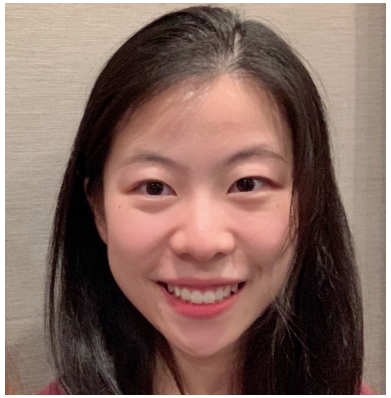
Neural networks with recurrent generative feedback

https://arxiv.org/abs/2007.09200

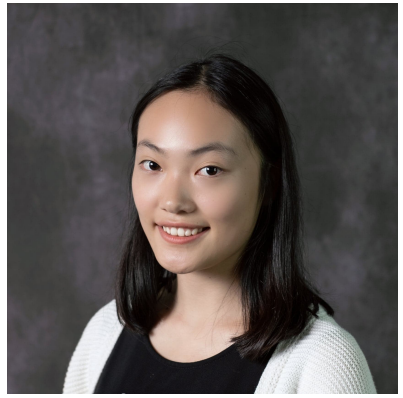
Yujia Huang, Caltech

yjhuang@caltech.edu

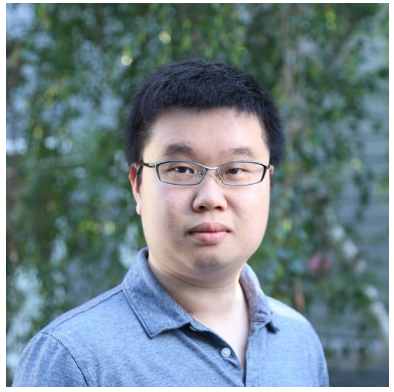


Yujia Huang, Caltech

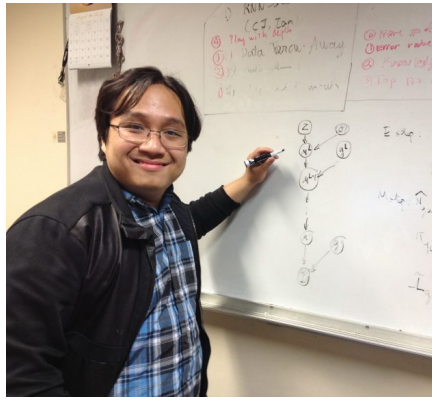
James Gornet, Caltech



Sihui Dai, Caltech



Zhiding Yu, NVIDIA

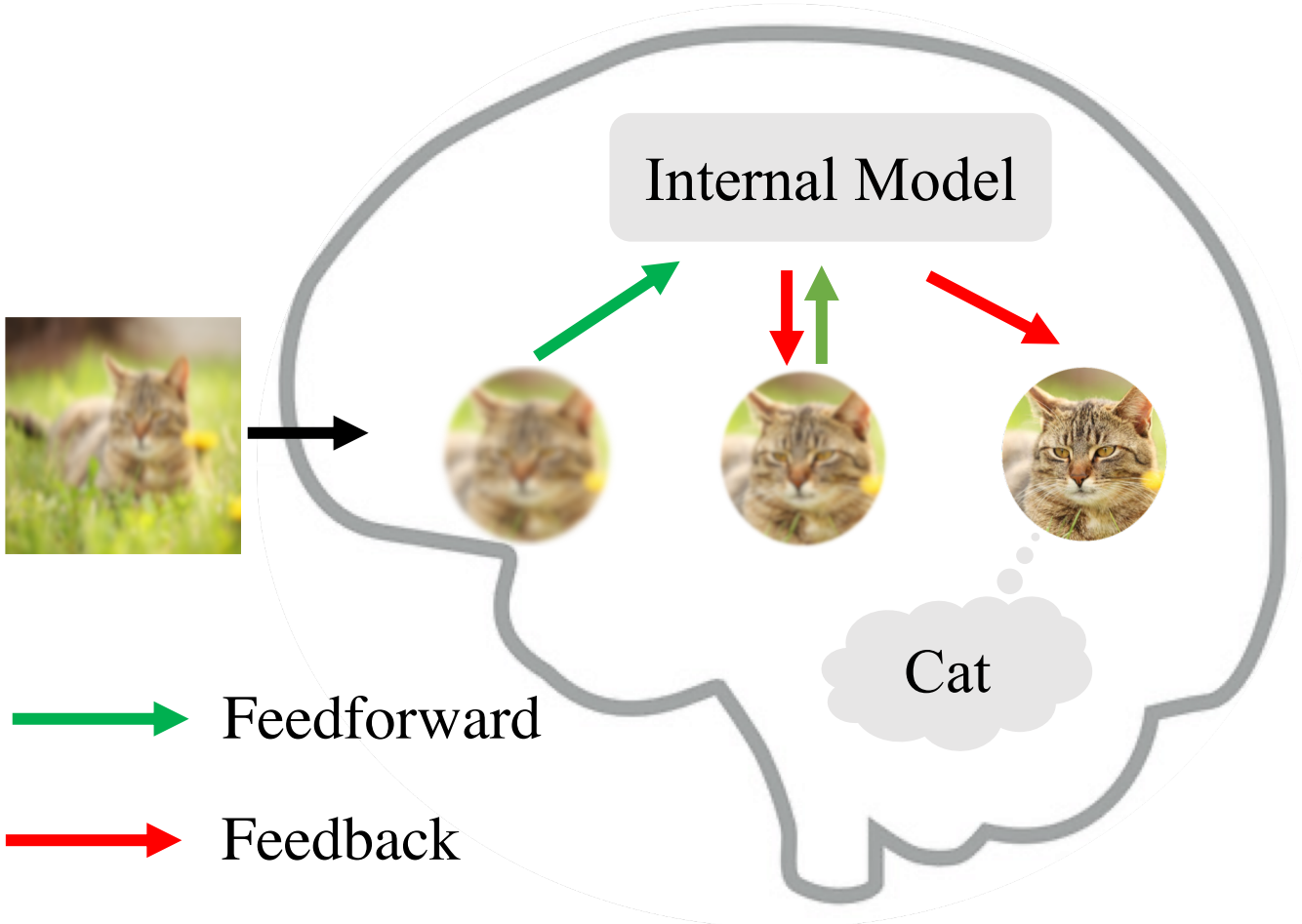


Tan Nguyen,
Rice University

Doris Y. Tsao, Caltech

Anima Anandkumar,
Caltech/NVIDIA

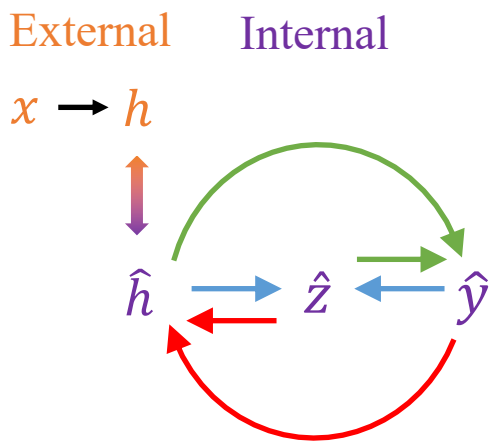
Intuition



Self-Consistency

Given a joint distribution $p(h, y, z; \theta)$ parameterized by θ, $(\hat{h}, \hat{y}, \hat{z})$ are self-consistent if they satisfy the following constraints:

$$\hat{y} = \arg\max_{y} p(y|\hat{h}, \hat{z}),$$
$$\hat{h} = \arg\max_{h} p(h|\hat{y}, \hat{z}),$$
$$\hat{z} = \arg\max_{z} p(z|\hat{h}, \hat{y})$$



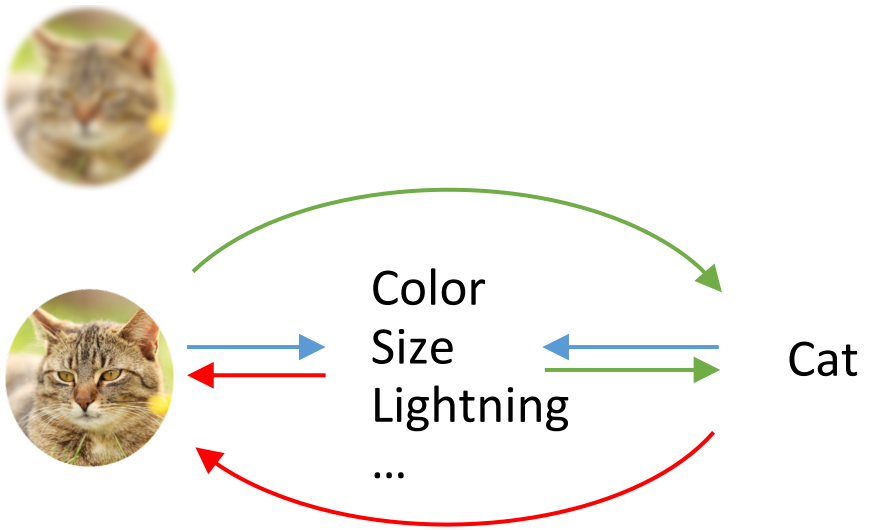
x: images

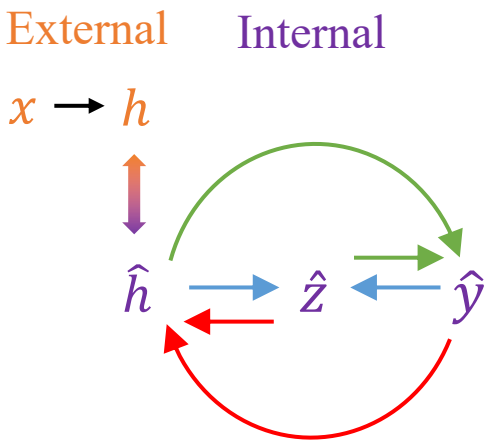
h: encoded features

z: latent variables

y: labels

Self-Consistency





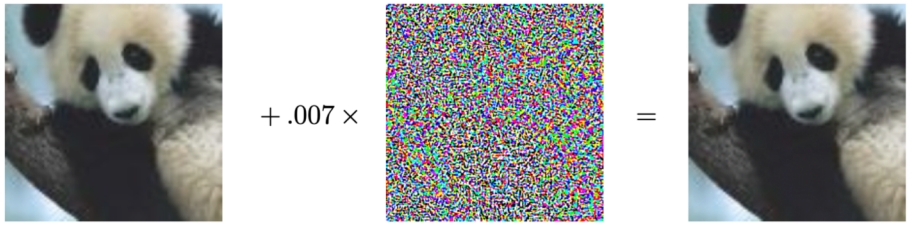
x: images

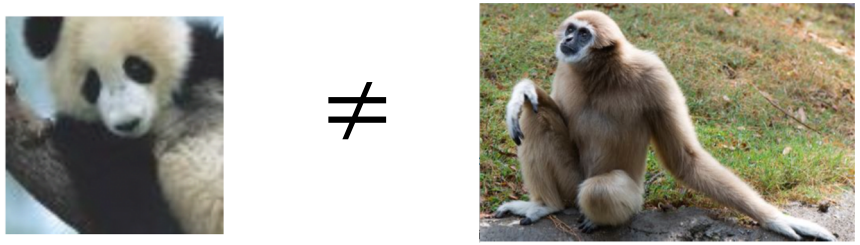
h: encoded features

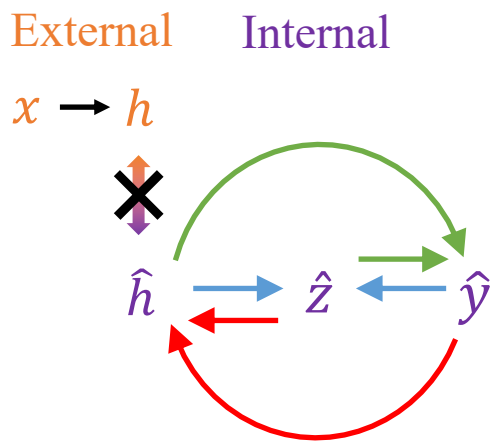
z: latent variables

y: labels

Self-Consistency







x: images

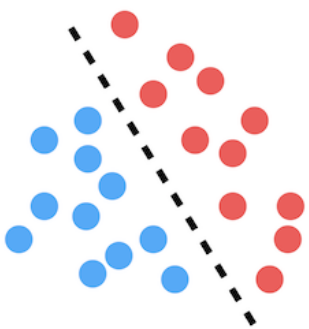
h: encoded features

z: latent variables

y: labels

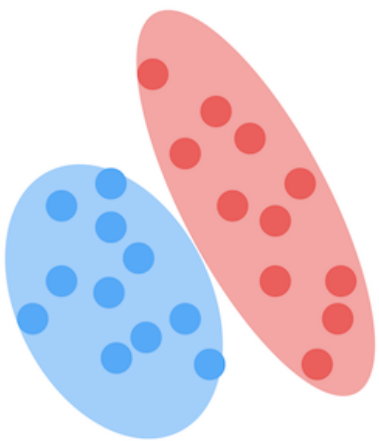
Generative Classifier

Logistic Regression



$p(y|x)$

Gaussian Naïve Classifier

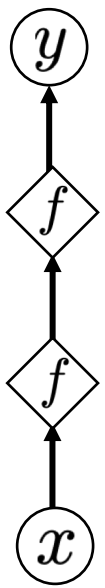


$p(x, y) \rightarrow p(y|x)$

A. Ng, and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Neurips 2002.

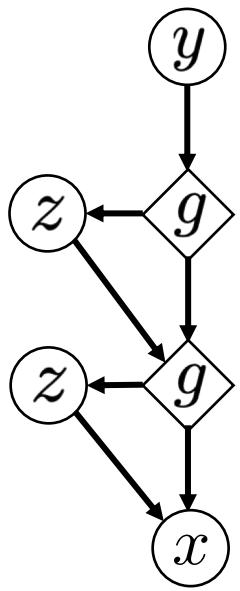
Deconvolutional generative model (DGM)

CNN



$p(y|x)$

DGM



$p(x, y, z) \rightarrow p(y|x, z)$

$$y \sim p(y)$$

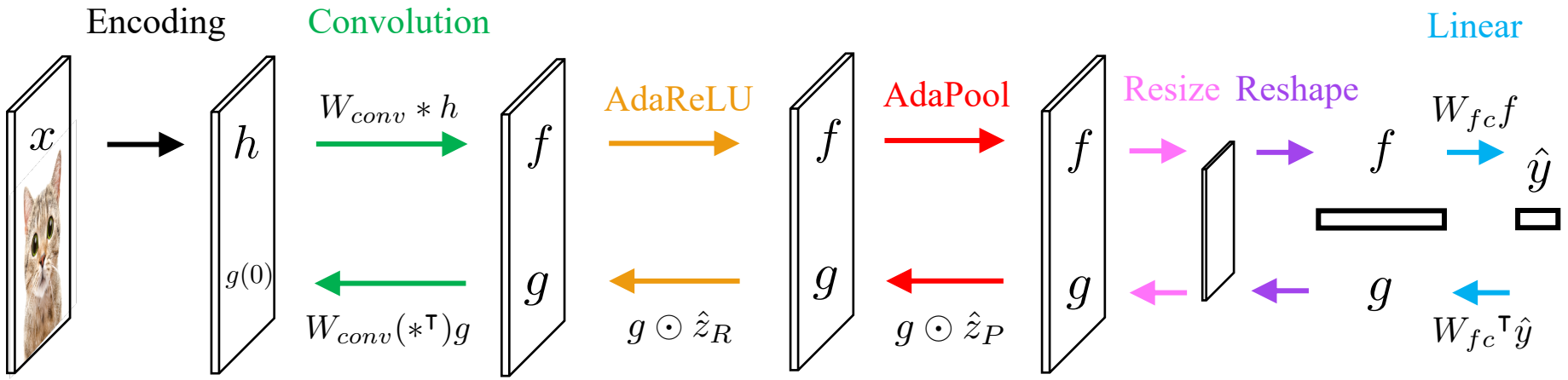
$$z_P^{(i)} \sim \text{Ber}\left(\frac{e^{b \cdot g^{(i)}}}{e^{b \cdot g^{(i)}} + 1}\right)$$

$$z_R^{(i)} \sim \text{Ber}\left(\frac{e^{b \cdot g^{(i)}}}{e^{b \cdot g^{(i)}} + 1}\right)$$

$$x \sim \mathcal{N}(g(0), \text{diag}(\sigma^2))$$

T. Nguyen, N. Ho, A. Patel, A. Anandkumar, M. I. Jordan, and R. G. Baraniuk. A bayesian perspective of convolutional neural networks through a deconvolutional generative model. arXiv:1811.02657, 2018.

Inference in the DGM



- MAP estimate of y: $\hat{y} = \mathrm{CNN}(h)$
- MAP estimate of h: $\hat{h} = g(0)$
- MAP estimate of z (informal): $\hat{z_R} = \mathbb{1}\{\sigma_{\mathrm{AdaReLU}} \neq 0\}$
  $\hat{z_P} = \mathbb{1}\{\sigma_{\mathrm{AdaPool}} \neq 0\}$

$$\sigma_{\mathrm{AdaReLU}}(f) = \begin{cases} \sigma_{\mathrm{ReLU}}(f), & \text{if } g \geq 0 \\ \sigma_{\mathrm{ReLU}}(-f), & \text{if } g < 0 \end{cases}$$

$$\sigma_{\mathrm{AdaPool}}(f) = \begin{cases} \sigma_{\mathrm{MaxPool}}(f), & \text{if } g \geq 0 \\ -\sigma_{\mathrm{MaxPool}}(-f), & \text{if } g < 0 \end{cases}$$

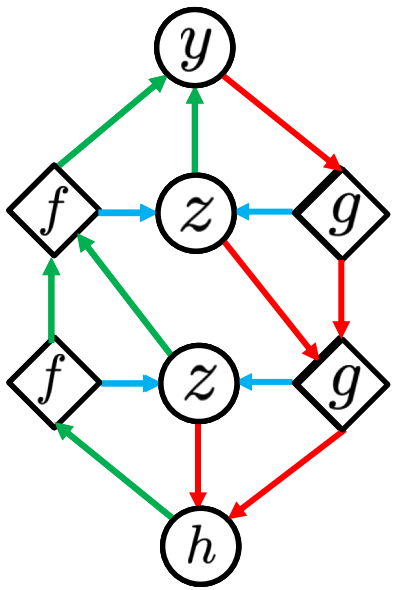
Iterative inference

Self-consistency

$$\hat{y} = \arg\max_{y} p(y|\hat{h}, \hat{z}),$$

$$\hat{h} = \arg\max_{h} p(h|\hat{y}, \hat{z}),$$

$$\hat{z} = \arg\max_{z} p(z|\hat{h}, \hat{y})$$



Feedforward

Feedback

Inference of z

y Label

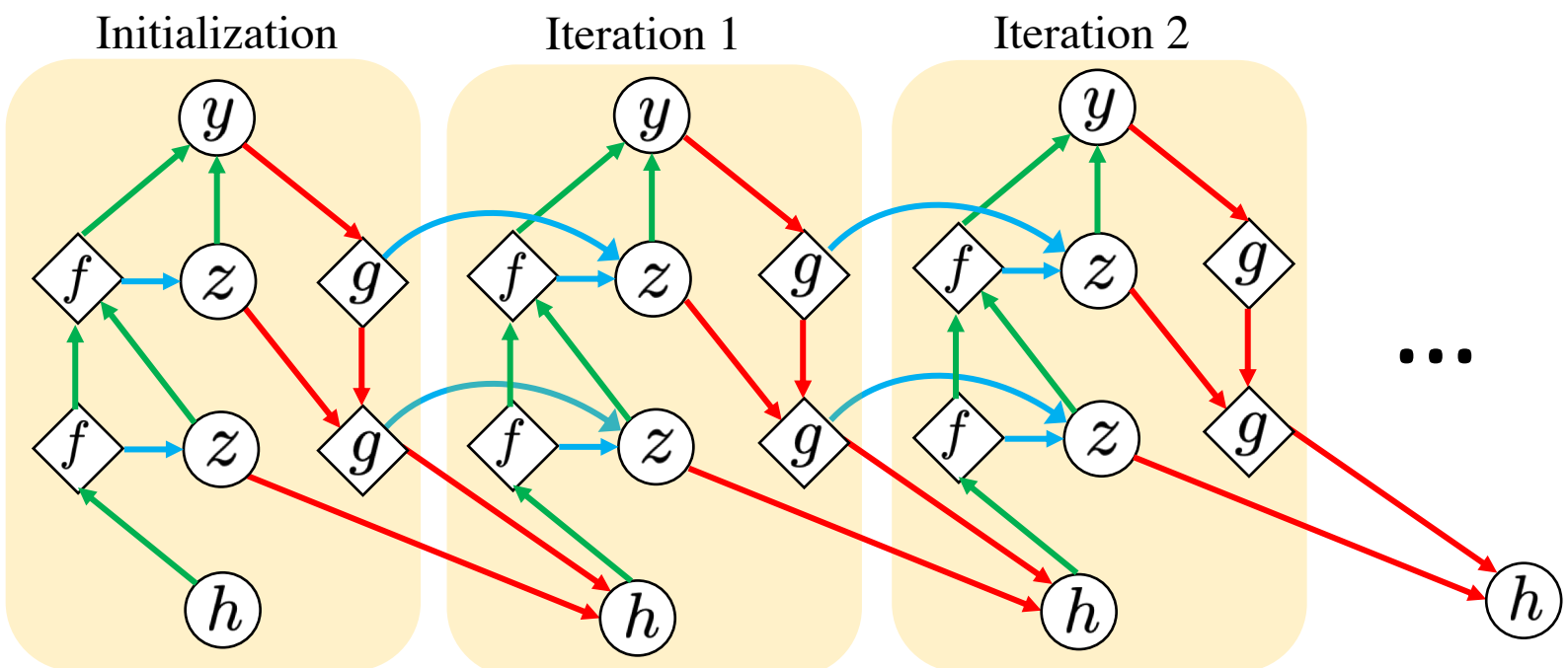
z Latent variables

h Image features

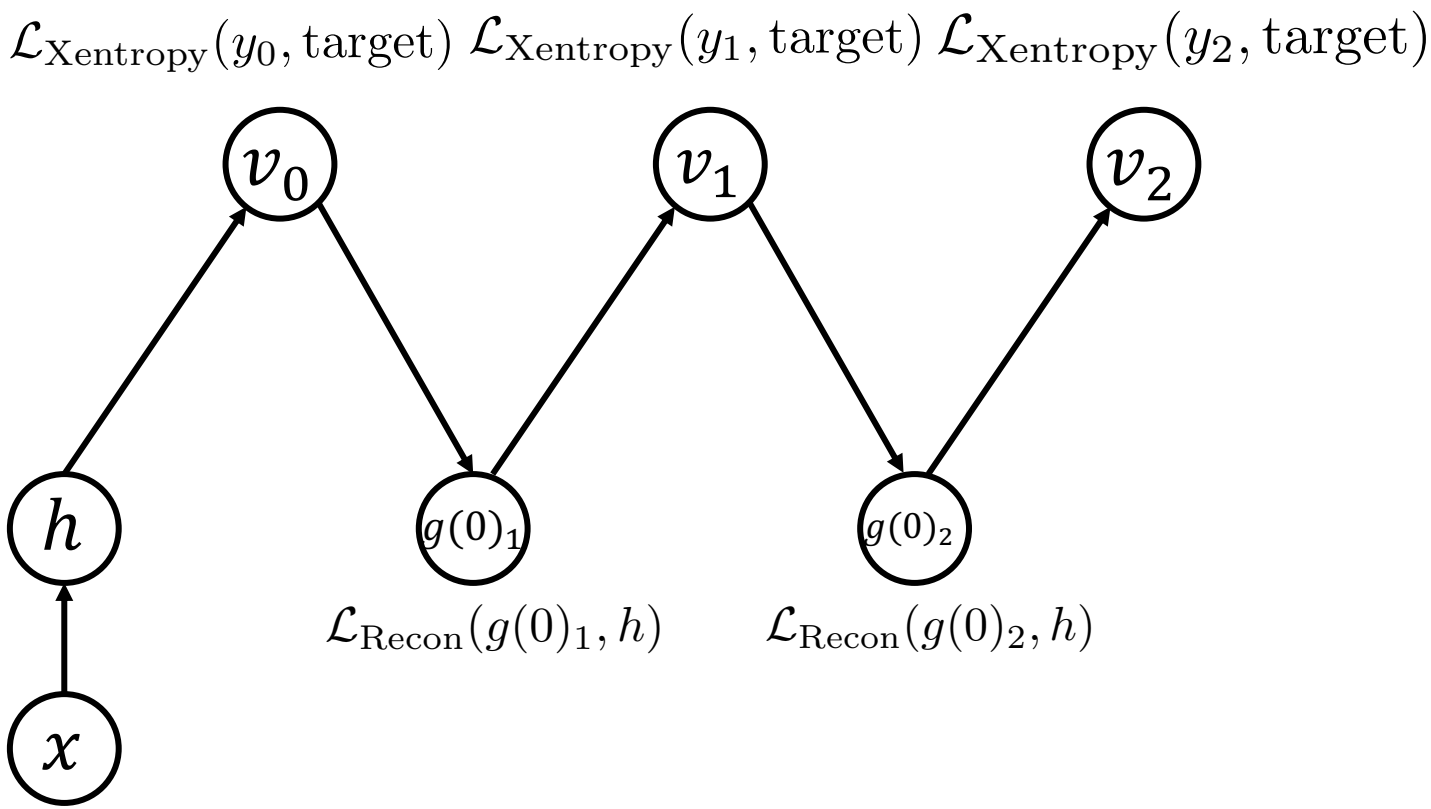
f Feedforward layer

g Feedback layer

CNN-F



Training of CNN-F



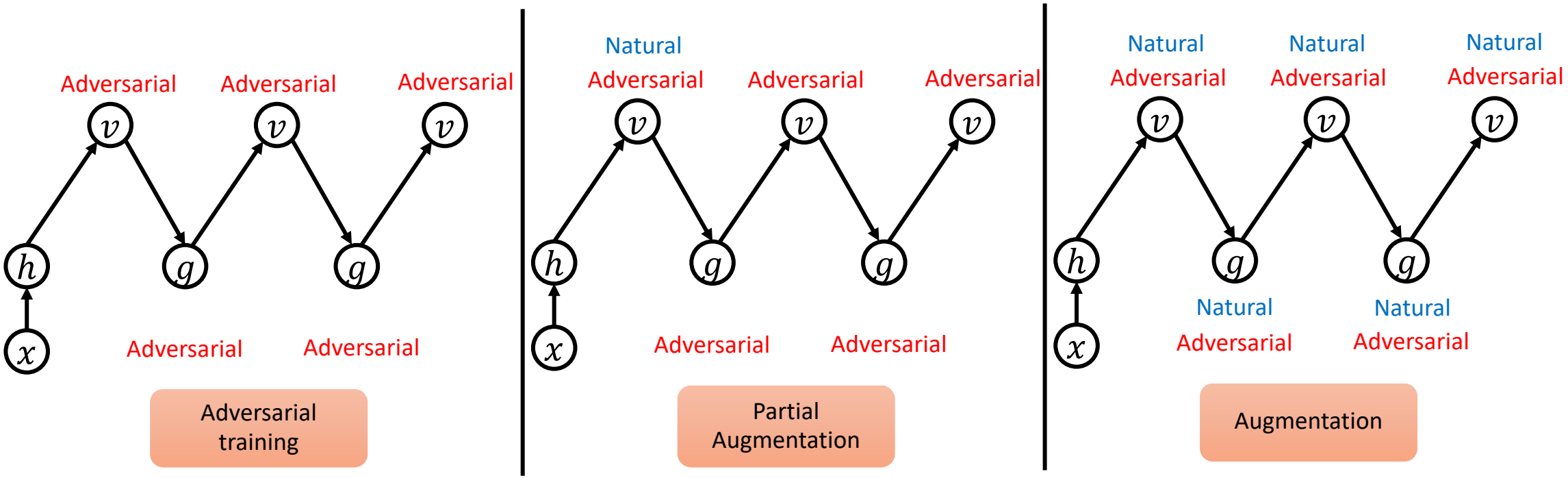
v Logits

h Image features

$g(0)$ Generated features

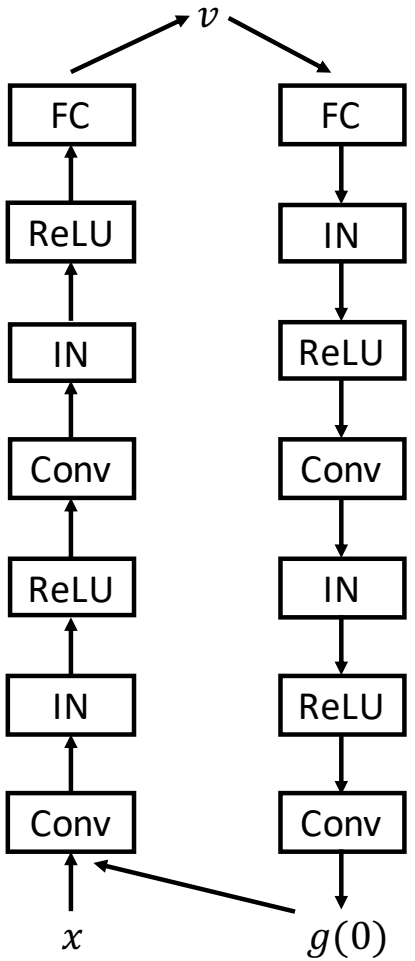
x Images

CNN-F with adversarial training

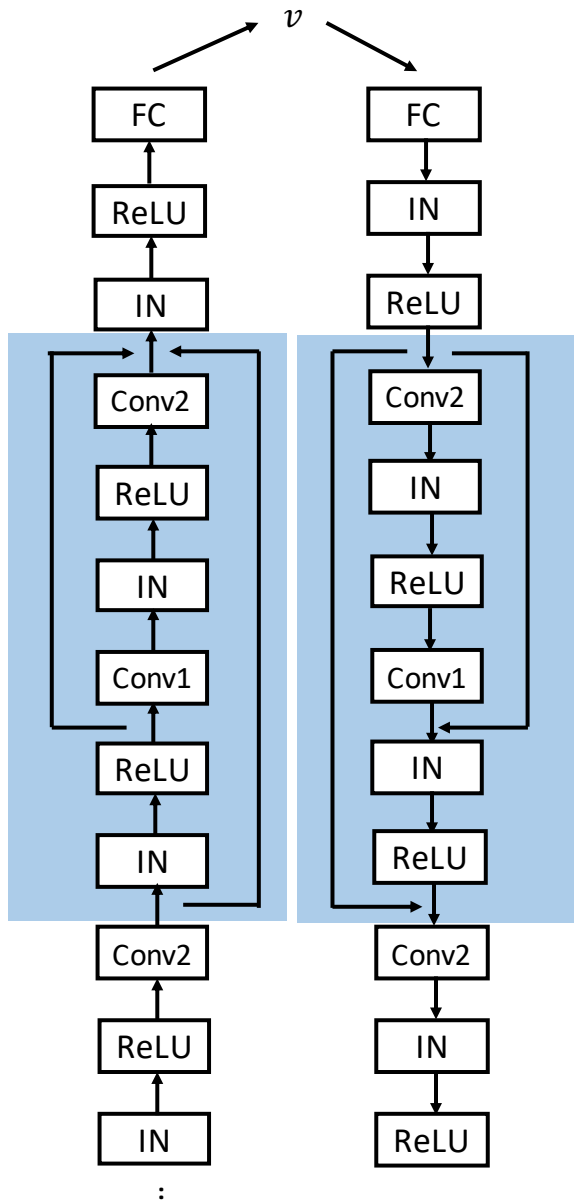


Reconstruction loss is always between adversarial and natural features.

CNN-F on all CNN architectures



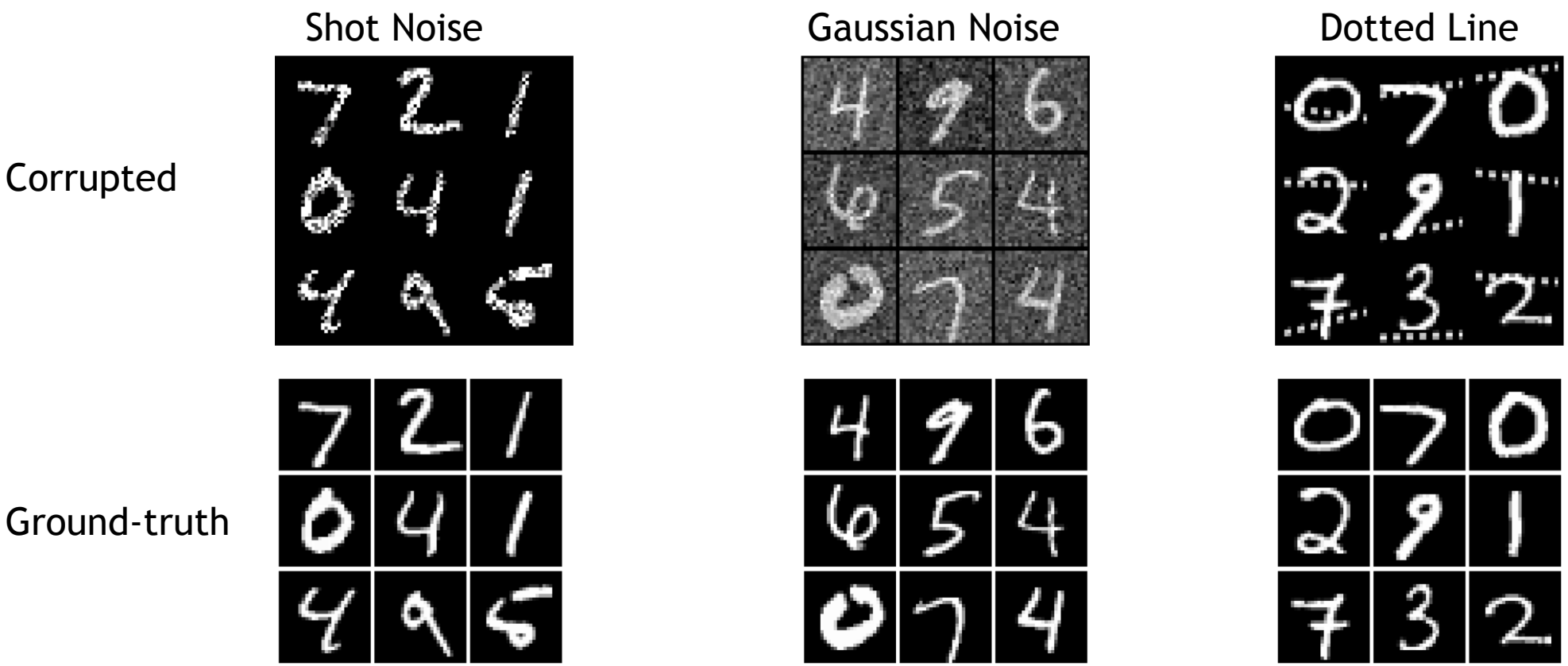
VGG/Allconv/...



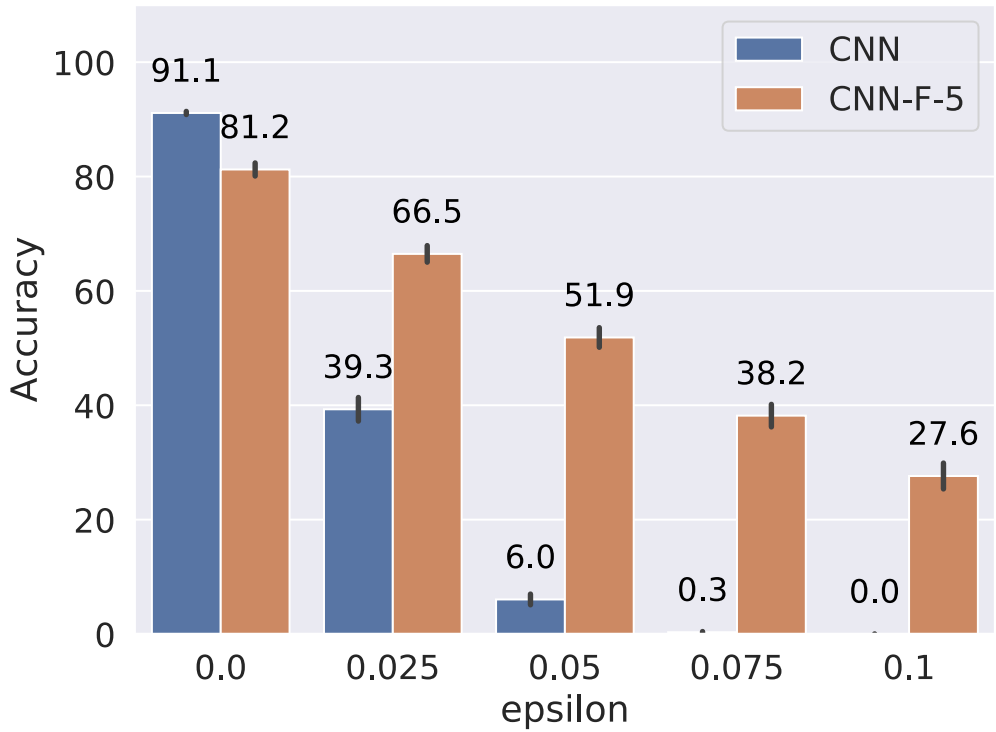
ResNet

IN: Instance Normalization

CNN-F repairs distorted images

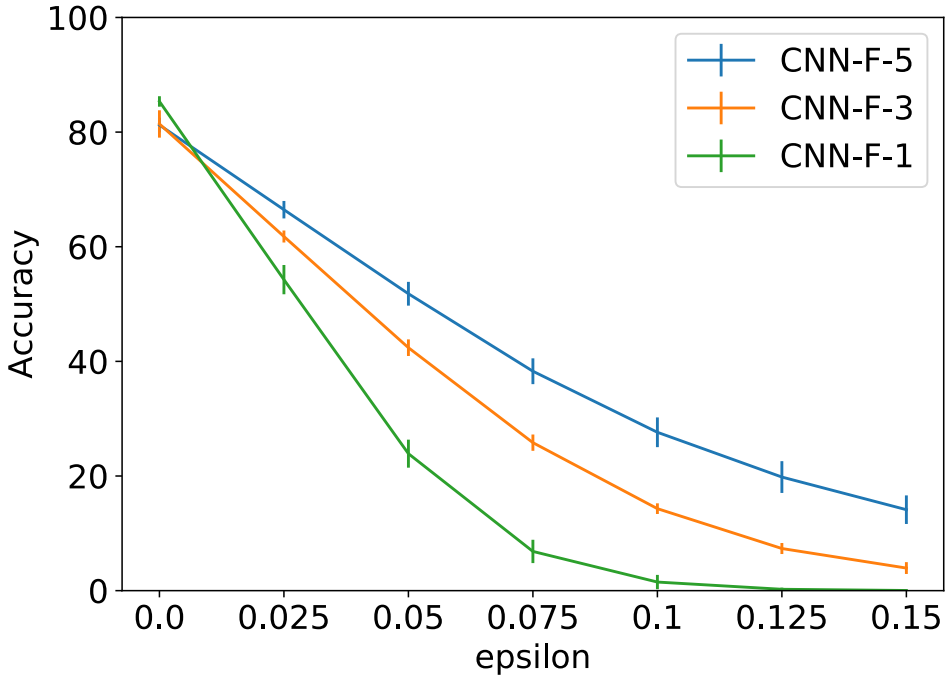


CNN-F improves adversarial robustness

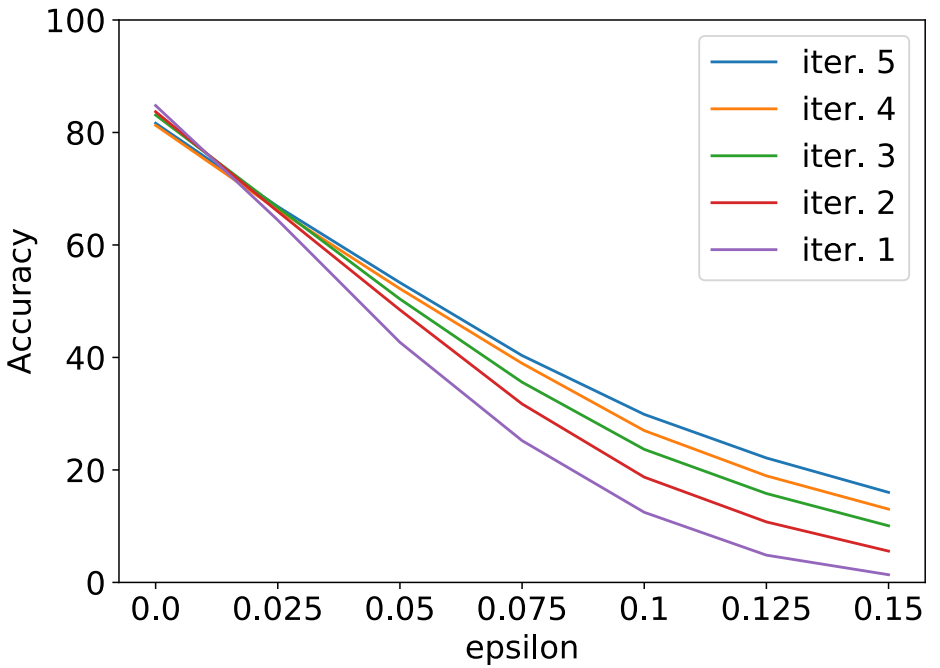


- Standard training on Fashion-MNIST.
- Attack with PGD-40.
- CNN-F has higher adversarial robustness than CNN.

CNN-F improves adversarial robustness



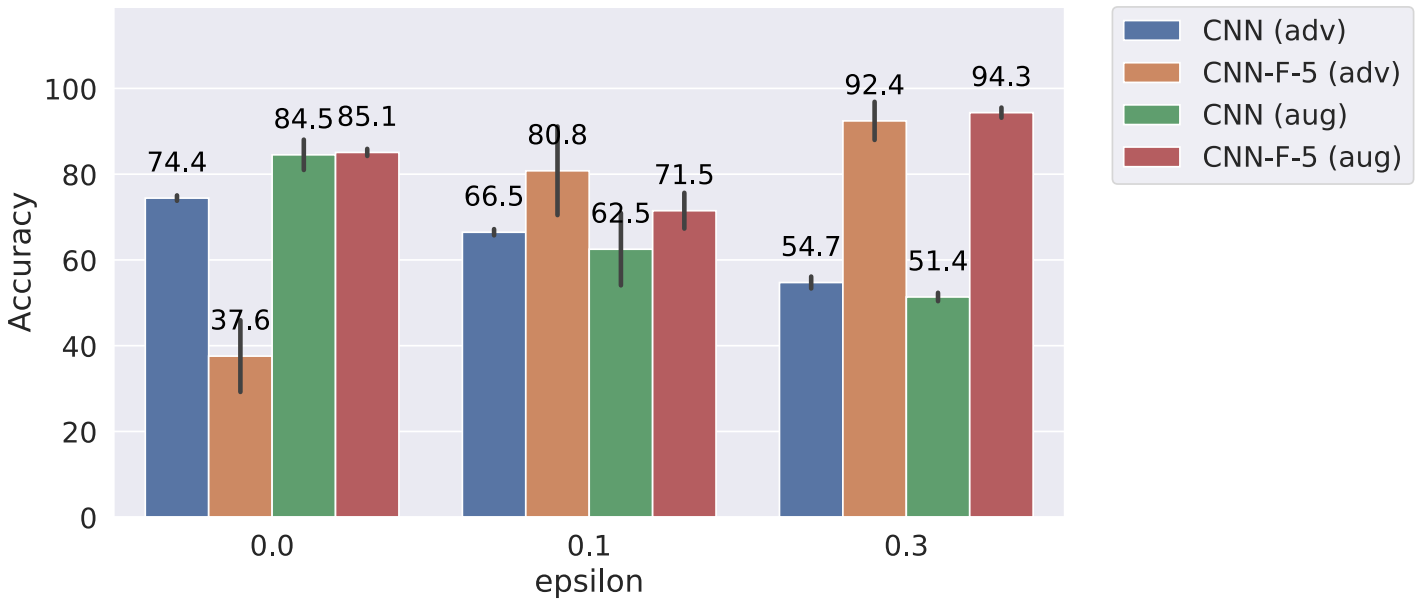
CNN-F trained with different iterations.



CNN-F tested with different iterations.

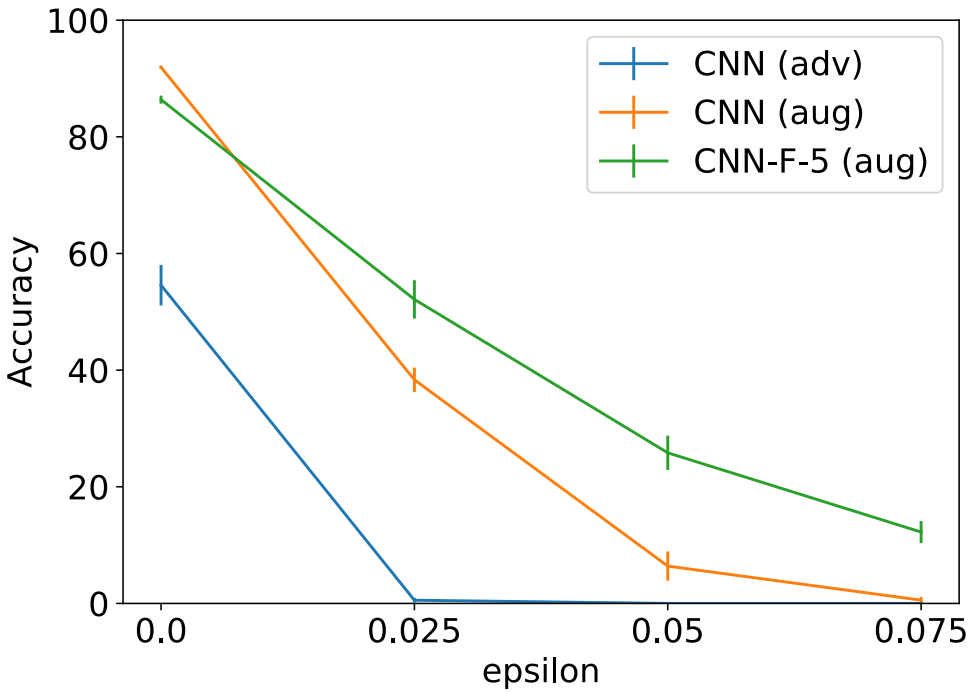
More iterations are needed for *harder* images.

CNN-F combined with adversarial training



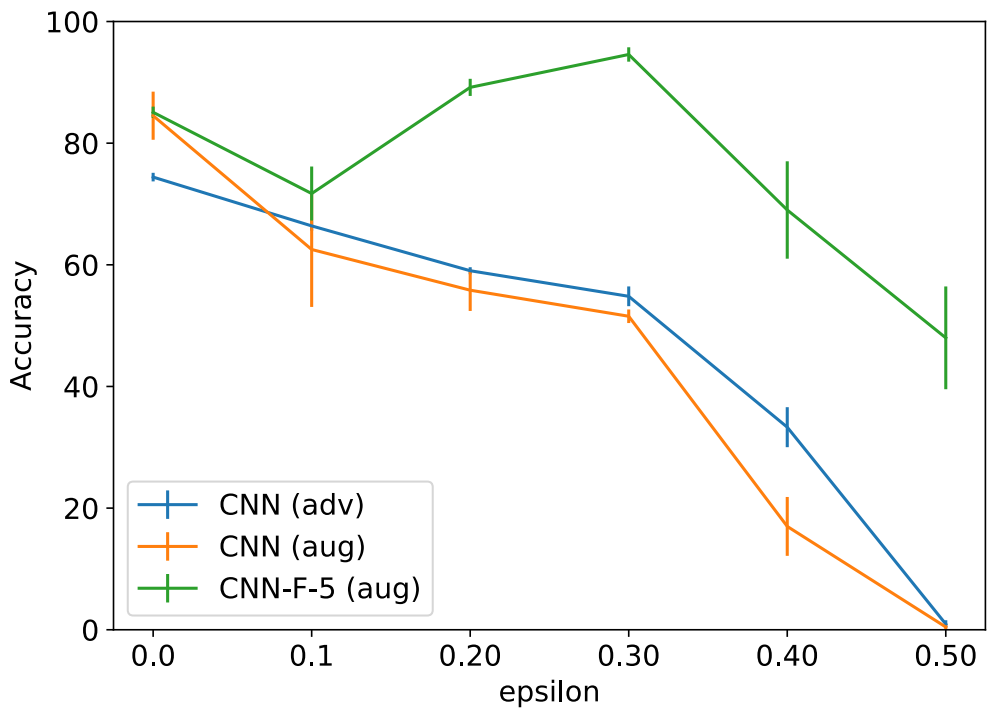
- Adversarial training on Fashion-MNIST.
- Trained with PGD-40 (eps=0.3). Attack with PGD-40.
- CNN-F augmented with adversarial images achieves high accuracy for both clean and adversarial data.

CNN-F generalizes better to different attacks



Trained with FGSM (eps=0.3).

Attack with PGD-40.



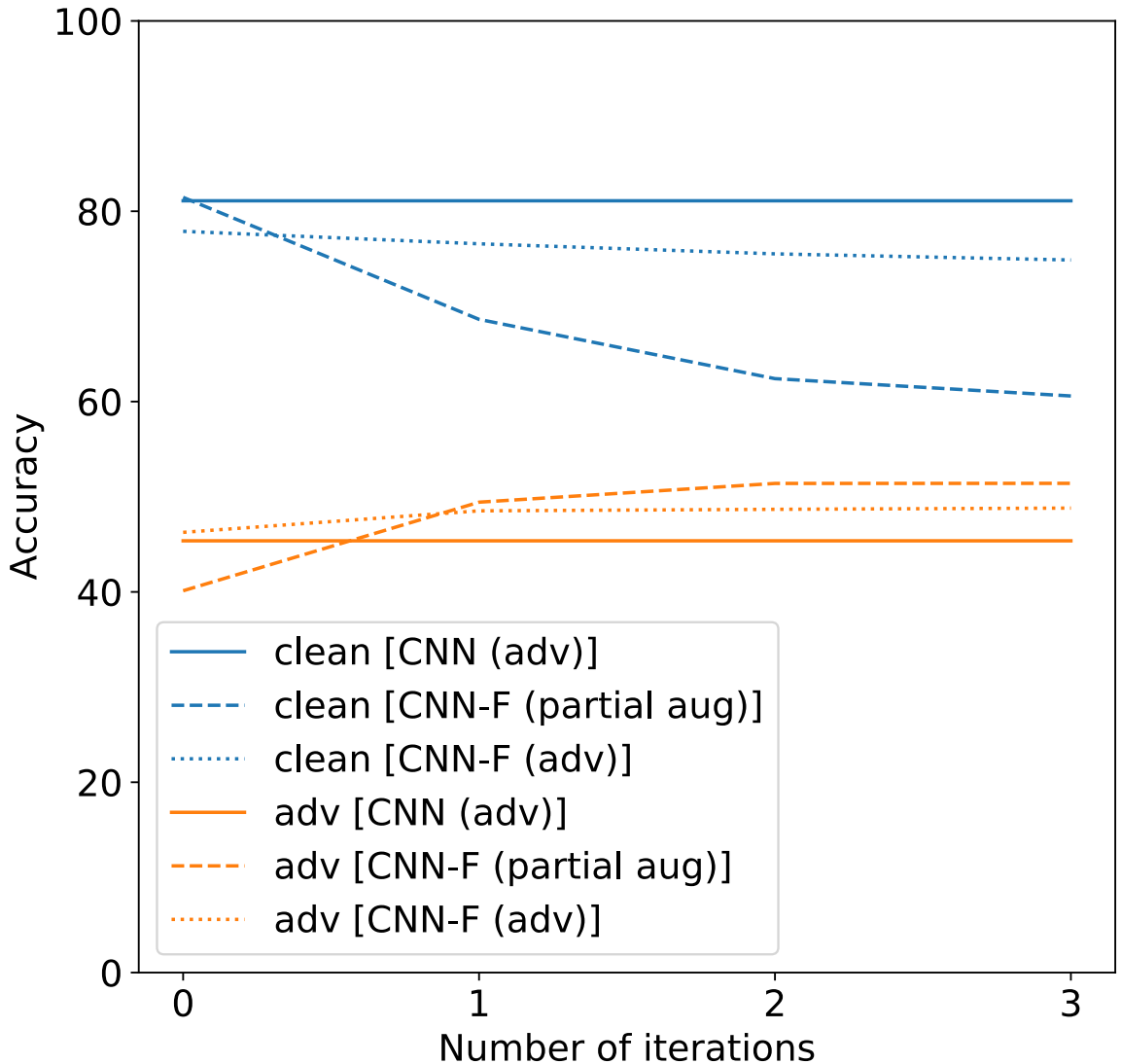
Trained with PGD-40 (eps=0.3).

Attack with PGD-40.

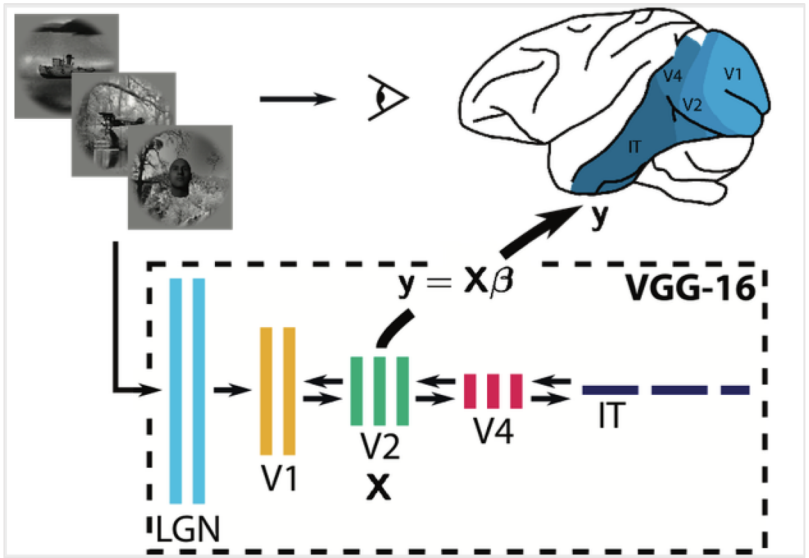
Feedback helps when there is distribution shift between training and testing data.

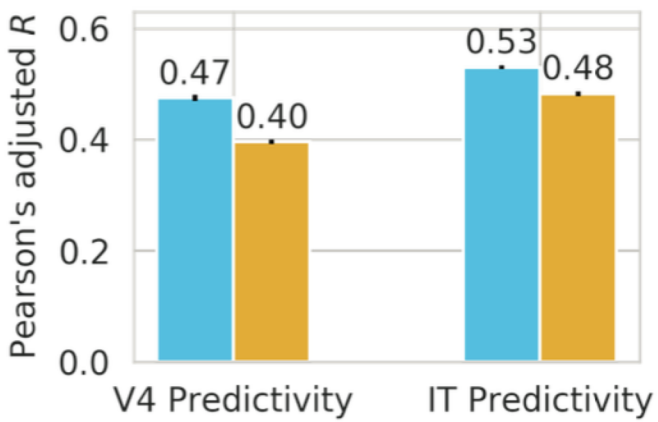
Train on CIFAR-10

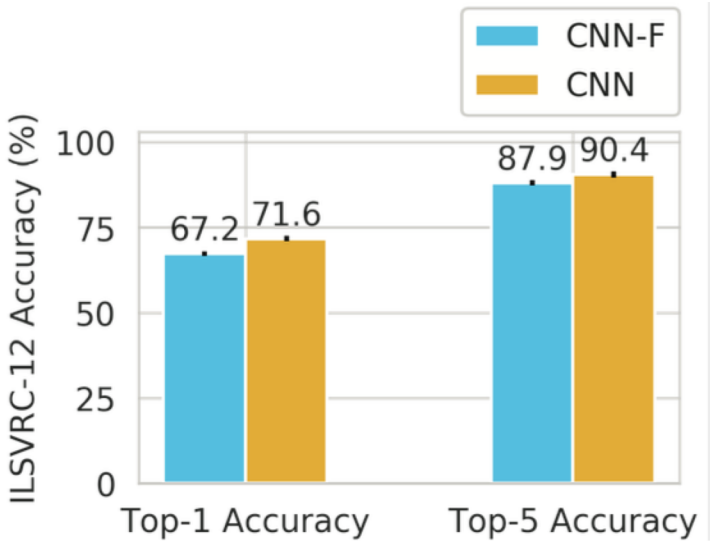
- CNN-F (on Wide ResNet) combined with adversarial training.
- Clean accuracy decreases over iterations.
- Adversarial accuracy increases over iterations.



Neuronal predictivity







- Used the fifth block and logits in VGG-16 to predict V4 and IT neuronal activities.
- CNN-F predicts V4 and IT neuronal activities better than CNN.
- Call for temporal neuronal data in the neuroscience community.

Conclusions and future works

Biological inspirations

- Recurrent feedback
    - Generative models (Bayesian brain)
    - Attention
- Lateral connections
- Sparsity

Inspirations from other fields

- Signal processing (Kalman filters …)
- Control (Feedback control, dynamical system)

Down-streaming tasks

- Robustness
- Few shot learning
- Uncertainty quantification

Thank You!