Sensitivity Analysis of Deep Neural Networks

Hai Shu[1] and Hongtu Zhu[2]

[1]The University of Texas MD Anderson Cancer Center

[2]AI Labs, Didi Chuxing, Beijing

AAAI-19
January 29, 2019

Background

- State-of-the-art DNNs can be sensitive to small perturbations. This impacts on safety-critical applications, e.g., self-driving cars, face recognition, …

Image by smoothgroover22 under licensed under CC BY-SA 2.0; changes made

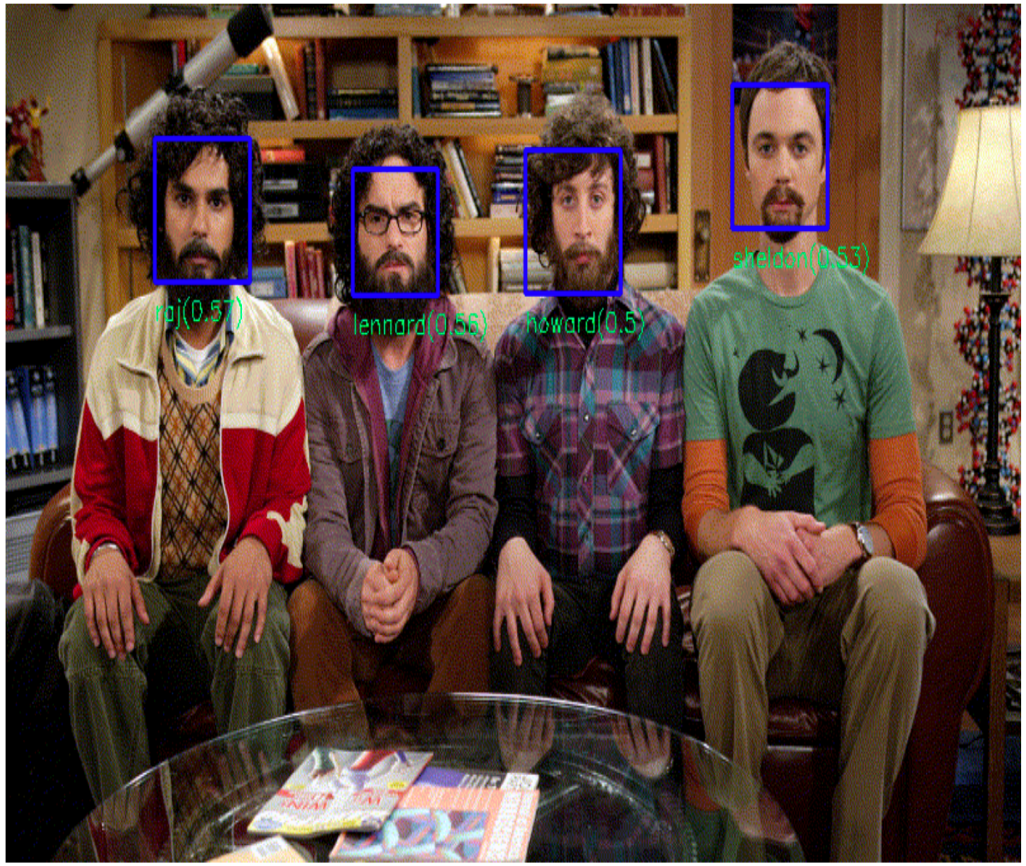


Image by Vincent Mühler under licensed under MIT License; changes made

Background (cont.)

- Commonly used measures for model sensitivity to perturbations, e.g., Jacobian norm (Novak et al. 2018) and Cook's influence (Cook 1986), are variant to scaling reparameterization of the perturbation.

Jacobian norm: $$\|\mathbf{J}(\boldsymbol{\alpha})\|_F := \left[\sum_{i=1}^{p}\left(\frac{\partial f}{\partial \alpha_i}\right)^2\right]^{1/2} = k\|\mathbf{J}(\boldsymbol{\alpha}')\|_F \quad \text{for } \boldsymbol{\alpha}' = k\boldsymbol{\alpha}\,.$$

Background (cont.)

- Commonly used measures for model sensitivity to perturbations, e.g., Jacobian norm (Novak et al. 2018) and Cook's influence (Cook 1986), are variant to scaling reparameterization of the perturbation.

Jacobian norm: $$\|\mathbf{J}(\boldsymbol{\alpha})\|_F := \left[\sum_{i=1}^{p}\left(\frac{\partial f}{\partial \alpha_i}\right)^2\right]^{1/2} = k\|\mathbf{J}(\boldsymbol{\alpha}')\|_F \quad \text{for} \quad \boldsymbol{\alpha}' = k\boldsymbol{\alpha}\,.$$

- The scaling-variant issue causes the misleading results for

Background (cont.)

- Commonly used measures for model sensitivity to perturbations, e.g., Jacobian norm (Novak et al. 2018) and Cook's influence (Cook 1986), are variant to scaling reparameterization of the perturbation.

Jacobian norm: $$\|\mathbf{J}(\boldsymbol{\alpha})\|_F := \left[\sum_{i=1}^{p}\left(\frac{\partial f}{\partial \alpha_i}\right)^2\right]^{1/2} = k\|\mathbf{J}(\boldsymbol{\alpha}')\|_F \quad \text{for} \quad \boldsymbol{\alpha}' = k\boldsymbol{\alpha}\,.$$

- The scaling-variant issue causes the misleading results for
  - Measuring the effect of the simultaneous perturbation to multiple parameters at different scales:

Weights/biases in a convolution layer

&

Shift/scale parameters in a batch normalization layer

Background (cont.)

- Commonly used measures for model sensitivity to perturbations, e.g., Jacobian norm (Novak et al. 2018) and Cook's influence (Cook 1986), are variant to scaling reparameterization of the perturbation.

Jacobian norm: $$\|\mathbf{J}(\boldsymbol{\alpha})\|_F := \left[\sum_{i=1}^{p}\left(\frac{\partial f}{\partial \alpha_i}\right)^2\right]^{1/2} = k\|\mathbf{J}(\boldsymbol{\alpha}')\|_F \quad \text{for} \quad \boldsymbol{\alpha}' = k\boldsymbol{\alpha}\,.$$

- The scaling-variant issue causes the misleading results for
  - Measuring the effect of the simultaneous perturbation to multiple parameters at different scales:

Weights/biases in a convolution layer	&	Shift/scale parameters in a batch normalization layer

  - Comparing the sensitivity of different DNNs, e.g., based on perturbations to:

All DNN 1's trainable parameters	vs.	All DNN 2's trainable parameters

Method

- **Aim**: develop a novel perturbation manifold and its associated local influence measure to evaluate various perturbations.

- **Invariance property**: our proposed measure is invariant under any diffeomorphic (e.g., scaling) reparameterization of the perturbation.

- **Applications**:
  - Detect potential “outliers”
  - Analyze the sensitivity of model architectures
  - Compare network’s sensitivity b/w training & test sets
  - Locate vulnerable areas

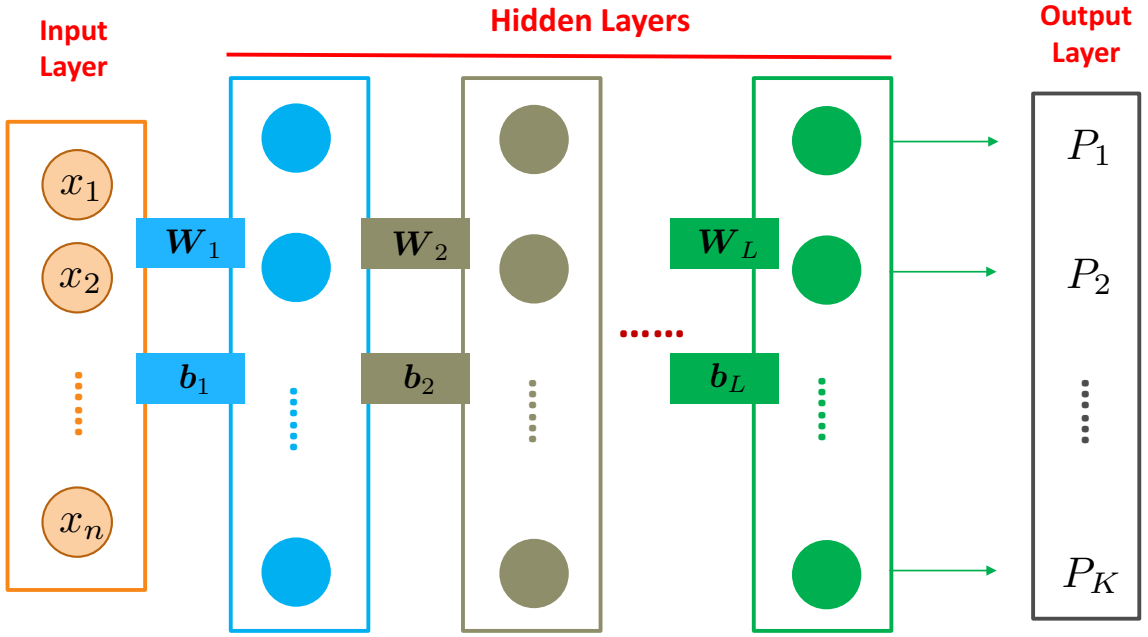
Perturbation manifold

- DNN's prediction probability: $P(y|\boldsymbol{x}, \boldsymbol{\theta})$

  $y \in \{1, \ldots, K\}$: class label

  $\boldsymbol{x}$: input image

  $\boldsymbol{\theta}$: trainable parameters



- Perturbation:

  $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)^T$: a perturbation vector

  $P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$: perturbed DNN's prediction probability s.t.

$$P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega}_0) = P(y|\boldsymbol{x}, \boldsymbol{\theta})$$

$\{P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})\}_{y=1}^{K}$ are positive and sufficiently smooth for all $\boldsymbol{\omega} \in \Omega$.

Perturbation manifold (cont.)

- Evaluate the discrepancy of the objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ by

$$\frac{[f(\boldsymbol{\omega}_1) - f(\boldsymbol{\omega}_2)]^2}{d^2(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)} \quad .$$

Perturbation manifold (cont.)

- Evaluate the discrepancy of the objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ by

$$\frac{[f(\boldsymbol{\omega}_1) - f(\boldsymbol{\omega}_2)]^2}{d^2(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)} .$$

- The above measure is scaling-variant if $d(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \|\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2\|$.

Perturbation manifold (cont.)

- Evaluate the discrepancy of the objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ by

$$\frac{[f(\boldsymbol{\omega}_1) - f(\boldsymbol{\omega}_2)]^2}{d^2(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)} .$$

- The above measure is scaling-variant if $d(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \|\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2\|$.

- Following Zhu et al. (2007) and Zhu, Ibrahim, and Tang (2011), design $d(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ on the manifold

$$\mathcal{M} = \{P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega\} .$$

Perturbation manifold (cont.)

- However, the metric tensor of $\mathcal{M} = \{P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega\}$:

$$\mathbf{G}_{\boldsymbol{\omega}}(\boldsymbol{\omega}) = \sum_{y=1}^{K} \partial_{\boldsymbol{\omega}}^{T} \ell(\boldsymbol{\omega}|y, \boldsymbol{x}, \boldsymbol{\theta}) \partial_{\boldsymbol{\omega}} \ell(\boldsymbol{\omega}|y, \boldsymbol{x}, \boldsymbol{\theta}) P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$$

is singular if $p > K$, so $\mathcal{M}$ is not a Riemannian manifold and its associated distance $d(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is not valid.

Notation: $\partial_{\boldsymbol{\omega}} = (\partial/\partial\omega_1, \ldots, \partial/\partial\omega_p)$

$$\ell(\boldsymbol{\omega}|y, \boldsymbol{x}, \boldsymbol{\theta}) = \log P(\boldsymbol{\omega}|y, \boldsymbol{x}, \boldsymbol{\theta})$$

Perturbation manifold (cont.)

Our solution:

- Use a low-dimensional transform

$$\boldsymbol{\nu} = \boldsymbol{\Lambda}_0^{1/2} \mathbf{U}_0^T \boldsymbol{\omega}$$

where $\boldsymbol{\Lambda}_0$ and $\mathbf{U}_0$ can be obtained from the compact SVD of the $p \times p$ matrix $\mathbf{G}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \mathbf{U}_0 \boldsymbol{\Lambda}_0 \mathbf{U}_0^T$ by our fast method with time complexity $O(\min\{pK^2, Kp^2\})$.

Perturbation manifold (cont.)

Our solution:

- Use a low-dimensional transform

$$\boldsymbol{\nu} = \boldsymbol{\Lambda}_0^{1/2} \mathbf{U}_0^T \boldsymbol{\omega}$$

where $\boldsymbol{\Lambda}_0$ and $\mathbf{U}_0$ can be obtained from the compact SVD of the $p \times p$ matrix $\mathbf{G}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \mathbf{U}_0 \boldsymbol{\Lambda}_0 \mathbf{U}_0^T$ by our fast method with time complexity $O(\min\{pK^2, Kp^2\})$.

- Let $P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\nu}) = P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega} = \mathbf{U}_0 \boldsymbol{\Lambda}_0^{-1/2} \boldsymbol{\nu} + \boldsymbol{\xi}_0)$ with $\boldsymbol{\xi}_0 = \boldsymbol{\omega}_0 - \mathbf{U}_0 \boldsymbol{\Lambda}_0^{-1/2} \boldsymbol{\nu}_0$.

$\mathbf{G}_{\boldsymbol{\nu}}(\boldsymbol{\nu}_0) = \mathbf{I}$ and $\mathbf{G}_{\boldsymbol{\nu}}(\boldsymbol{\nu}) \succ 0$ in an open ball $B_{\boldsymbol{\nu}_0}$ centered at $\boldsymbol{\nu}_0 = \boldsymbol{\Lambda}_0^{1/2} \mathbf{U}_0^T \boldsymbol{\omega}_0$.

Perturbation manifold (cont.)

Our solution:

- Use a low-dimensional transform

$$\boldsymbol{\nu} = \boldsymbol{\Lambda}_0^{1/2} \mathbf{U}_0^T \boldsymbol{\omega}$$

where $\boldsymbol{\Lambda}_0$ and $\mathbf{U}_0$ can be obtained from the compact SVD of the $p \times p$ matrix $\mathbf{G}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \mathbf{U}_0 \boldsymbol{\Lambda}_0 \mathbf{U}_0^T$ by our fast method with time complexity $O(\min\{pK^2, Kp^2\})$.

- Let $P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\nu}) = P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega} = \mathbf{U}_0 \boldsymbol{\Lambda}_0^{-1/2} \boldsymbol{\nu} + \boldsymbol{\xi}_0)$ with $\boldsymbol{\xi}_0 = \boldsymbol{\omega}_0 - \mathbf{U}_0 \boldsymbol{\Lambda}_0^{-1/2} \boldsymbol{\nu}_0$.

$\mathbf{G}_{\boldsymbol{\nu}}(\boldsymbol{\nu}_0) = \mathbf{I}$ and $\mathbf{G}_{\boldsymbol{\nu}}(\boldsymbol{\nu}) \succ 0$ in an open ball $B_{\boldsymbol{\nu}_0}$ centered at $\boldsymbol{\nu}_0 = \boldsymbol{\Lambda}_0^{1/2} \mathbf{U}_0^T \boldsymbol{\omega}_0$.

- Perturbation manifold around $\boldsymbol{\nu}_0$:

$$\mathcal{M}_{\boldsymbol{\nu}_0} = \{p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\nu}) : \boldsymbol{\nu} \in B_{\boldsymbol{\nu}_0}\}$$

Local influence measure

- Evaluate the discrepancy of the objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ by

$$\mathrm{I}_C(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \frac{[f(\boldsymbol{\omega}_1) - f(\boldsymbol{\omega}_2)]^2}{S_C^2(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)}$$

where $S_C(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)$ is the distance between $\boldsymbol{\nu}_i = \boldsymbol{\Lambda}_0^{1/2}\mathbf{U}_0^T\boldsymbol{\omega}_i, i = 1, 2$ along a smooth curve $C(t) = P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\nu}(t))$ on $\mathcal{M}_{\boldsymbol{\nu}_0}$.

Local influence measure

- Evaluate the discrepancy of the objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ by

$$\mathrm{I}_C(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \frac{[f(\boldsymbol{\omega}_1) - f(\boldsymbol{\omega}_2)]^2}{S_C^2(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)}$$

  where $S_C(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)$ is the distance between $\boldsymbol{\nu}_i = \boldsymbol{\Lambda}_0^{1/2}\mathbf{U}_0^T\boldsymbol{\omega}_i, i = 1, 2$ along a smooth curve $C(t) = P(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\nu}(t))$ on $\mathcal{M}_{\boldsymbol{\nu}_0}$.

- Our local influence measure for any objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_0$:

$$\mathrm{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \max_C \lim_{t \to 0} \mathrm{I}_C(\boldsymbol{\omega}(t), \boldsymbol{\omega}(0)) = \nabla_{f(\boldsymbol{\nu}_0)} \nabla_{f(\boldsymbol{\nu}_0)}^T$$

  with $\nabla_{f(\boldsymbol{\nu}_0)} := \partial_{\boldsymbol{\omega}} f(\boldsymbol{\omega}_0)\mathbf{U}_0\boldsymbol{\Lambda}_0^{-1/2}$.

Local influence measure (cont.)

- Our local influence measure for any objective function $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_0$:

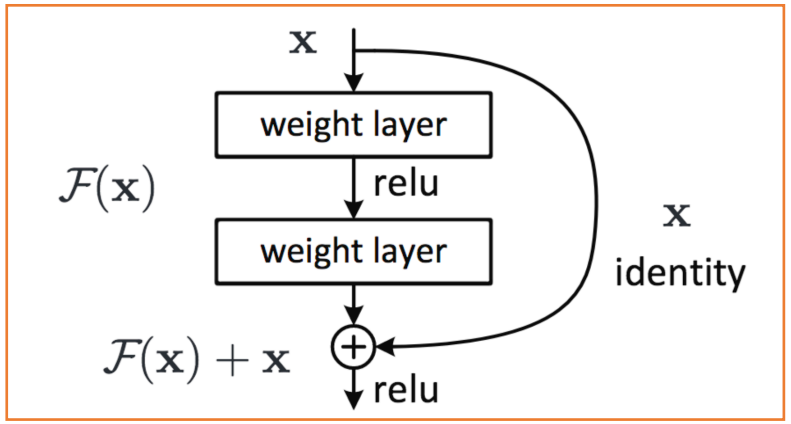
$$\mathrm{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \max_{C} \lim_{t \to 0} \mathrm{I}_C(\boldsymbol{\omega}(t), \boldsymbol{\omega}(0)) = \nabla_{f(\boldsymbol{\nu}_0)} \nabla_{f(\boldsymbol{\nu}_0)}^T$$

with $\nabla_{f(\boldsymbol{\nu}_0)} := \partial_{\boldsymbol{\omega}} f(\boldsymbol{\omega}_0) \mathbf{U}_0 \boldsymbol{\Lambda}_0^{-1/2}$.

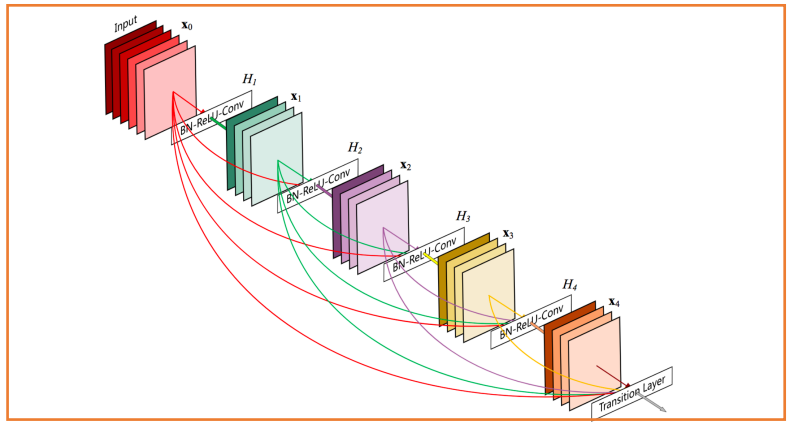
- Invariance property:

Theorem 1. *Suppose that $\boldsymbol{\varphi}$ is a diffeomorphism of $\boldsymbol{\omega}$. Then, $\mathrm{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0)$ is invariant with respect to any reparameterization corresponding to $\boldsymbol{\varphi}$.*

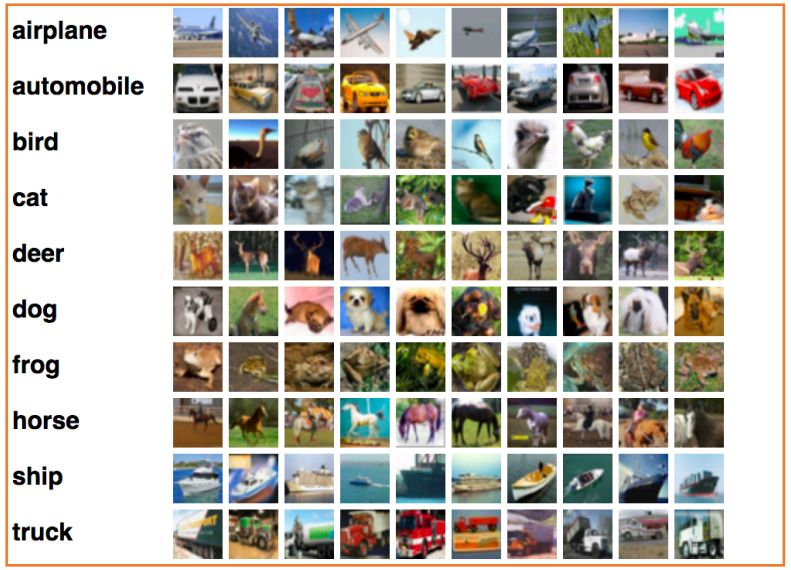
Experiments



Building block of ResNet50
(He et al. 2016)

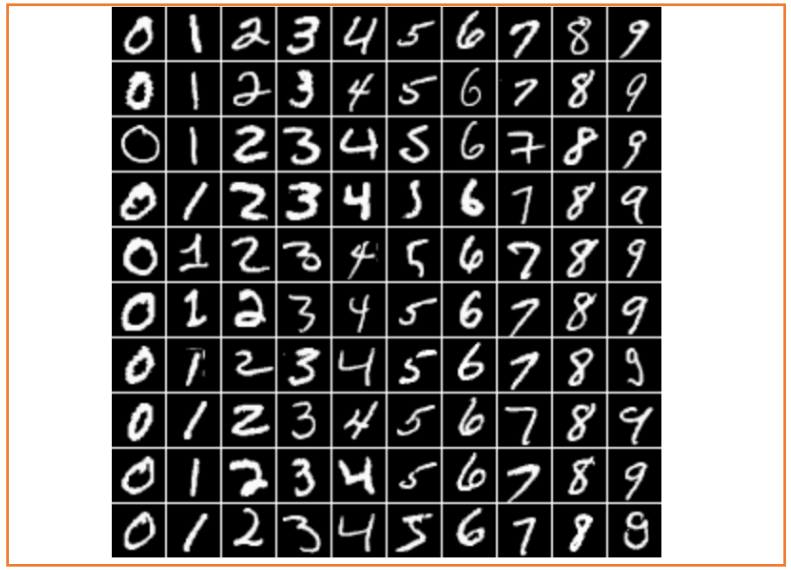


Building block of DenseNet121
(Huang et al. 2017)



CIFAR10: train/val/test=45k/5k/10k

https://www.cs.toronto.edu/~kriz/cifar.html



MNIST: train/val/test=54k/6k/10k

http://yann.lecun.com/exdb/mnist/

Experiments (cont.)

- Perturbation cases:

DNN's prediction probability: $P(y|\boldsymbol{x}, \boldsymbol{\theta})$

y : class label $\quad$ $\boldsymbol{x}$: input image

$\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_L^T)^T$: DNN's trainable parameters

$\boldsymbol{\theta}_l$: parameters of l-th DNN's trainable layer

$$\text{Case 1} : \boldsymbol{x} + \Delta\boldsymbol{x}; \quad \text{Case 2} : \boldsymbol{\theta} + \Delta\boldsymbol{\theta}; \quad \text{and} \quad \text{Case 3} : \boldsymbol{\theta}_l + \Delta\boldsymbol{\theta}_l$$

Table 1: Accuracy for models trained without data augmentation

	CIFAR10		MNIST	
Model	Training	Testing	Training	Testing
ResNet50	99.78%	88.70%	99.87%	99.29%
DenseNet121	99.87%	91.16%	99.998%	99.58%

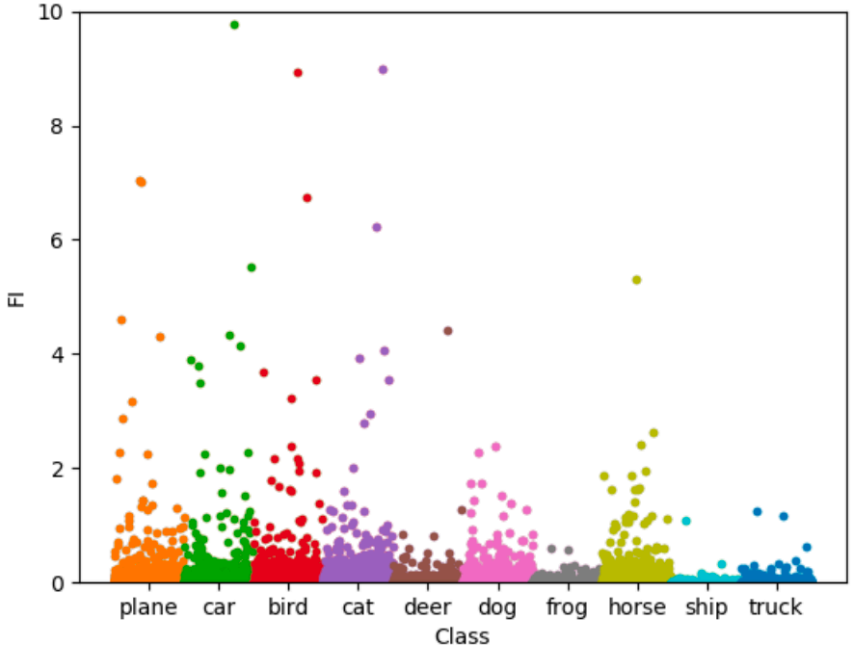
Experiments (cont.)

1. Detect potential outliers:

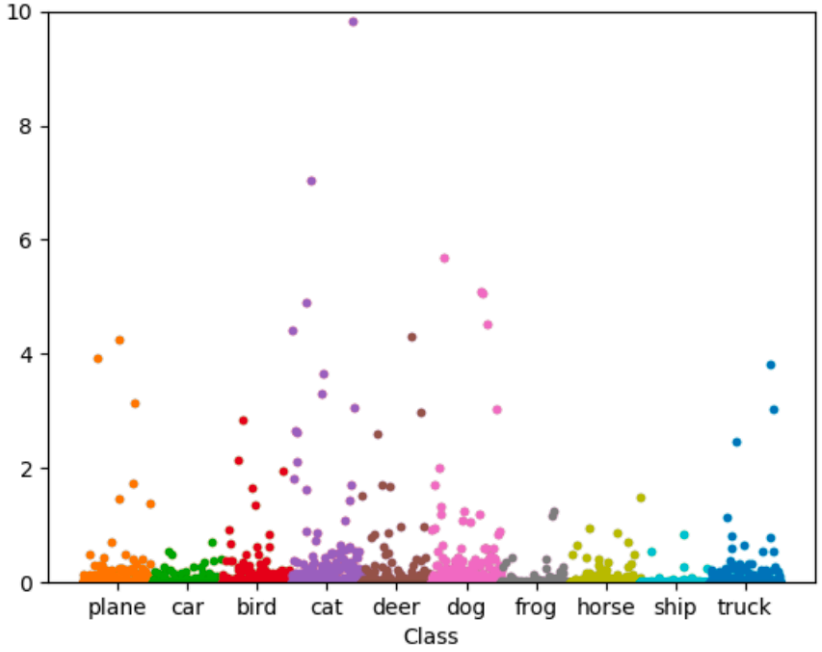
Compute each training image's FI under $\boldsymbol{x} + \Delta\boldsymbol{x}$ with $f = -\log P(y = y_{\text{true}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.

(R) car→truck FI=9.79 | (R) cat→dog FI=8.99 | (R) bird→dog FI=8.93 | (R) plane→ship FI=7.04 | (R) plane→ship FI=7.02

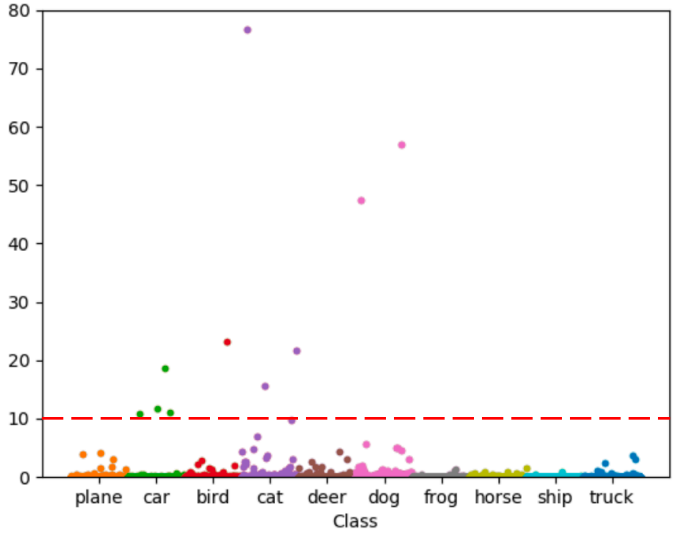
(D) cat→horse FI=76.7 | (D) dog→bird FI=57.0 | (D) dog→bird FI=47.5 | (D) bird→plane FI=23.2 | (D) cat→deer FI=21.6



(a) All FIs for ResNet50 on CIFAR10



(b) FIs$\leq$10 for DenseNet121 on CIFAR10

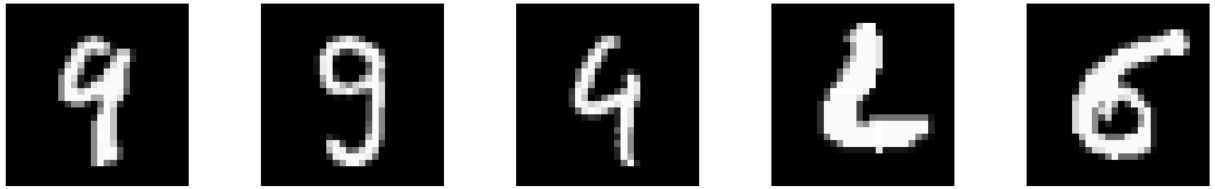


(c) All FIs for DenseNet121 on CIFAR10

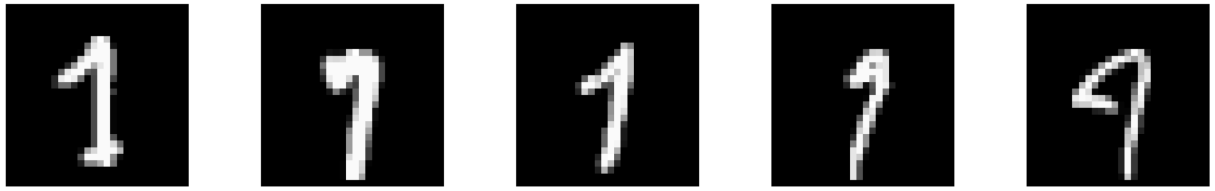
Experiments (cont.)

1. Detect potential outliers:

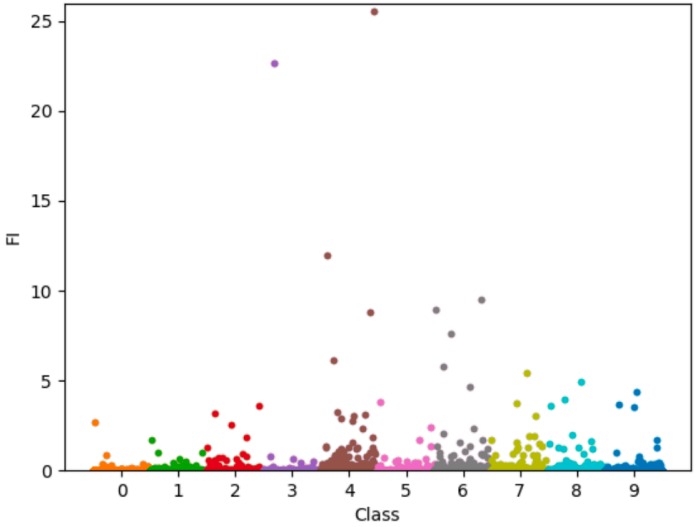
Compute each training image's FI under $\boldsymbol{x} + \Delta\boldsymbol{x}$ with $f = -\log P(y = y_{\text{true}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.



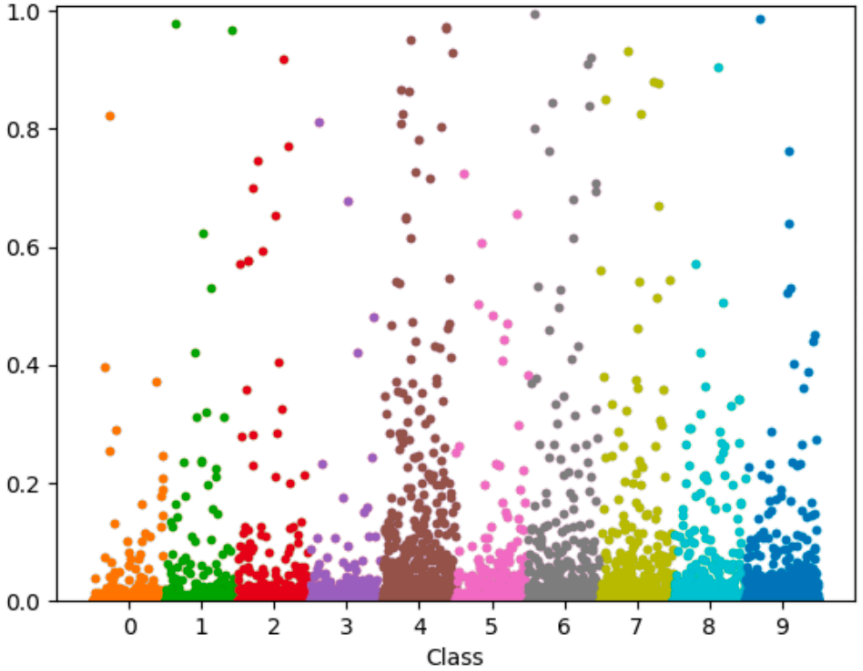
(R) 4 → 9 FI=25.6 | (R) 3 → 9 FI=22.6 | (R) 4 → 9 FI=12.0 | (R) 6 → 2 FI=9.49 | (R) 6 → 5 FI=8.92



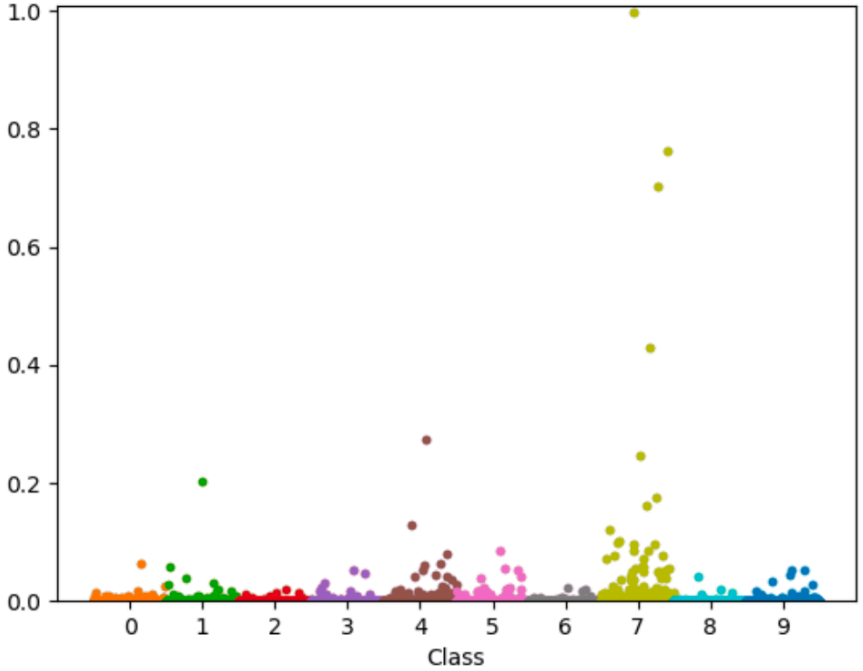
(D) 7 → 7 FI=0.998 | (D) 7 → 7 FI=0.763 | (D) 7 → 7 FI=0.703 | (D) 7 → 7 FI=0.430 | (D) 4 → 4 FI=0.273



(d) All FIs for ResNet50 on MNIST



(e) FIs≤1 for ResNet50 on MNIST



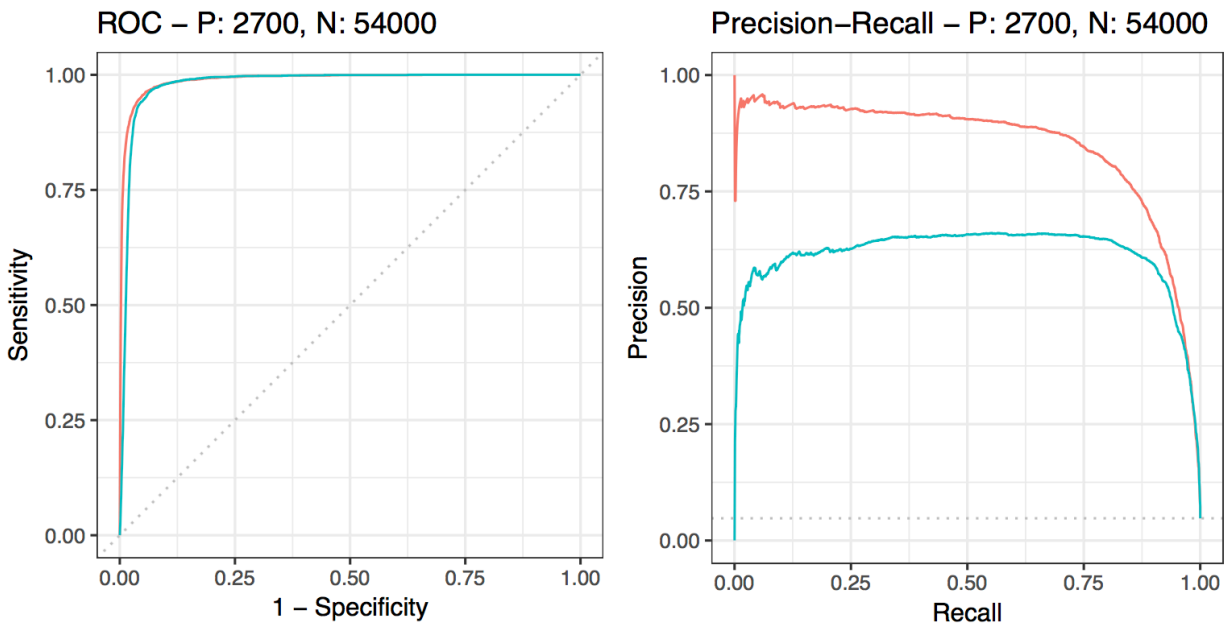
(f) All FIs for DenseNet121 on MNIST

Experiments (cont.)

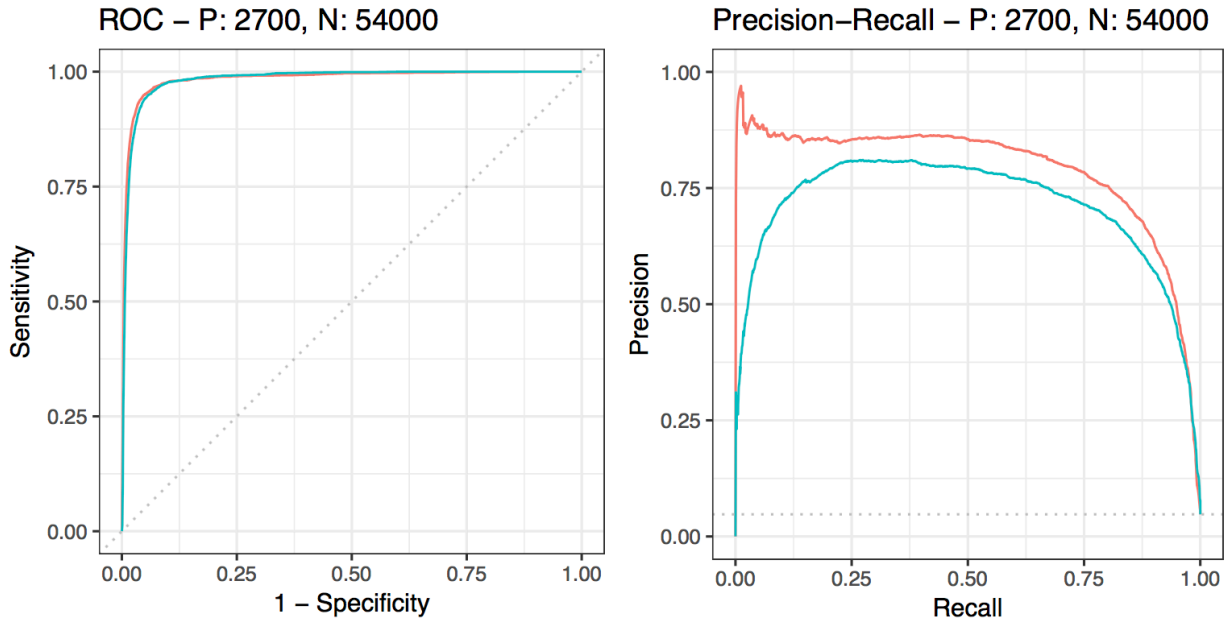
1. Detect potential outliers:

FI measure vs. *Jacobian norm* in detecting simulated outliers (images with overlapped digits)

Simulated outliers



(a) ResNet50



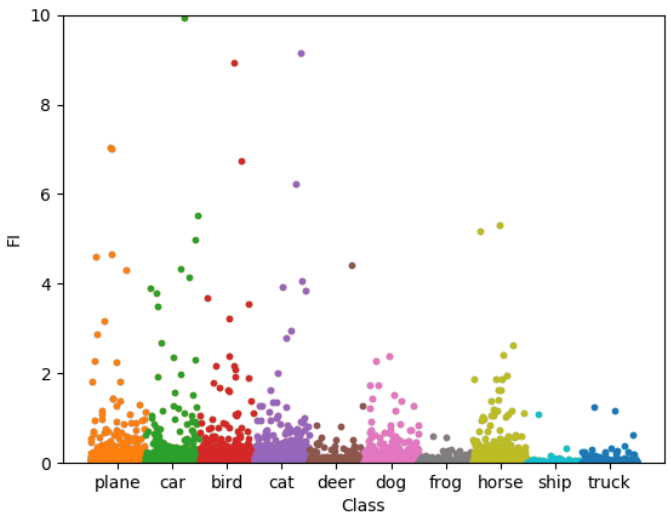
(b) DenseNet121

Our FI (red); Jacobian norm (blue)

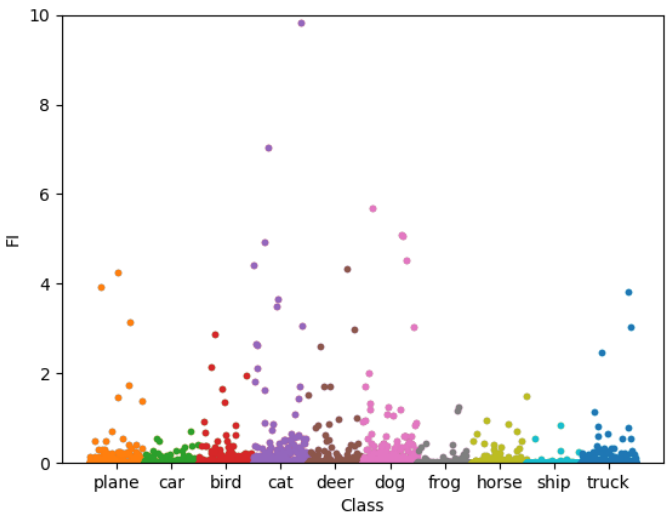
Experiments (cont.)

2. Analyze the sensitivity of model architectures:

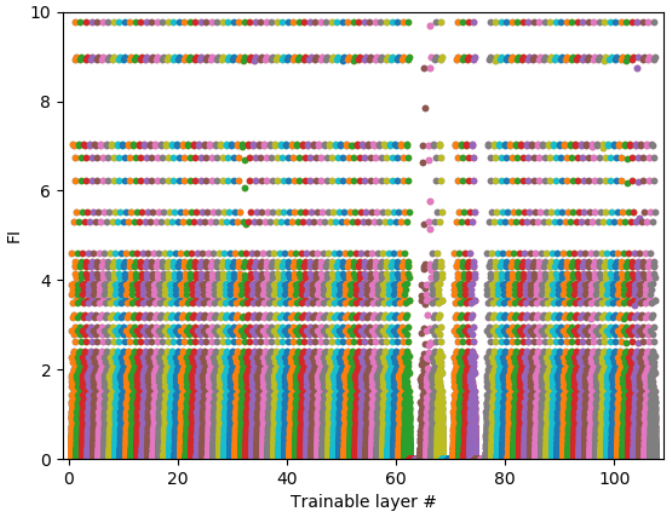
Compute each training image's FI under
(1) $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$ or
(2) $\boldsymbol{\theta}_l + \Delta\boldsymbol{\theta}_l$ with
$f = -\log P(y = y_{\text{true}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.



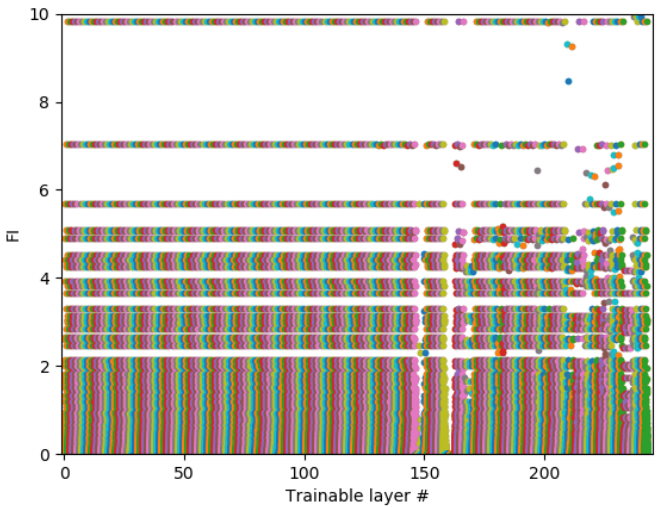
(a) ResNet50 on CIFAR10 for (1)



(b) DenseNet121 on CIFAR10 for (1)



(c) ResNet50 on CIFAR10 for (2)

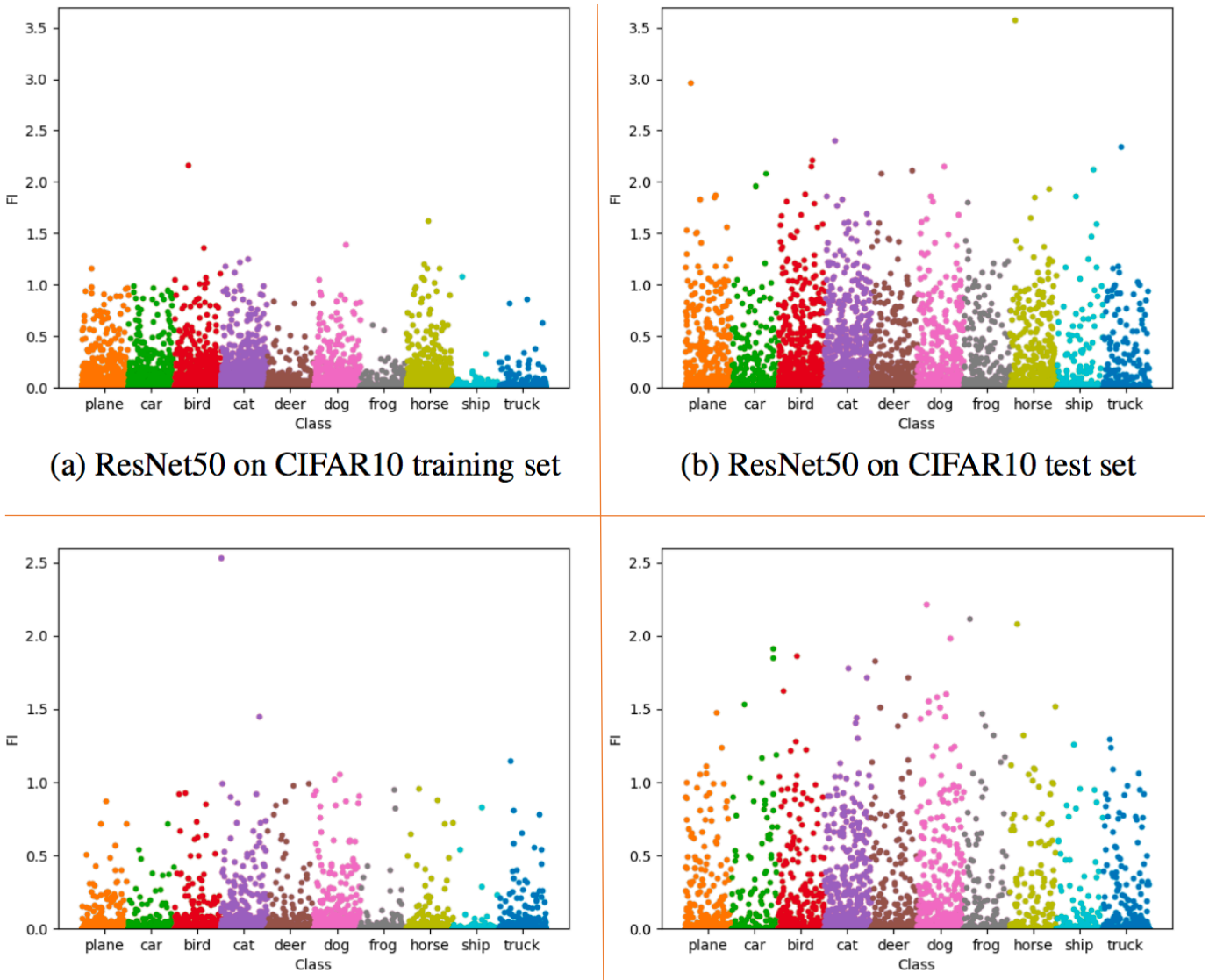


(d) DenseNet121 on CIFAR10 for (2)

Experiments (cont.)

3. Compare network sensitivity b/w training & test sets:

Compute each image's FI under $\boldsymbol{x} + \Delta\boldsymbol{x}$ for training/test sets with $f = -\log P(y = y_{\text{pred}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.



(a) ResNet50 on CIFAR10 training set

(b) ResNet50 on CIFAR10 test set

(c) DenseNet121 on CIFAR10 training set

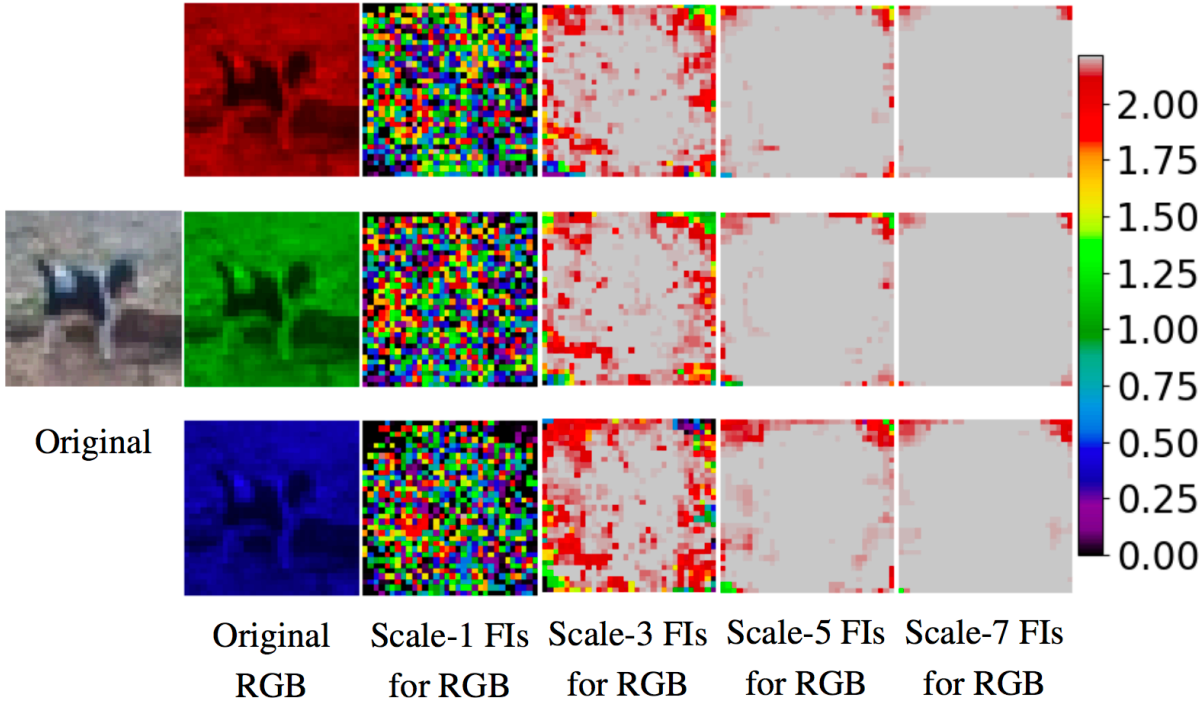
(d) DenseNet121 on CIFAR10 test set

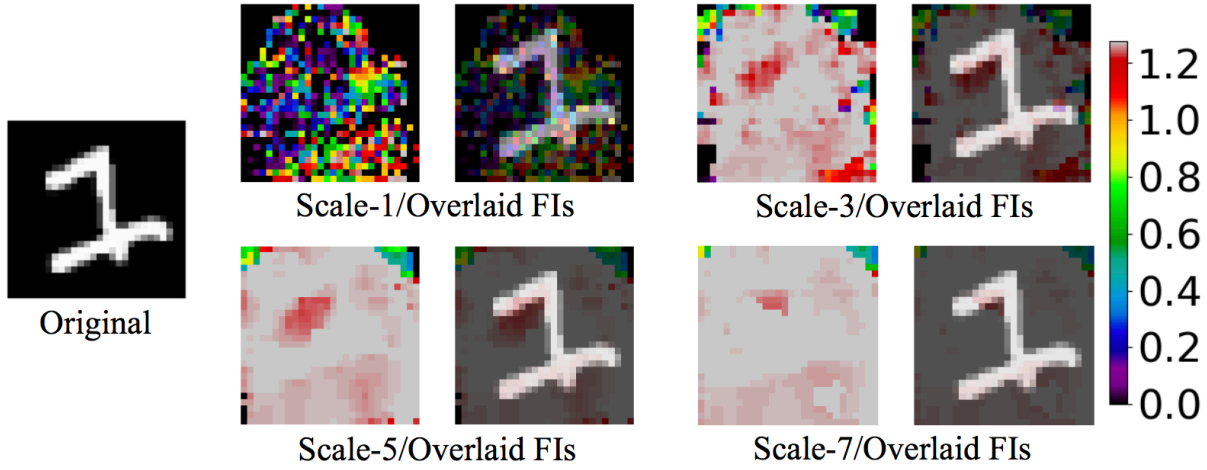
Experiments (cont.)

4. Locate vulnerable areas:

Let $f = -\log P(y = y_{\text{pred}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.
Compute each pixel's FI
under $\boldsymbol{x}_{\mathcal{N}(i)} + \Delta \boldsymbol{x}_{\mathcal{N}(i)}$.

Scale-k $\mathcal{N}(i)$:
the k x k square centered at pixel i.



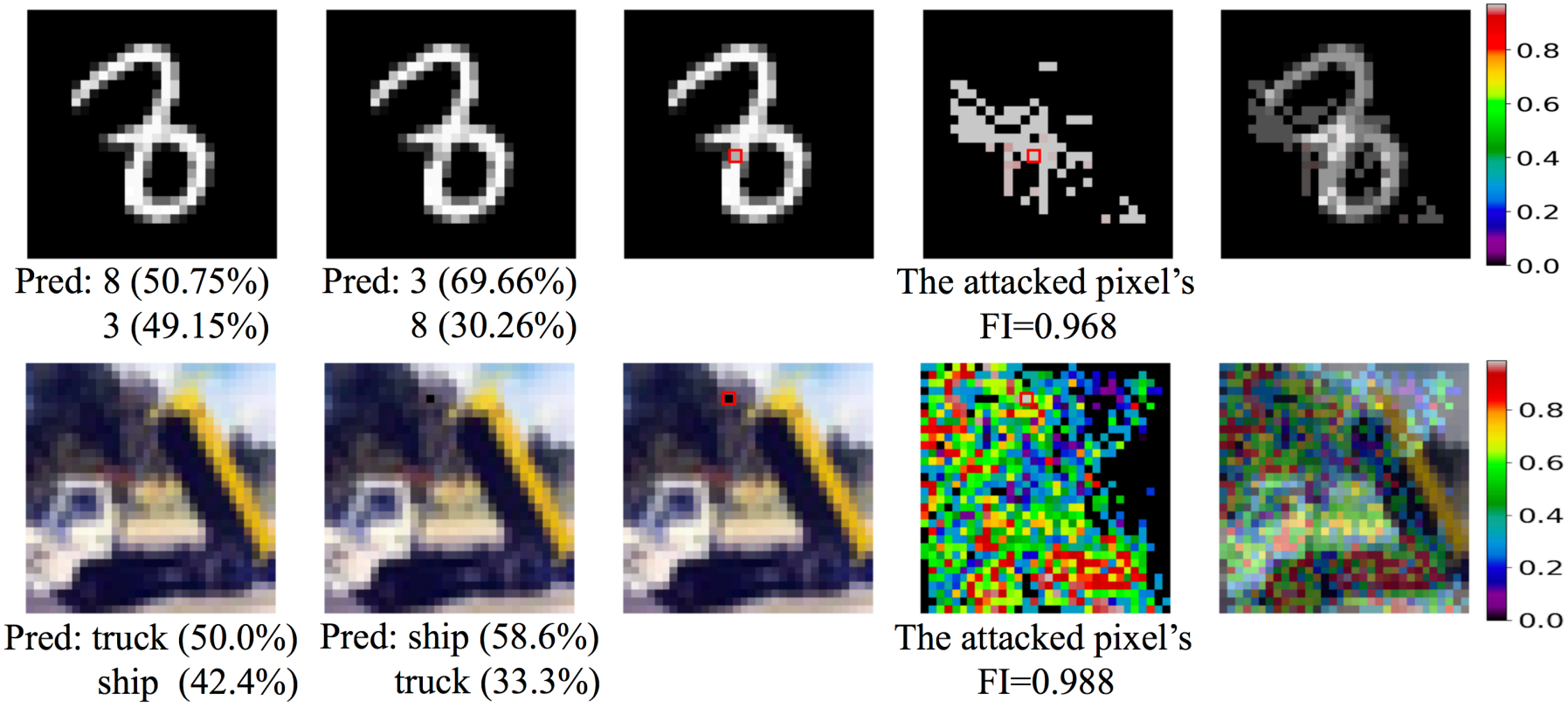


Experiments (cont.)

4. Locate vulnerable areas: Let $f = -\log P(y = y_{\text{pred}} | \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\omega})$.

Compute each pixel's FI under $\boldsymbol{x}_{\mathcal{N}(i)} + \Delta \boldsymbol{x}_{\mathcal{N}(i)}$.

Scale-1 $\mathcal{N}(i)$: the 1 x 1 square centered at pixel i .



(a) Original images (b) Adversarial images (c) Annotated Adv. images (d) Pixel-wise FI maps (e) Overlaid FI maps

References

Cook, R. D. (1986). Assessment of local influence. *Journal of the Royal Statistical Society. Series B (Methodological)* 48(2):133–169.

He, K.; Zhang, X.; Ren, S.; and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. (2017). Densely connected convolutional networks. In *CVPR*, 4700–4708.

Novak, R.; Bahri, Y.; Abolafia, D. A.; Pennington, J.; and Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: an empirical study. In *ICLR*. arXiv:1802.08760.

Zhu, H.; Ibrahim, J. G.; Lee, S.; Zhang, H.; et al. (2007). Perturbation selection and influence measures in local influence analysis. *The Annals of Statistics* 35(6):2565–2588.

Zhu, H.; Ibrahim, J. G.; and Tang, N. (2011). Bayesian influence analysis: a geometric approach. *Biometrika* 98(2):307– 323.

Thank you!

Q & A