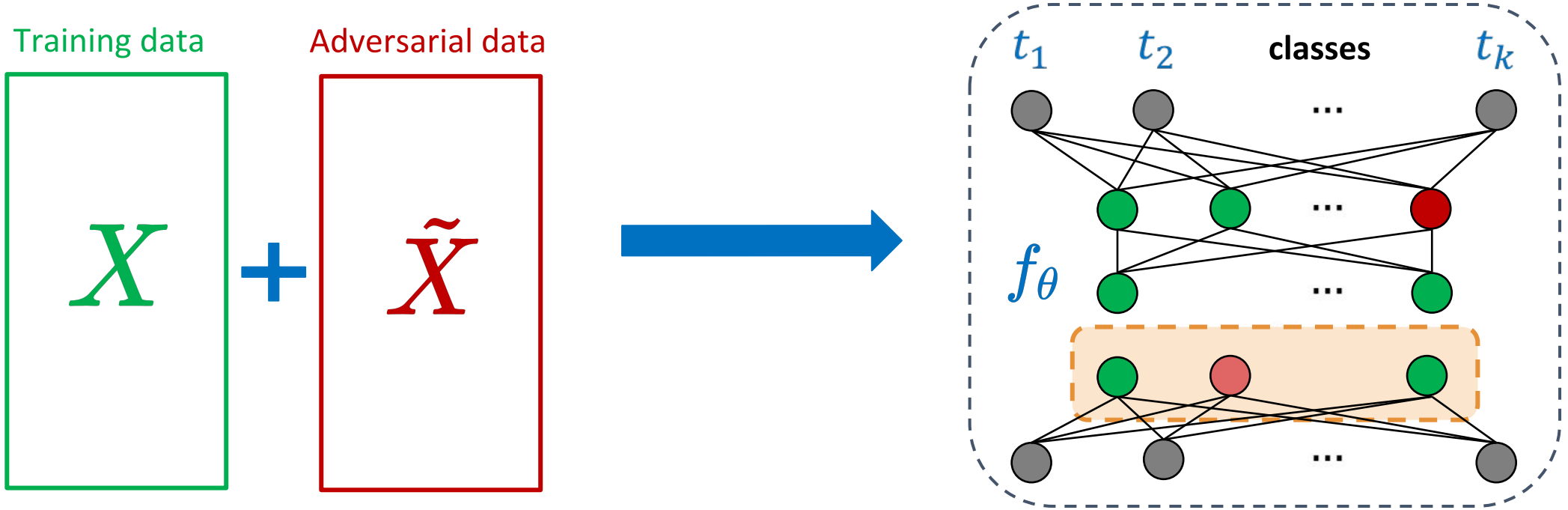
Vulnerability of a feature is the ***difference*** between the ***clean*** and ***adversarial*** feature.



Vulnerability of a feature: $v(z_{lk}, \tilde{z}_{lk}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} | z_{lk} - \tilde{z}_{lk} |$

Vulnerability of a layer

Vulnerability of a layer is the ***sum of vulnerabilities of all latent-features in that layer.***

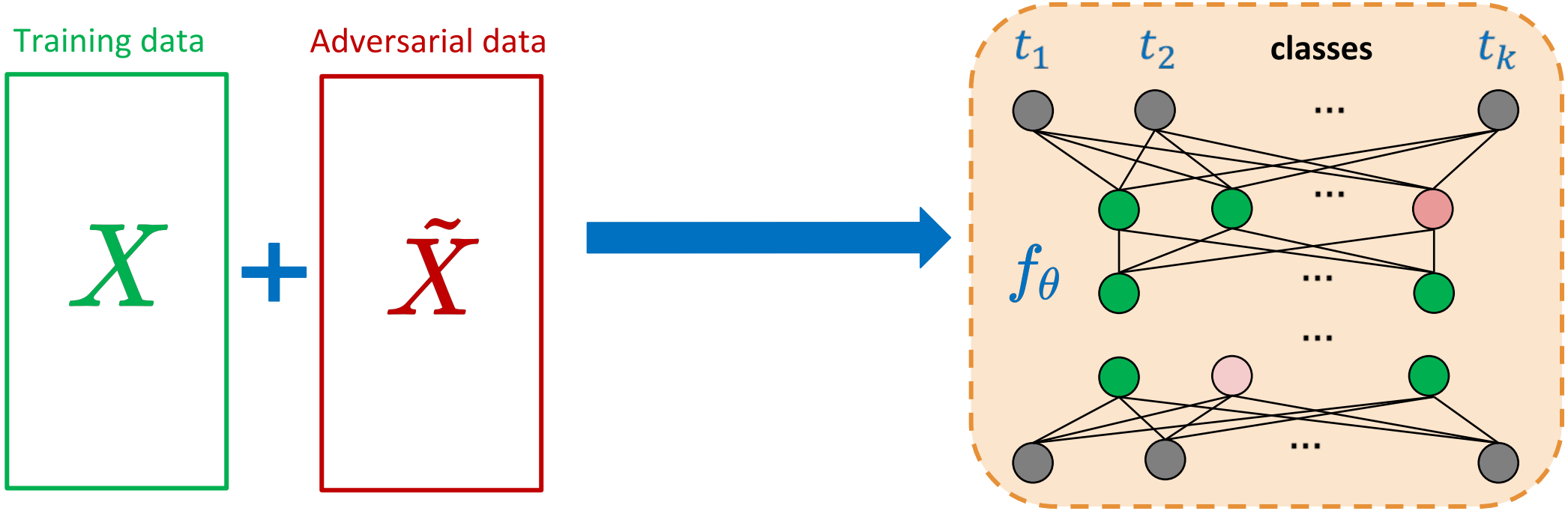


Vulnerability of a layer: $\overline{v_l} = \frac{1}{N_l} \sum_{k=1}^{k=N_l} v(z_{lk}, \tilde{z}_{lk})$

# of features in a layer

Vulnerability Suppression (VS)

Vulnerability suppression loss (VS loss) ***minimizes*** the ***vulnerability of the network***.

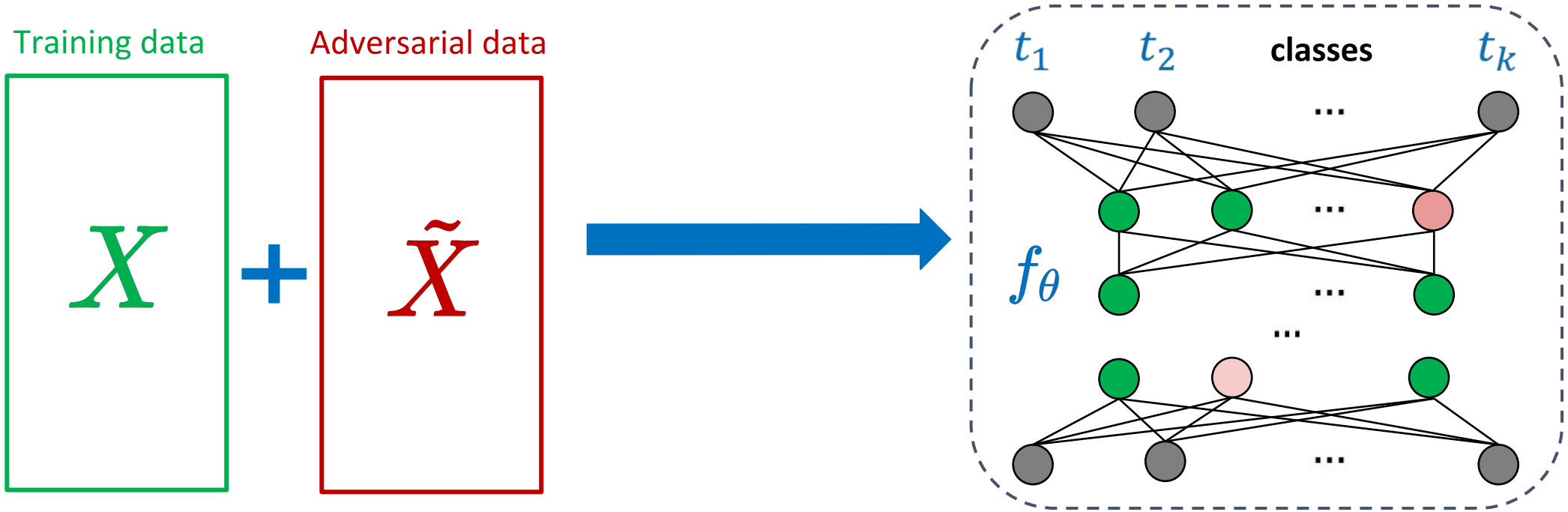


Vulnerability of a network: $V(f_\theta(X), f_\theta(\tilde{X})) = \frac{1}{L-1}\sum_{l=1}^{l=L-1} \overline{v_l}$

# of layers

Vulnerability Suppression (VS)

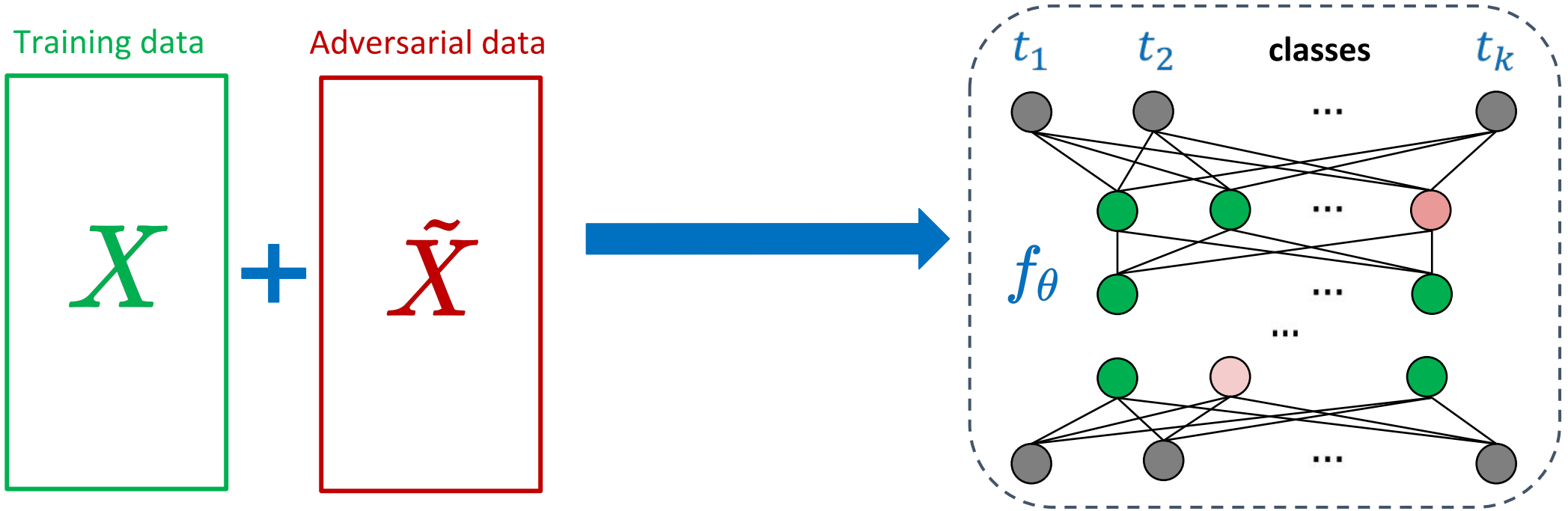
VS loss ***minimizes*** the overall ***vulnerability of the network***.



Objective: $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left\{ \underbrace{J(\theta, x, y)}_{\text{classification loss}} + \underbrace{\lambda \cdot V(f_\theta(x), f_\theta(\tilde{x}))}_{\text{vulnerability suppression loss}} \right\}$

Adversarial Neural Pruning (ANP)

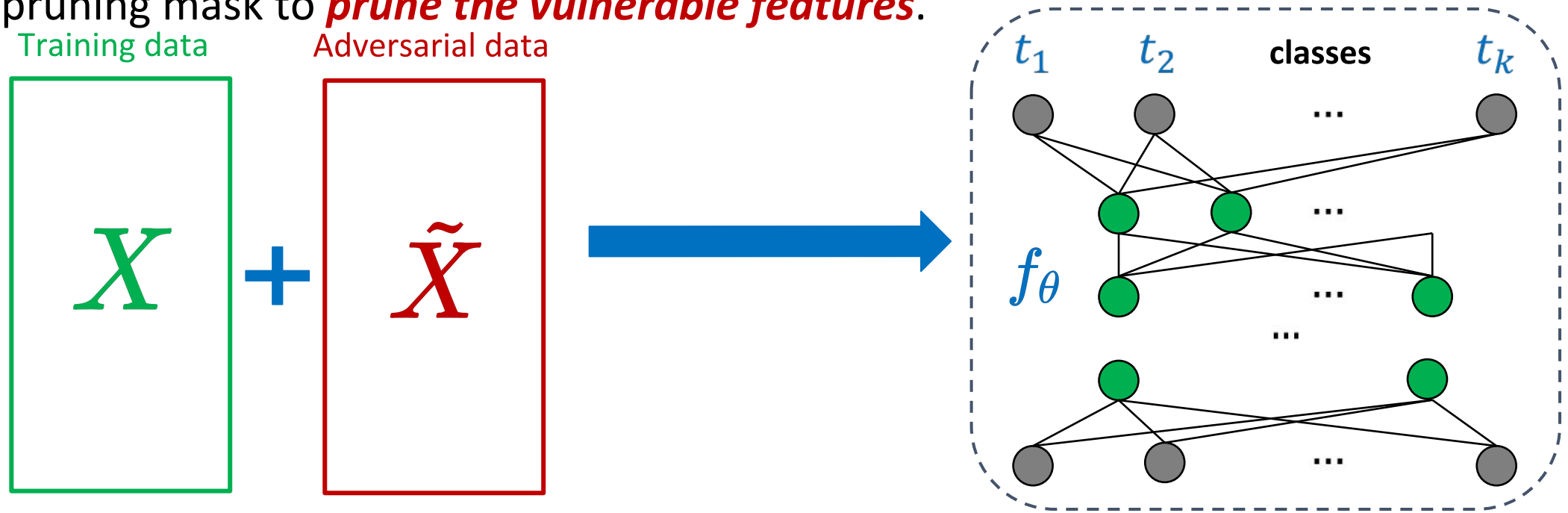
ANP uses pruning as a defense mechanism and sets the ***vulnerable-features to zero***.



ANP learns to prune the vulnerable features in a Bayesian framework to obtain a ***robust*** and ***sparse*** model.

Adversarial Neural Pruning with Vulnerability Suppression

ANP-VS suppresses the vulnerability of latent-features and learns a Bayesian pruning mask to ***prune the vulnerable features***.



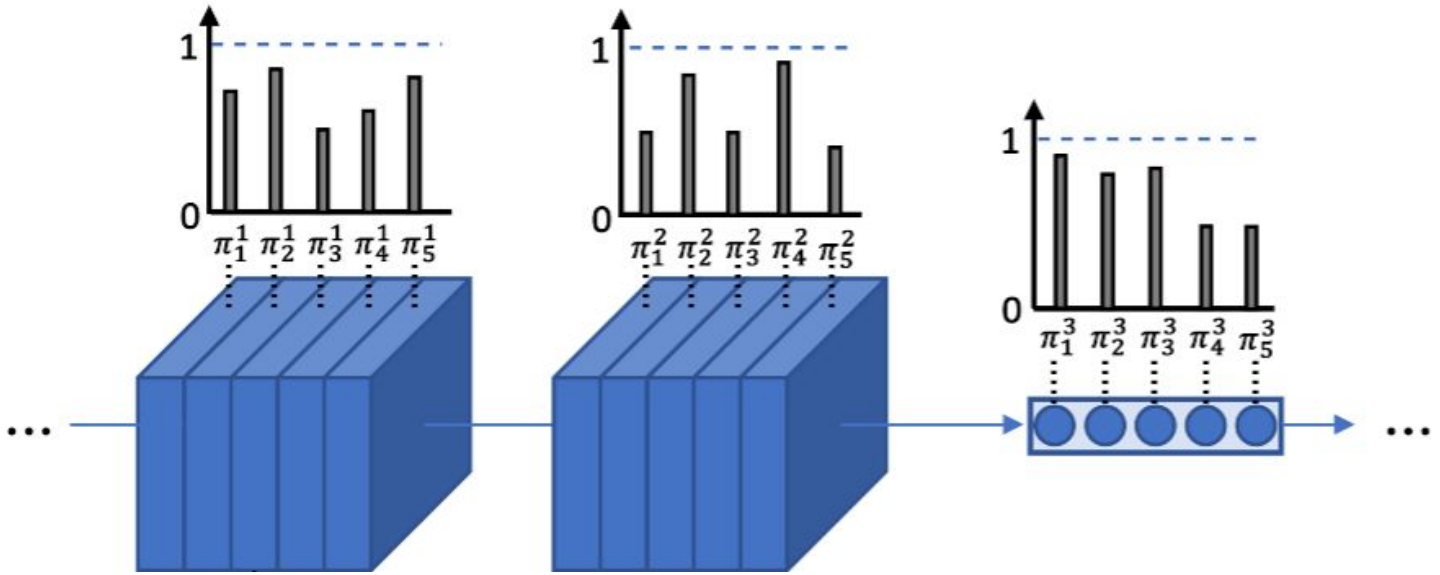
Objective:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big\{ \underbrace{J(\theta \odot M, x, y)}_{\text{classification loss}} + \underbrace{\lambda \cdot V(f_\theta(x), f_\theta(\tilde{x}))}_{\text{vulnerability suppression loss}}$$

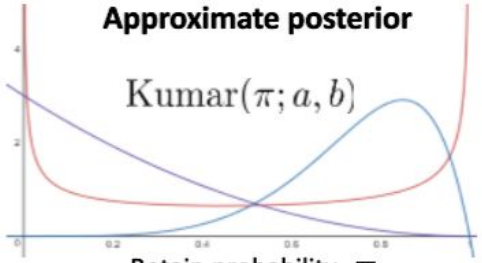
$$\min_{M} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left(\mathcal{L}\left(\theta \odot M, \tilde{x}, y\right)\right)$$

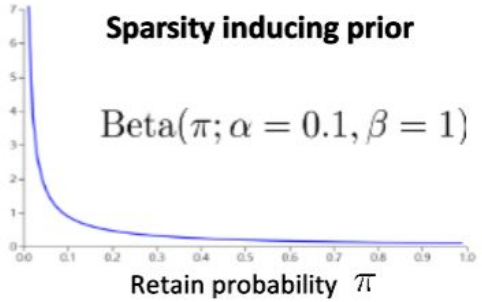
Adversarial Beta-Bernoulli Dropout

ANP with Beta-Bernoulli Dropout [Lee et al., 2018] models the dropout probability for each channel/neuron with the ***sparsity inducing Beta-Bernoulli distribution***.



Convolution layers with *n* channels



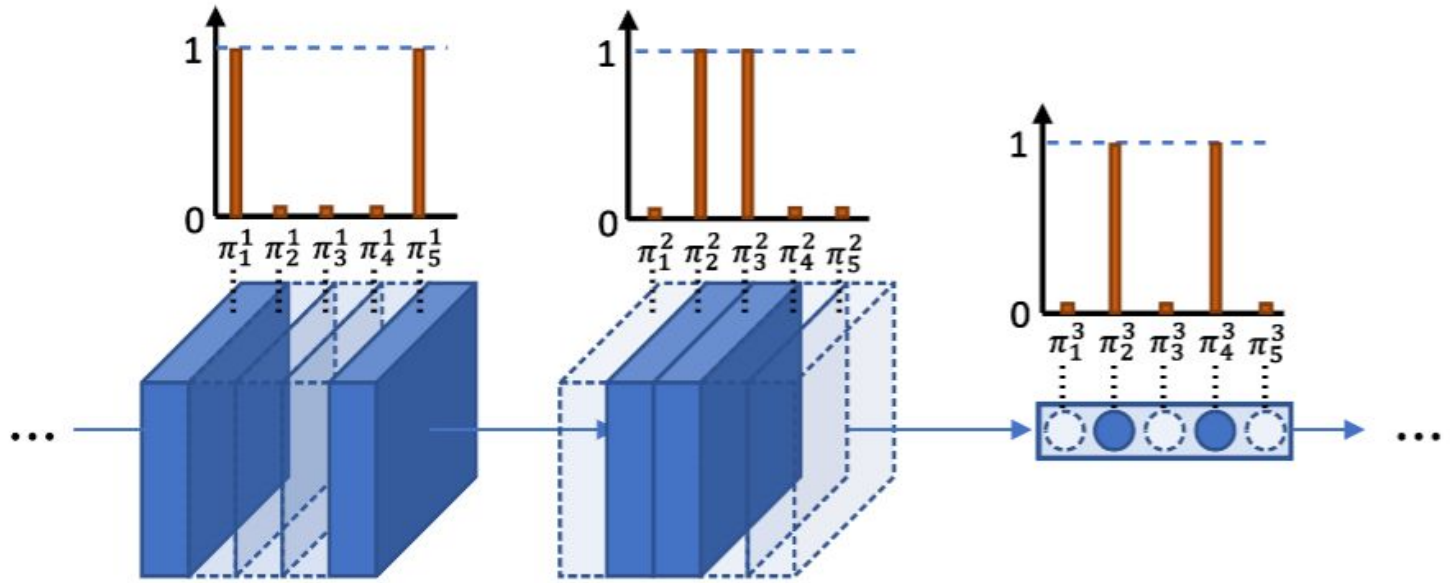


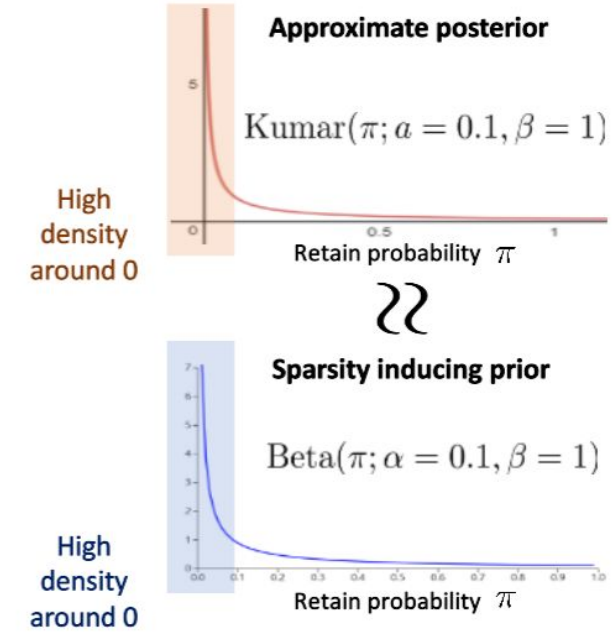
The ***activated channels/neurons*** are modelled according to the ***Bernoulli distribution***.

[Lee et al., 2018] Adaptive Network Sparsification via Dependent Variational Beta-Bernoulli Dropout. *arXiv preprint arXiv:1805.10896*

Adversarial Beta-Bernoulli Dropout

ANP with Beta-Bernoulli Dropout [Lee et al., 2018] generates dropout mask from ***sparsity inducing Beta-Bernoulli prior***.





Objective: $\min_M \left\{ \sum_{n=1}^{N} \mathbb{E}_q \left[\log p\left(y_n | f(\tilde{x}_n; \theta \odot M)\right] - D_{\mathrm{KL}} \left[q\left(M; \pi\right) || p(M|\pi) \right] \right\}$

ANP is general, and can be applied to any ***Bayesian pruning technique***.

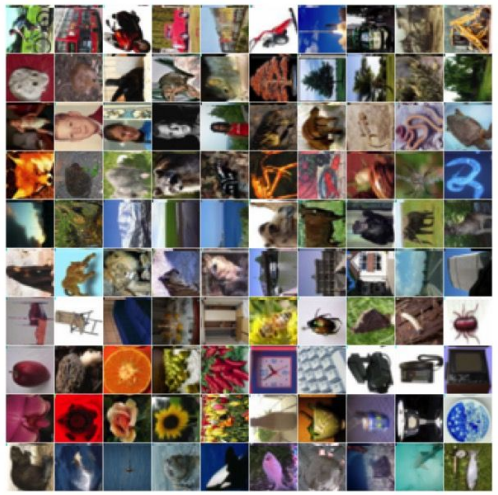
[Lee et al., 2018] Adaptive Network Sparsification via Dependent Variational Beta-Bernoulli Dropout. *arXiv preprint arXiv:1805.10896*

Dataset

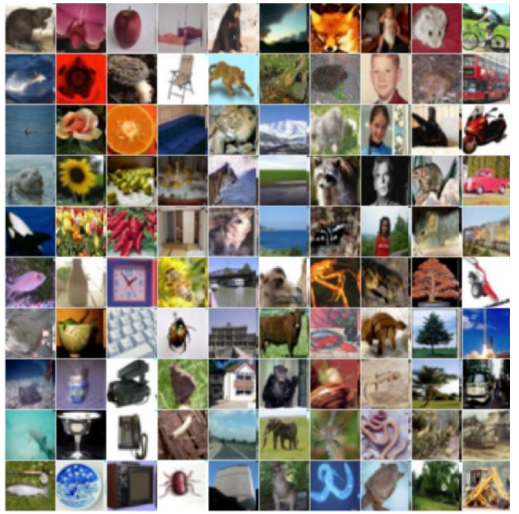
We evaluate our model and baselines on ***three benchmark datasets***.

MNIST [Lecun, 1998]
A dataset with 60,000 gray scale images of ***handwritten digits with ten classes***.

CIFAR10 [Krizhevsky, 2012]
A dataset with 60,000 images from ***ten animal and vehicle classes***.



CIFAR100 [Krizhevsky, 2012]
A dataset with 60,000 images from ***100 generic object classes***.



[Lecun, 1998] Lecun, Y. The MNIST database of handwritten digits.
[Krizhevsky, 2012] Krizhevsky, A. Learning multiple layer of features from tiny images. University of Toronto 2012

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

Standard: Base convolutional network.

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

Bayesian Pruning (BP) [Lee et al., 2018]: Base network with Beta-bernoulli dropout.

[Lee et al., 2018] Adaptive Network Sparsification via Dependent Variational Beta-Bernoulli Dropout. *arXiv preprint arXiv:1805.10896*

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
AT	87.50	49.85	63.70	0.050	0.047	100.0	1.00	0.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

Adversarial Training (AT) [Kurakin et al., 2016, Madry et al., 2016]*:* Adversarial trained network.

[Kurakin et al., 2016] Adversarial Machine Learning at Scale. ICLR 2016
[Madry et al., 2016] Towards deep learning models resistant to adversarial attacks. ICLR 2018

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
AT	87.50	49.85	63.70	0.050	0.047	100.0	1.00	0.00
AT BNN	86.69	51.87	64.92	0.267	0.238	200.0	0.50	0.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

AT BNN [Liu et al., 2019]: Adversarial Bayesian trained neural network.

[Liu et al., 2019] ADV-BNN: Improved Adversarial Defense Through Robust Bayesian Neural Network. ICLR 2019

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
AT	87.50	49.85	63.70	0.050	0.047	100.0	1.00	0.00
AT BNN	86.69	51.87	64.92	0.267	0.238	200.0	0.50	0.00
Pre. AT	87.50	52.25	66.10	0.041	0.036	100.0	1.00	0.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

Pretrained AT (Pre. AT) [Hendrycks et al., 2019]: Adversarial training on a pretrained base model.

[Hendrycks et al., 2019] Using Pre-Training can Improve Model Robustness and Uncertainty. ICML 2019

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
AT	87.50	49.85	63.70	0.050	0.047	100.0	1.00	0.00
AT BNN	86.69	51.87	64.92	0.267	0.238	200.0	0.50	0.00
Pre. AT	87.50	52.25	66.10	0.041	0.036	100.0	1.00	0.00
ADMM	78.15	47.37	62.15	0.034	0.030	100.0	1.00	75.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

ADMM [Ye et al., 2019]: Concurrent weight pruning and adversarial training.

[Ye et al., 2019] Adversarial Robustness vs. Model Compression, or Both? ICCV 2019

Result on CIFAR-10 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	92.76	13.79	41.65	0.077	0.065	100.0	1.00	0.00
BP	92.91	14.30	42.88	0.037	0.033	12.41	2.34	75.92
AT	87.50	49.85	63.70	0.050	0.047	100.0	1.00	0.00
AT BNN	86.69	51.87	64.92	0.267	0.238	200.0	0.50	0.00
Pre. AT	87.50	52.25	66.10	0.041	0.036	100.0	1.00	0.00
ADMM	78.15	47.37	62.15	0.034	0.030	100.0	1.00	75.00
TRADES	80.33	52.08	64.80	0.045	0.042	100.0	1.00	0.00
ANP-VS	**88.18**	**56.21**	**71.44**	**0.019**	**0.016**	**12.27**	**2.41**	**76.53**

TRADES [Zhang et al., 2019]: Explicit trade-off between natural and robust generalization.

[Zhang et al., 2019] Theoretically Principled Trade-off between Robustness and Accuracy. ICML 2019

Result on CIFAR-100 dataset

Our proposed ***ANP-VS*** outperforms all the baselines.

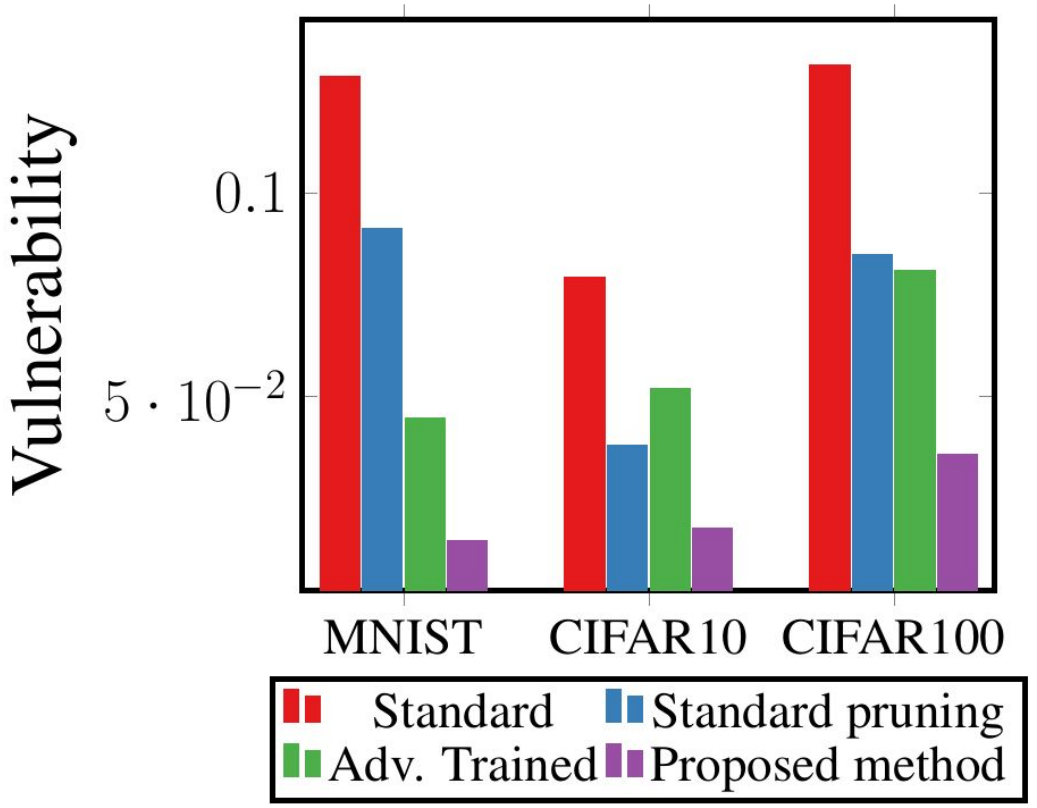
Model	*Clean acc.*	*Adv. (WB)*	*Adv. (BB)*	*Vul. (WB)*	*Vul. (BB)*	*Memory*	*xFLOPS*	*Sparsity*
Standard	67.44	2.81	14.94	0.143	0.119	100.0	1.00	0.00
BP	69.40	3.12	16.39	0.067	0.059	18.59	1.95	63.48
AT	57.79	19.07	32.47	0.079	0.071	100.0	1.00	0.00
AT BNN	53.75	19.40	30.38	0.446	0.385	200.0	0.50	0.00
Pre. AT	57.14	19.86	35.42	0.071	0.065	100.0	1.00	0.00
ADMM	52.52	19.65	31.30	0.060	0.056	100.0	1.00	65.00
TRADES	56.70	21.21	32.81	0.065	0.060	100.0	1.00	0.00
ANP-VS	**59.15**	**22.35**	**37.01**	**0.035**	**0.030**	**16.74**	**2.02**	**68.80**

Results

Both our models outperforms the baselines.

	Model	Clean acc.	Adv. (WB)	Vul. (WB)
CIFAR-10	*AT*	87.50	49.85	0.050
	AT-VS	87.44	51.52	0.024
	ANP	88.36	55.63	0.022
	ANP-VS	**88.18**	**56.21**	**0.016**
CIFAR-100	*AT*	57.79	19.07	0.079
	AT-VS	57.74	20.06	0.061
	ANP	58.47	22.20	0.037
	ANP-VS	**59.15**	**22.35**	**0.035**

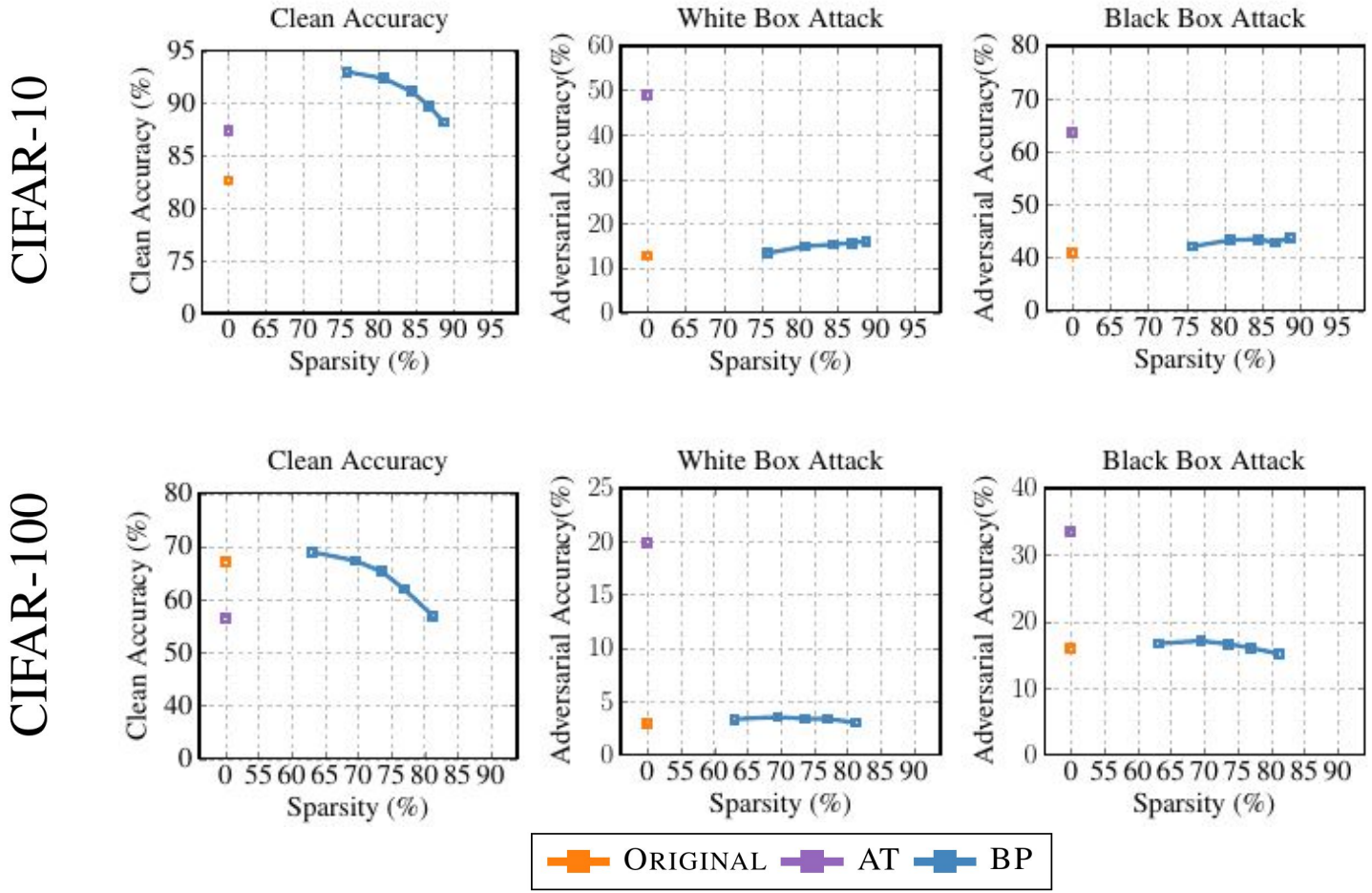
Performance of different components



Mean distortion

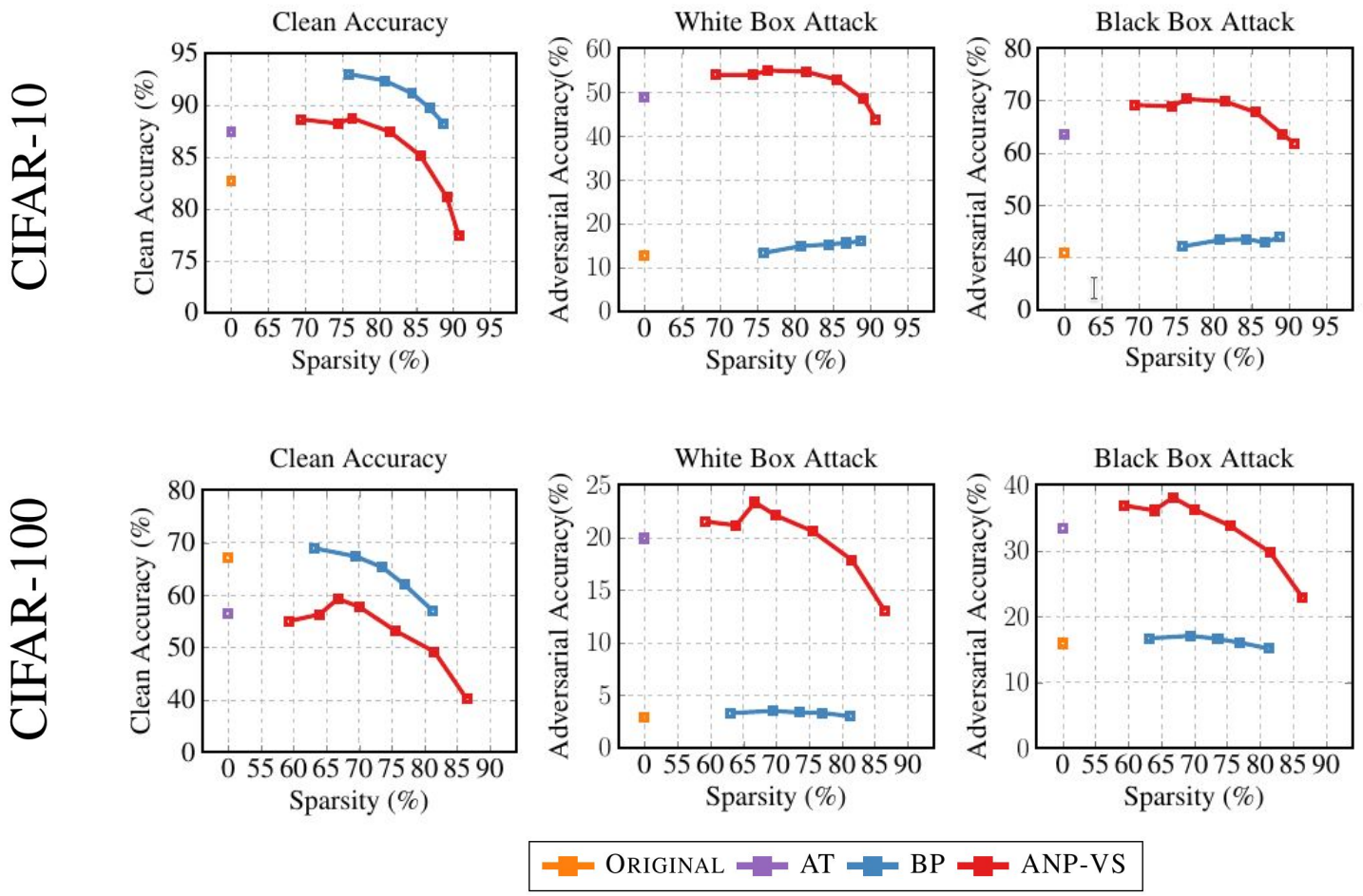
Performance with higher compression

ANP-VS outperforms the baselines even with ***higher sparsity of 80%***.



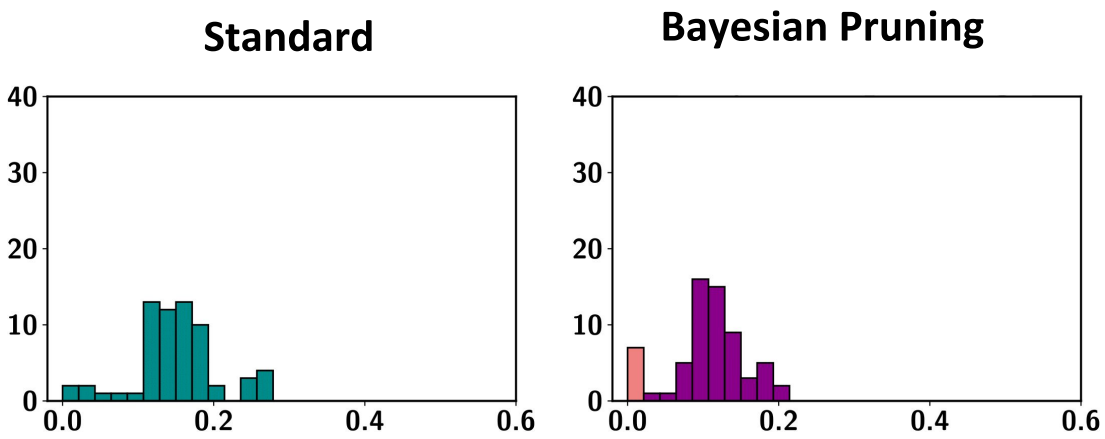
Performance with higher compression

ANP-VS outperforms the baselines even with ***higher sparsity of 80%***.



Vulnerability of input-layer features

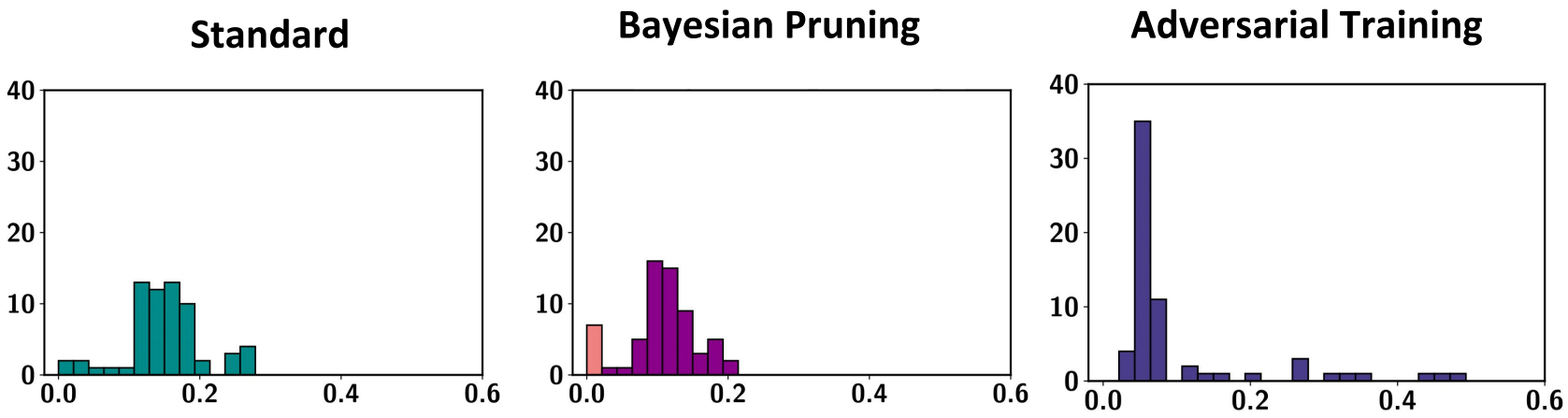
Bayesian pruning ***zeros out some of the distortions*** in tha latent-features.



However, it does not consider the ***distortion of the features*** while pruning.

Vulnerability of input-layer features

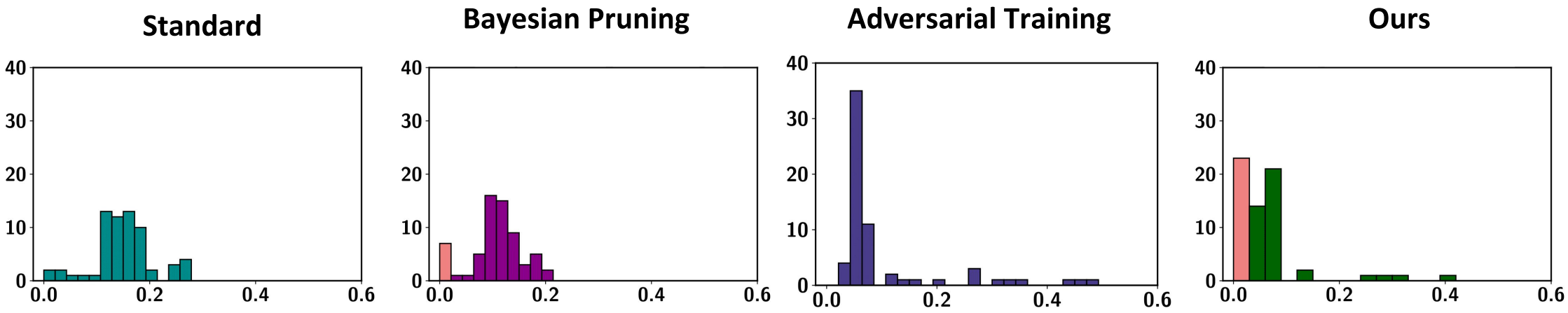
Adversarial training ***reduces the distortion*** level of all features.



However, adversarial training ***does not zero out*** the vulnerable latent-features.

Vulnerability of input-layer features

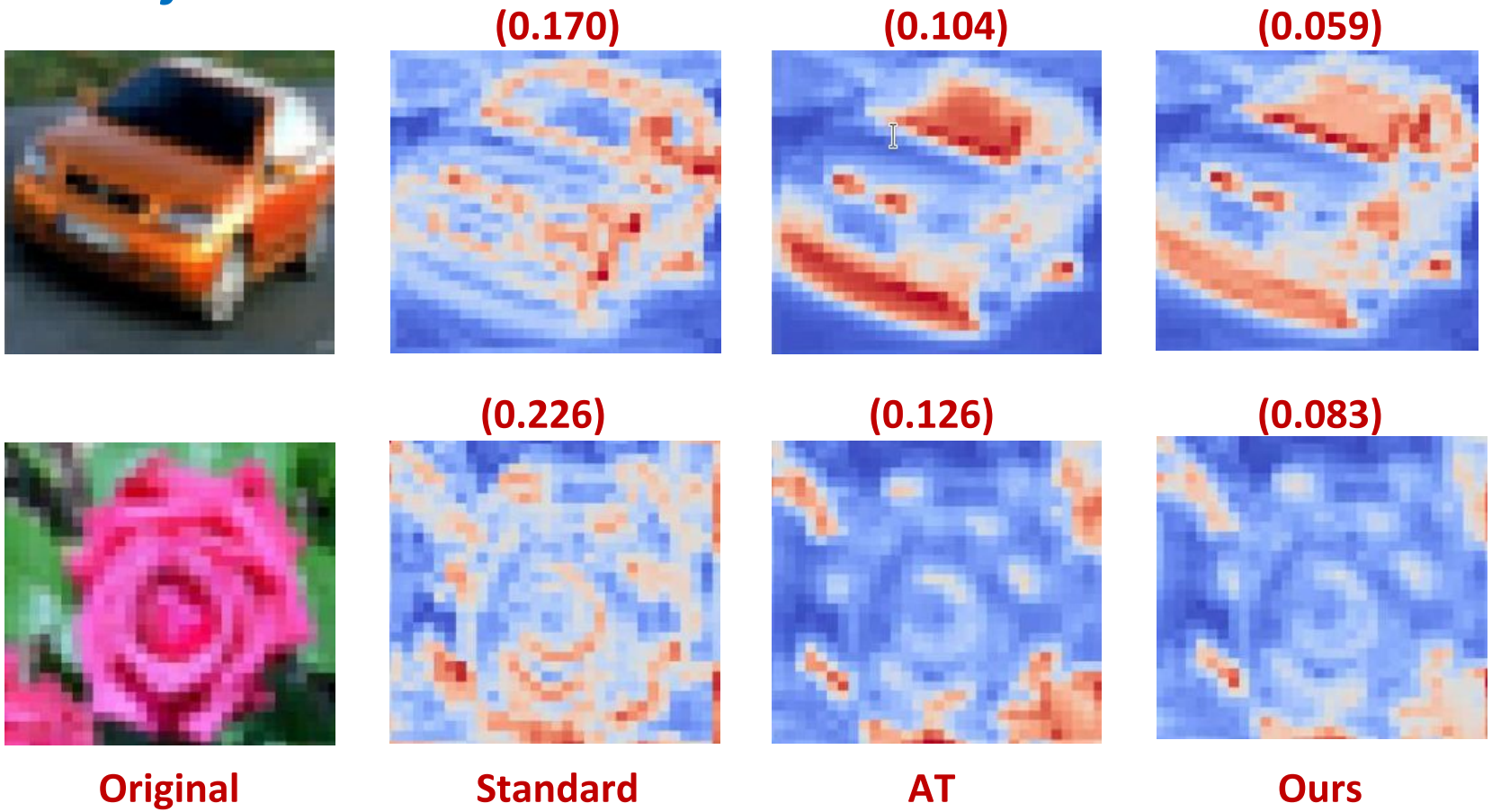
ANP-VS leads to ***reduction in latent-features distortion*** which results in robustness.



ANP-VS has the ***largest number of features with zero distortion***, and low distortion level in general.

Latent-features visualization

Our proposed method leads to significant reduction in the ***vulnerability of latent-features***.

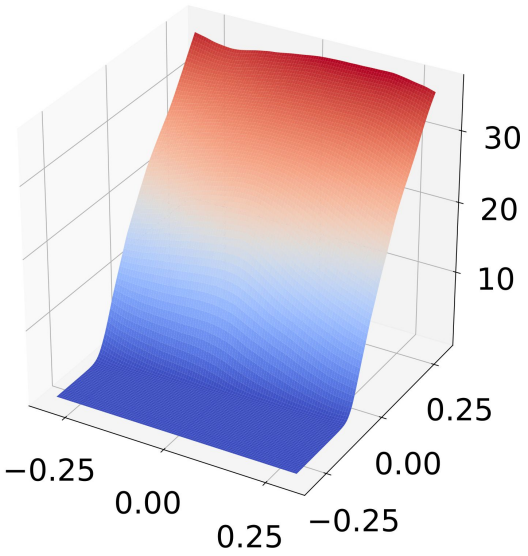


Visualization of the vulnerability of the latent-features with respect to the input pixels for various datasets.

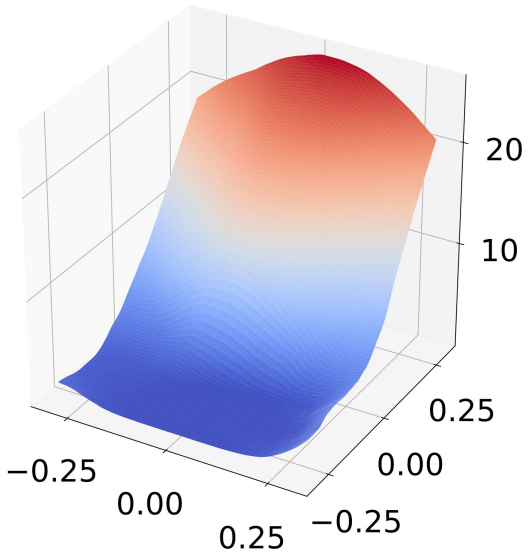
Loss surface visualization

Also, our proposed method achieves ***smoother loss surface***.

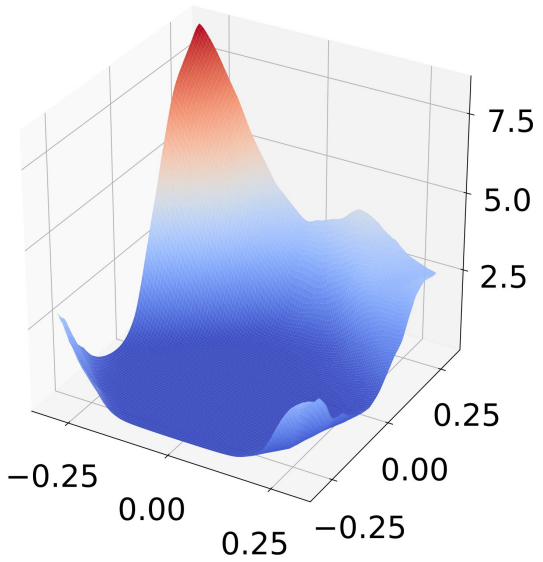
Standard



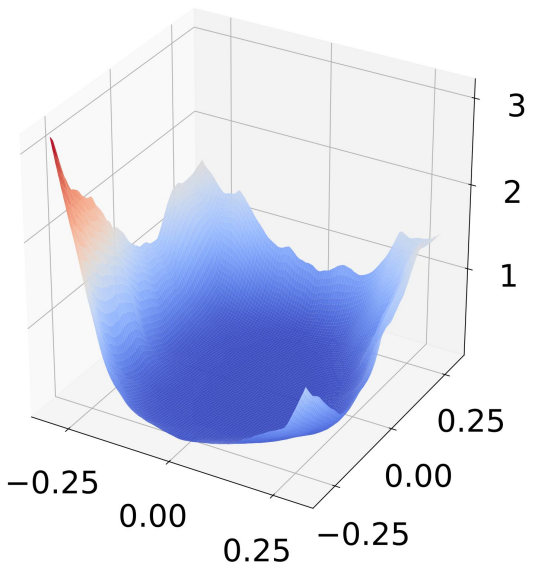
Bayesian Pruning



Adversarial Training



Ours



It indicates the ***absence*** of ***gradient obfuscation***, demonstrating the effectiveness of our method.

Conclusion

- We tackle ***the fundamental cause*** of ***vulnerability of deep networks*** by investigating the distortion of ***latent-features***.
- Adversarial Neural Pruning with Vulnerability Suppression loss (ANP-VS) ***prunes the vulnerable features*** and ***minimizes the feature vulnerability*** in order to ***improve adversarial robustness***.
- Results show that our models ***minimizes the feature vulnerability, improves robustness*** with ***negligible memory and computational requirements*.**
- We believe that our paper can be ***an essential part toward building memory-efficient robust models***.

Codes available at https://github.com/divyam3897/ANP_VS

KAIST

AITRICS

Thank you