Revisiting Ensembles in an Adversarial Context: Improving Natural Accuracy

Aditya Saligrama (MIT PRIMES) and Guillaume Leclerc (MIT)

ICLR 2020 Workshop on Towards Trustworthy ML: Rethinking Security and Privacy for ML

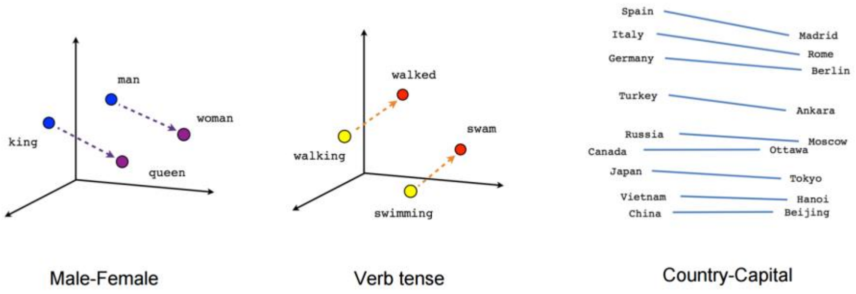
April 26, 2020

Deep learning and adversarial examples

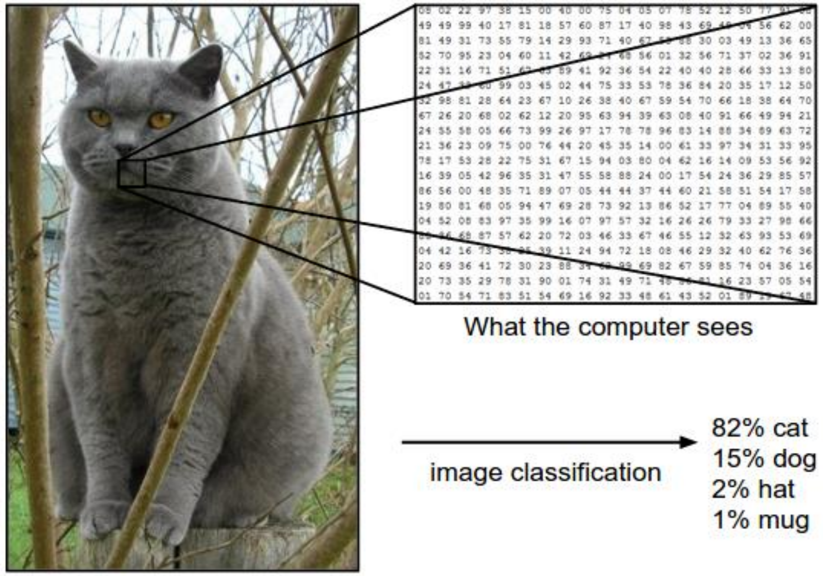
Deep learning

- Has become ubiquitous in the last few years and can outperform humans on some tasks

(DeepAI 2019)



(Ruizendaal 2017)

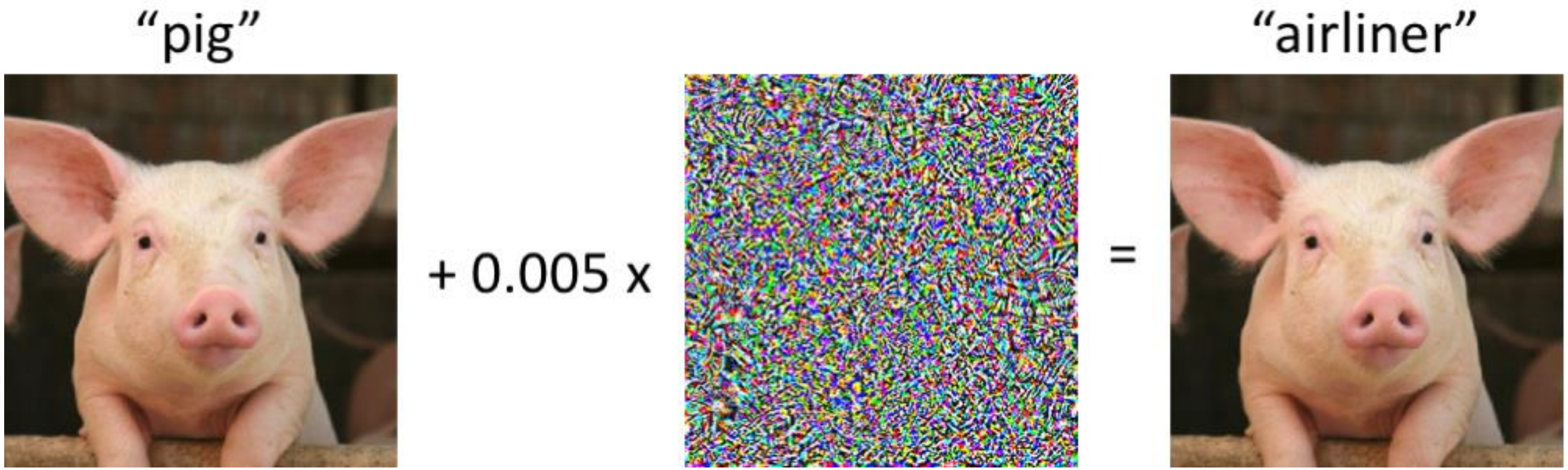


(Karpathy 2015)

Adversarial attacks

- Modify image in a set S, such as L2-ball of size ε, to maximize loss L
  - Imperceptible to human observer
  - Fools deep learning models

$$\hat{\delta} = \underset{||\delta||<\epsilon}{\operatorname{argmax}} L(\theta, x + \delta, y)$$

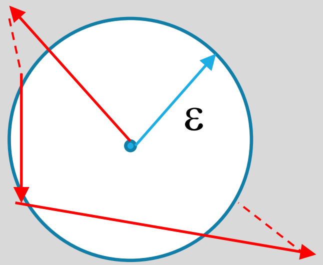


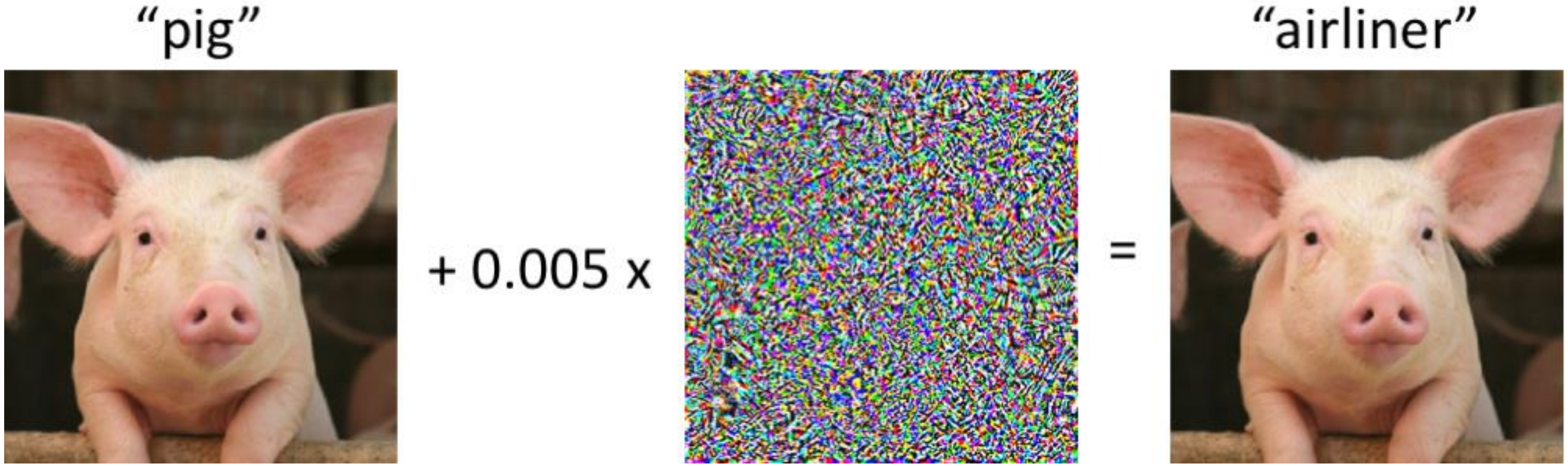
(Mądry and Schmidt 2018)

Adversarial attacks

- Modify image in a set S, such as L2-ball of size ε, to maximize loss L
  - Imperceptible to human observer
  - Fools deep learning models
- Many ways of synthesizing adversarial examples:
  - Such as PGD - projected gradient descent (Mądry et al. 2017)

$$\hat{\delta} = \underset{||\delta||<\epsilon}{\operatorname{argmax}} L(\theta, x+\delta, y)$$





(Mądry and Schmidt 2018)

Robust training

- Train robust model θ on dataset *D*:
  - Resistant to adversarial attacks
  - Robust training via PGD (Mądry et al. 2017)
    - Many other ways…

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}\left[\max_{||\delta||\leq\epsilon} L(\theta, x+\delta, y)\right]$$

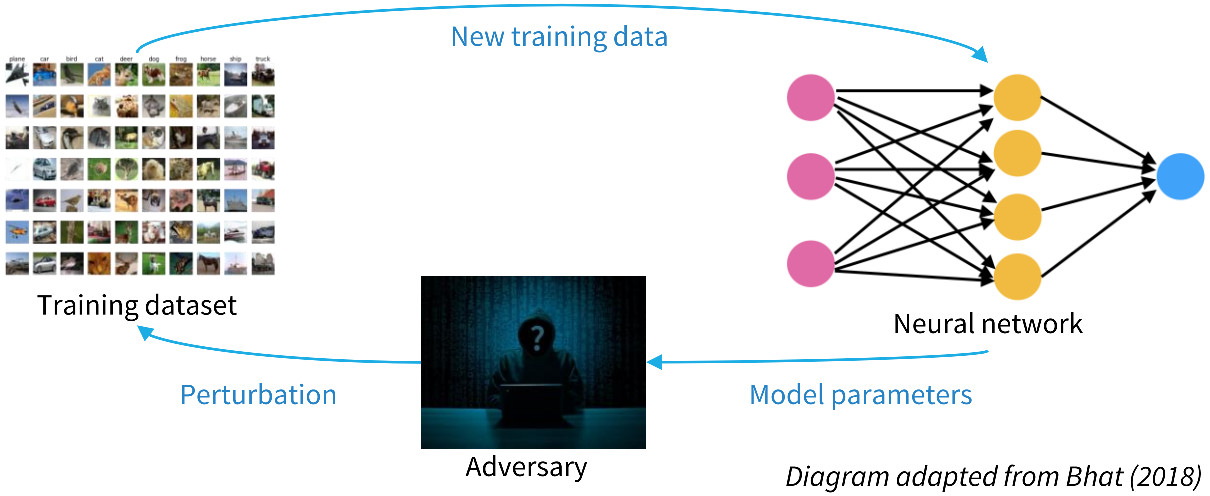


Diagram adapted from Bhat (2018)

Robust training

- Train robust model θ on dataset *D*:
  - Resistant to adversarial attacks
  - Robust training via PGD (Mądry et al. 2017)
    - Many other ways…

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}\left[\max_{||\delta||\leq\epsilon} L(\theta, x+\delta, y)\right]$$

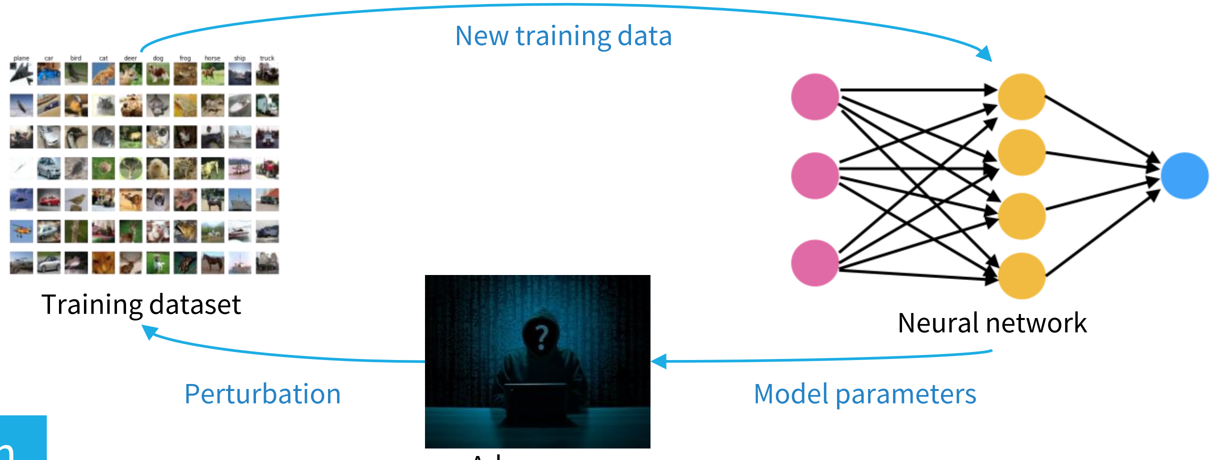


Diagram adapted from Bhat (2018)

ResNet18 models (He et al. 2015)
trained on CIFAR10

	Natural train	Robust train (ε=0.5)
Natural test		
Adv. test (ε=0.5)		

Robust training

- Train robust model θ on dataset *D*:
  - Resistant to adversarial attacks
  - Robust training via PGD (Mądry et al. 2017)
    - Many other ways…

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}\left[\max_{||\delta||\leq\epsilon} L(\theta, x+\delta, y)\right]$$

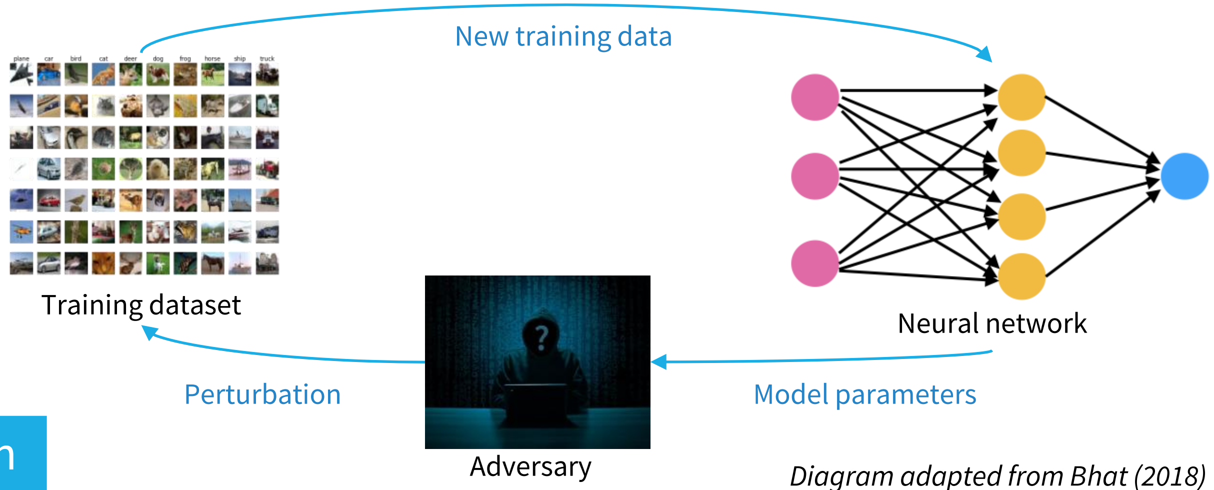


Diagram adapted from Bhat (2018)

ResNet18 models (He et al. 2015)
trained on CIFAR10

	Natural train	Robust train (ε=0.5)
Natural test	95%	88%
Adv. test (ε=0.5)	0%	69%

Metrics

- Assess resistance to adversarial attacks at multiple attack strengths
  - Adversary can choose any arbitrary attack strength against deployed model
- We define AUC metric as

$$AUC(\epsilon_{target}) = \frac{1}{\epsilon_{target}} \int_0^{\epsilon_{target}} A(\epsilon) d\epsilon.$$

- In practice, evaluate as a Riemann sum
- Use this metric in addition to assessing accuracy at defined attack strengths

Ensembling schemes

Adversarial ensembling

Using ensembling for training (lots of prior work, different from previous slide):

- Vanilla ensembling (baseline for this talk)
  - Random initializations, train M standard models
- Ensemble Adversarial Training (Tramèr et al. 2017)
  - Collect adversarial examples from multiple models
  - Transfer examples to train single model
- Ensemble diversity (Pang et al. 2019)
  - Coupled training of all *M* models to promote diversity

	Robust training (Mądry et al. 2017)	Vanilla ensembling	Ensemble diversity (Pang et al. 2019)
Natural test	88%	94%	93%
Adv. test	69% (ε=0.5)	0%	30% (ε=0.02)

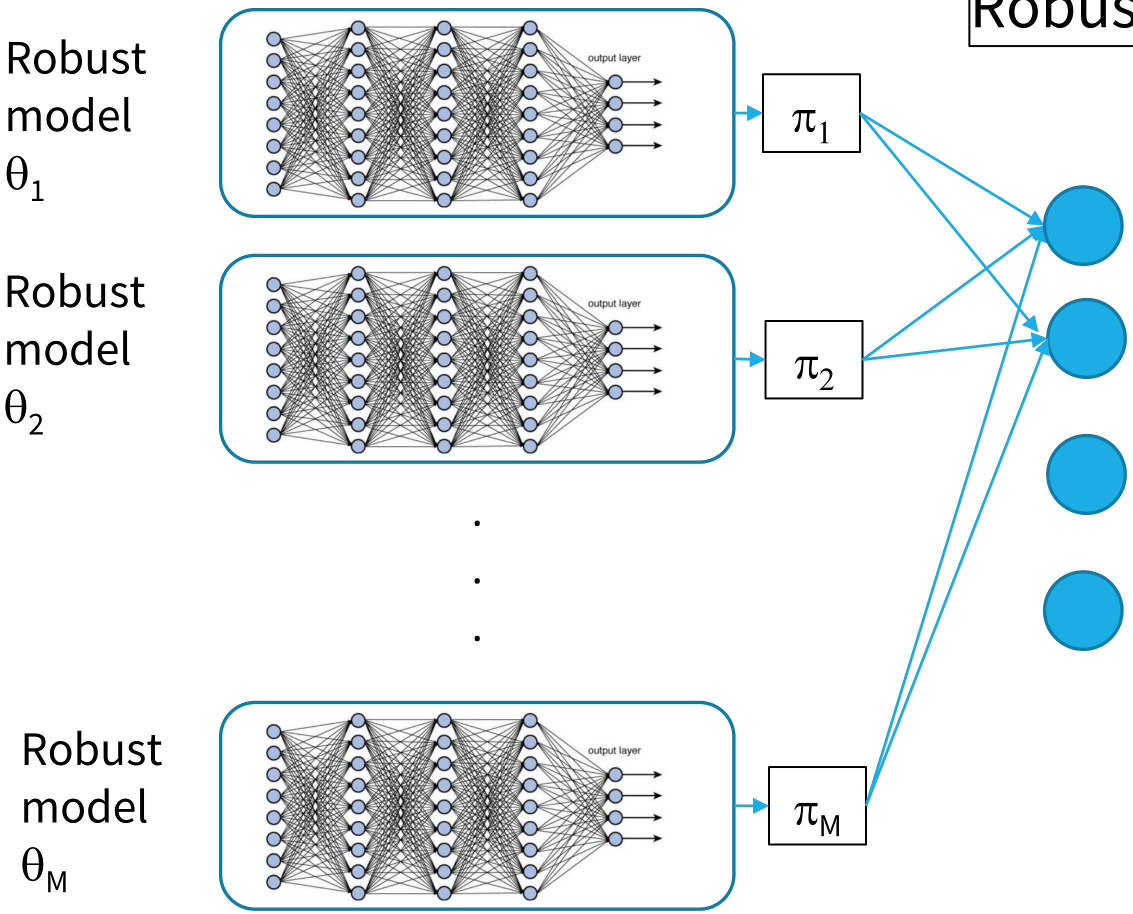
Our proposed methods

Robust ensembling

- Train M independent models robustly
  - *i*'th model with seed *i*

$$\hat{\theta}_{\mathrm{i}} = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}\left[\max_{||\delta||\leq\epsilon} L(\theta, x+\delta, y)\right]$$

Robust training with initialization seed i



$$c(x, \boldsymbol{\theta}, \boldsymbol{\pi}) = \max_y \sum_{i=1}^{M} \pi_i \theta_i(x, y)$$

$\theta_i(x, y)$: model *i*'s probability of class *y* on instance *x*

How to understand ensembles?

Value of the game (discrete):

- Player: random strategy over M models
  - Probability $\pi_1 \ldots \pi_M$
- Adversary: perturbation $\delta_1 \ldots \delta_S$ ($S \to \infty$) with probability $q_1 \ldots q_S$

$$\ell(\mathbf{q}, \pi, L) = E_{\delta \sim \mathbf{q}} E_{\theta_j \sim \pi} L(\theta_j, x + \delta, y)$$

Player strategy →

Adversary strategy ↓

	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$
δ_1	Loss		
δ_2			
δ_3			

Key point: Adversary plays against ensemble rather than single model for each instance

$$\min_\pi \max_{\mathbf{q}} \ell(\mathbf{q}, \pi, L) \leq \max_\delta \frac{1}{M} \sum_j L(\theta_j, x + \delta, y)$$

vs.

$$\max_{\delta \in S} L(\theta, x + \delta, y)$$

How to understand ensembles?

Value of the game (discrete):

- Player: random strategy over M models
    - Probability $\pi_1 \ldots \pi_M$
- Adversary: perturbation $\delta_1 \ldots \delta_S$ ($S \rightarrow \infty$) with probability $q_1 \ldots q_S$

$$\ell(\mathbf{q}, \pi, L) = E_{\delta \sim \mathbf{q}} E_{\theta_j \sim \pi} L(\theta_j, x + \delta, y)$$

Player strategy →

Adversary strategy ↓

	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$
δ_1	Loss		
δ_2			
δ_3			

Key point: Adversary plays against ensemble rather than single model for each instance

$$\min_\pi \max_{\mathbf{q}} \ell(\mathbf{q}, \pi, L) \leq \max_\delta \frac{1}{M} \sum_j L(\theta_j, x + \delta, y)$$

vs.

$$\max_{\delta \in S} L(\theta, x + \delta, y)$$

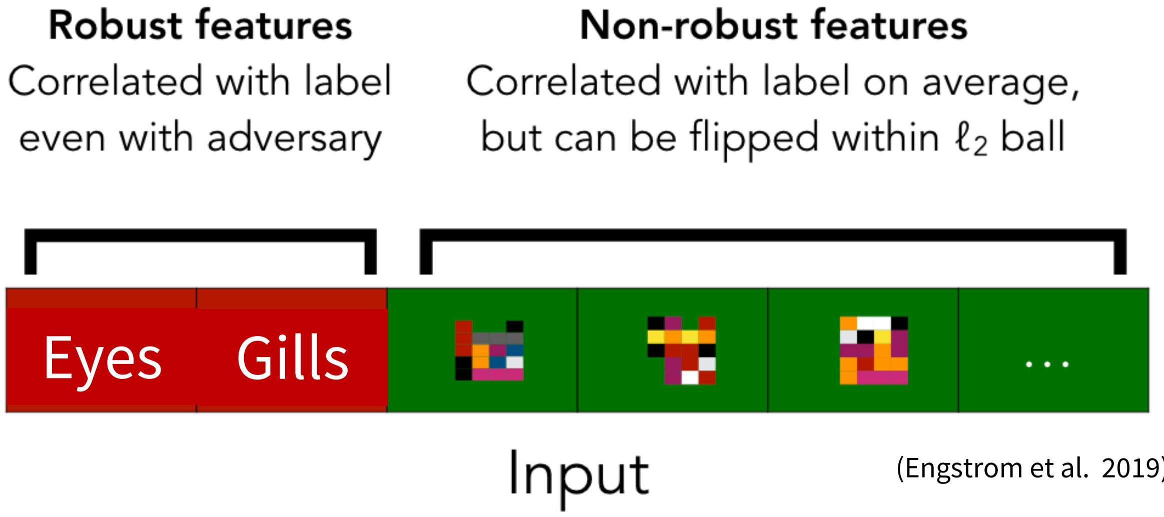
robust ensemble loss ≤ single robust model loss
Why? Choose **q** to focus on single model

This allows accuracy to increase per model in the ensemble for a given ε

Robust and non-robust features

- Images comprised of robust and non-robust features (Ilyas et al. 2019)
- **Key insight: Robust features do not have enough info about particular instances**
  - **Non-robust features contain remaining info**

Robust features



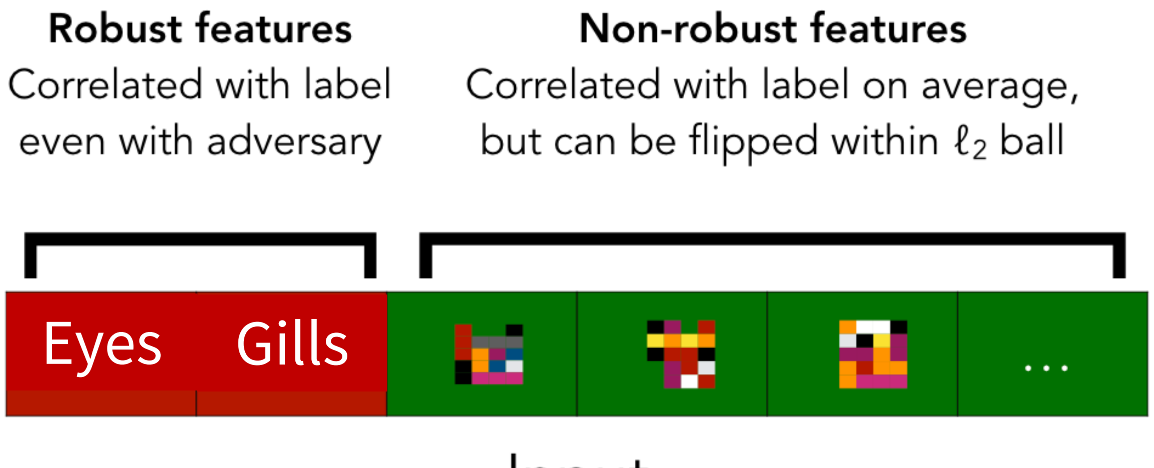
(Engstrom et al. 2019)

Robust + non-robust features

Robust and non-robust features

- Images comprised of robust and non-robust features (Ilyas et al. 2019)
  - Training at lower ε means less resistance to non-robust features and better natural accuracy
- **Key insight 1: Lower train ε confers better natural accuracy at the cost of robustness**
  - **Objective: Combine with ensembling to maintain robustness with better natural accuracy**
- **Key insight 2: Robust features do not have enough info about particular instances**
  - **Non-robust features contain remaining info**
  - **Objective: Augment non-robust features with robust features without losing robustness**

Robust features



(Engstrom et al. 2019)

Robust + non-robust features

Robust ensembling: Results

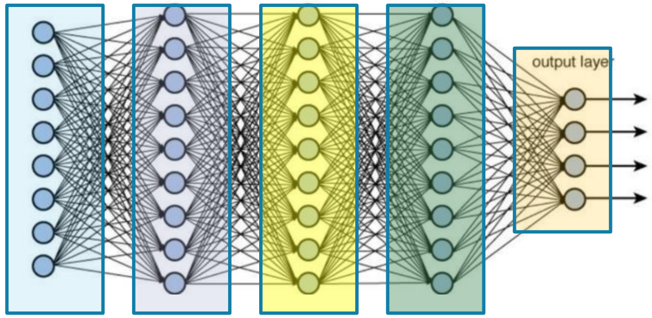
Number of models (train $\varepsilon = 0.5$)	Natural accuracy	Adversarial accuracy ($\varepsilon = 0.5$)
1	88.30%	68.73%
2	88.92%	71.19%
4	89.07%	72.53%
8	89.36%	73.08%
12	89.28%	73.34%
16	89.18%	73.37%

	Single non-robust model	Single robust model (train $\varepsilon = 0.5$)	Robust ensemble (8 models, train $\varepsilon = 0.22$)
Natural test	94.6%	88.3%	94.0%
Adv. Test ($\varepsilon = 0.5$, $k = 7$)	0.4%	68.7%	68.8%
AUC(0.5) w/4 increments	0.067	0.767	0.781

Composite ensembling

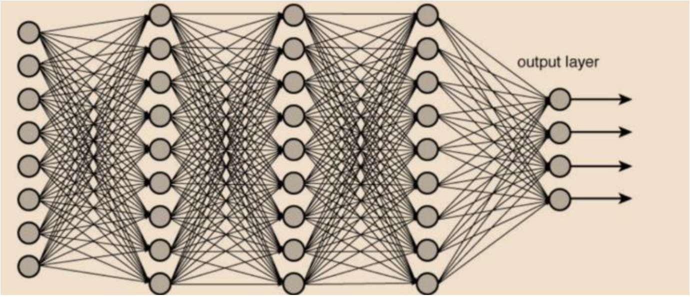
Robust

Strong



Robust

Weak

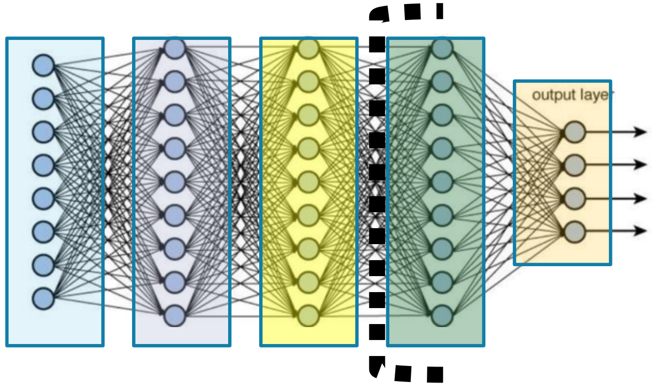


Composite ensembling

Extract Last Layers

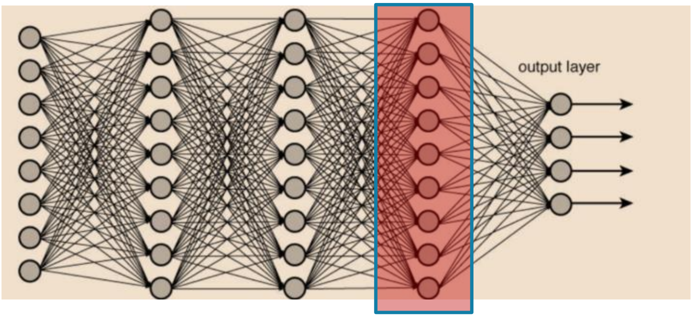
Robust

Strong



Robust

Weak

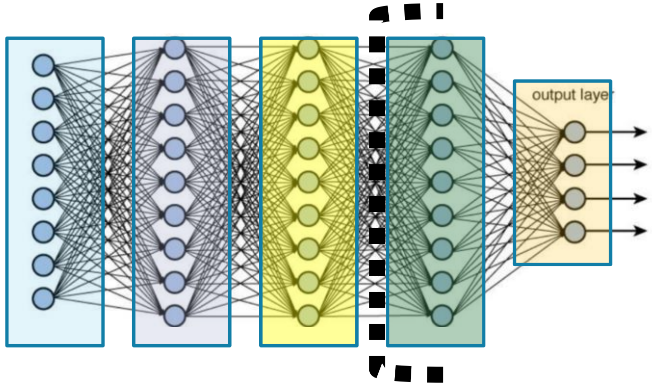


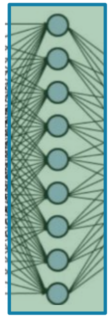
Composite ensembling

Replicate Last Robust Layer

Robust

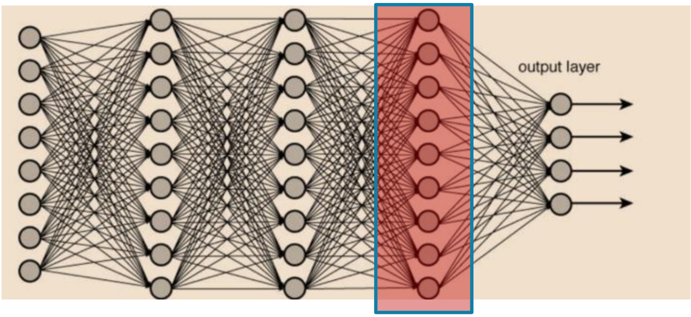
Strong

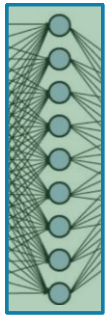




Robust

Weak



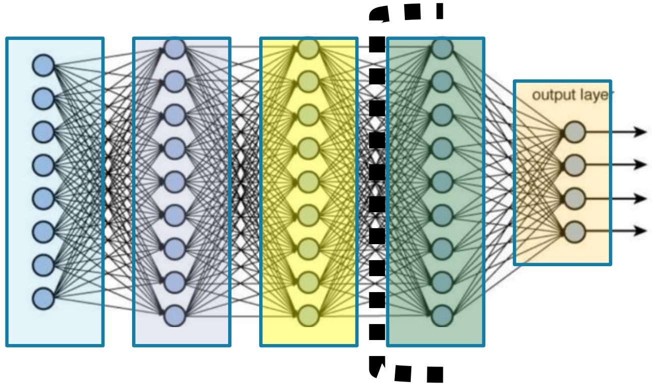


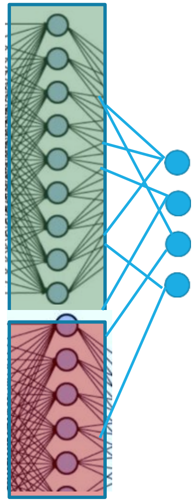
Composite ensembling

Replicate Last Robust Layer + Attach Natural Last Layer + Train Last Composite Layer Independently

Robust

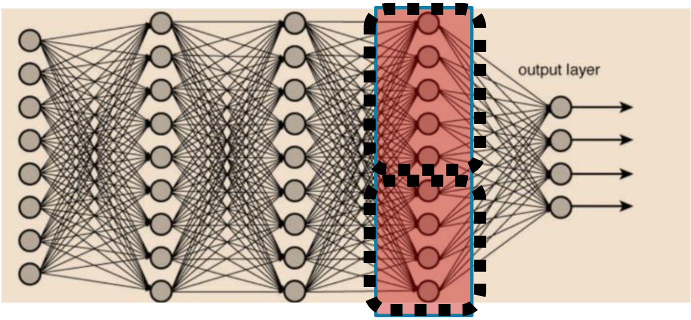
Strong

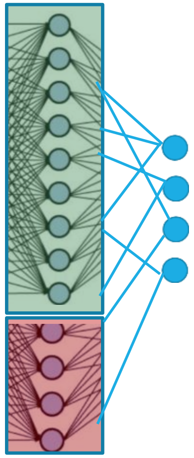




Robust

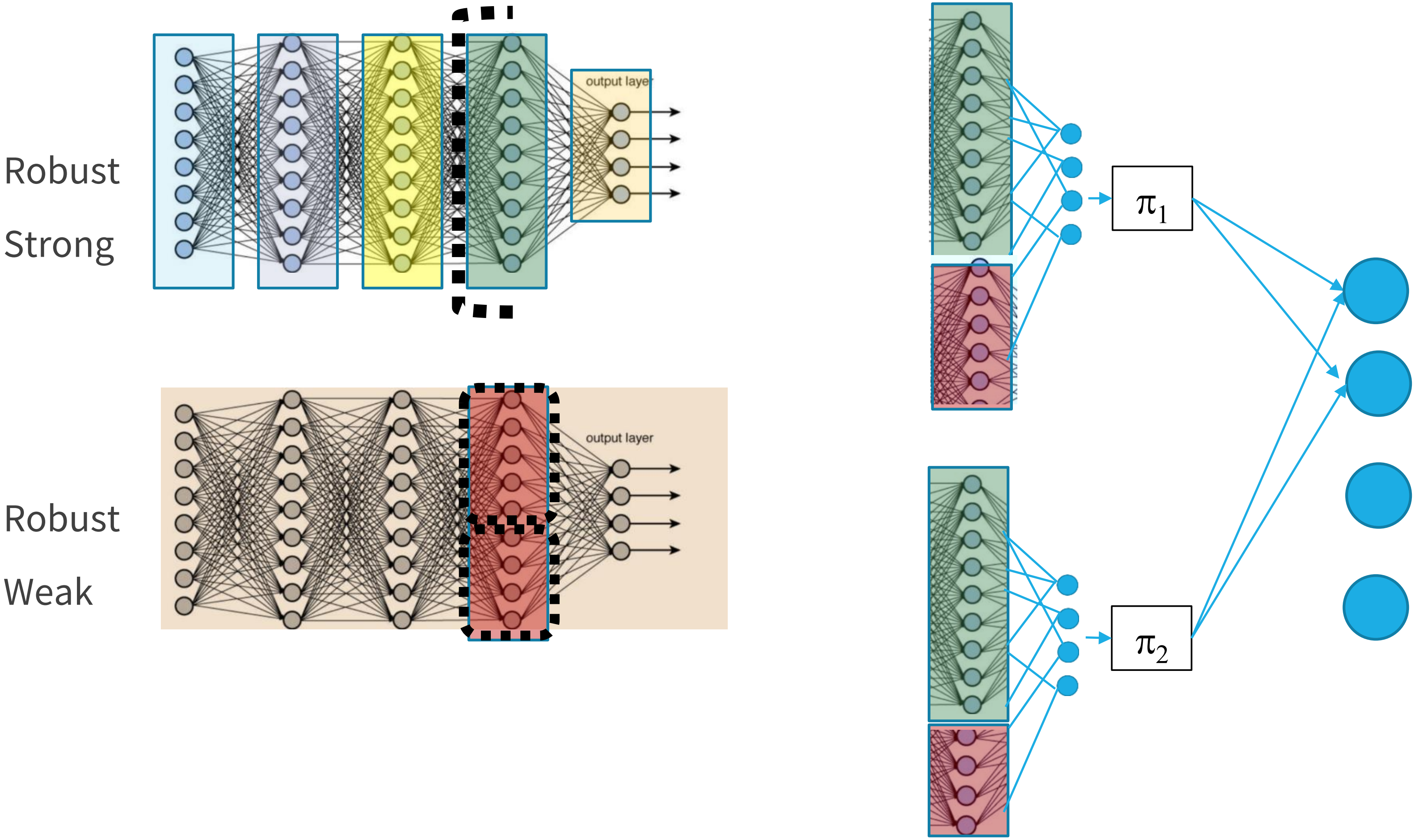
Weak





Composite ensembling?

Replicate Last Robust Layer + Attach Natural Last Layer + Train Last Composite Layer Independently



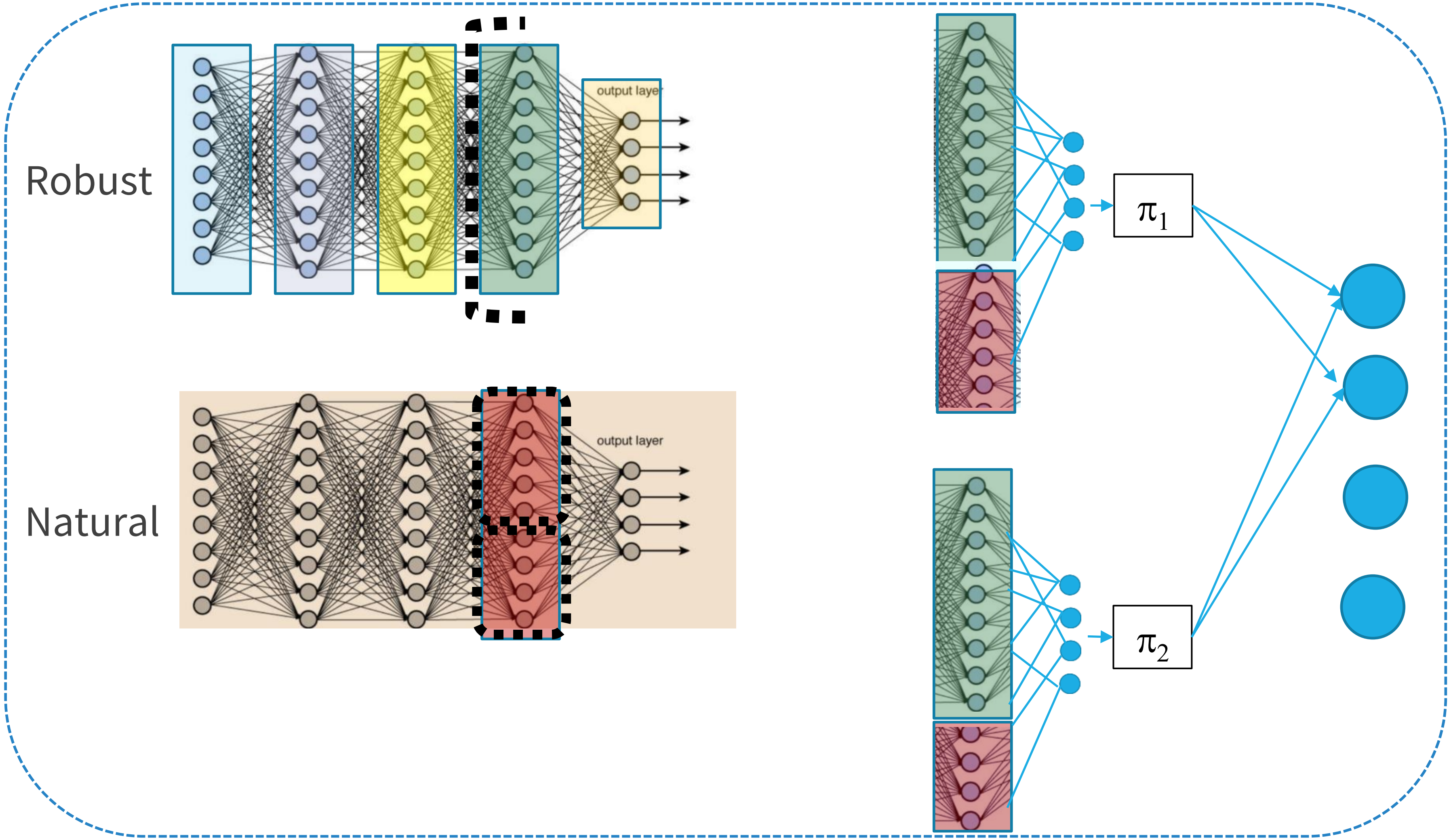
Composite prediction = ensemble weighted average

Composite acc. ≥ single robust model acc.

Composite ensembling: Results

	Single non-robust model	Single robust model (train ε = 0.5)	Robust ensemble (8 models, train ε = 0.22)	1-composite (train ε = 0.4, 0.05 trained at ε = 0.4)
Natural test	94.6%	88.3%	94.0%	91.4%
Adv. Test (ε = 0.5, k = 7)	0.4%	68.7%	68.8%	68.0%
AUC(0.5) w/4 increments	0.067	0.767	0.781	0.769

Meta-composite ensembling

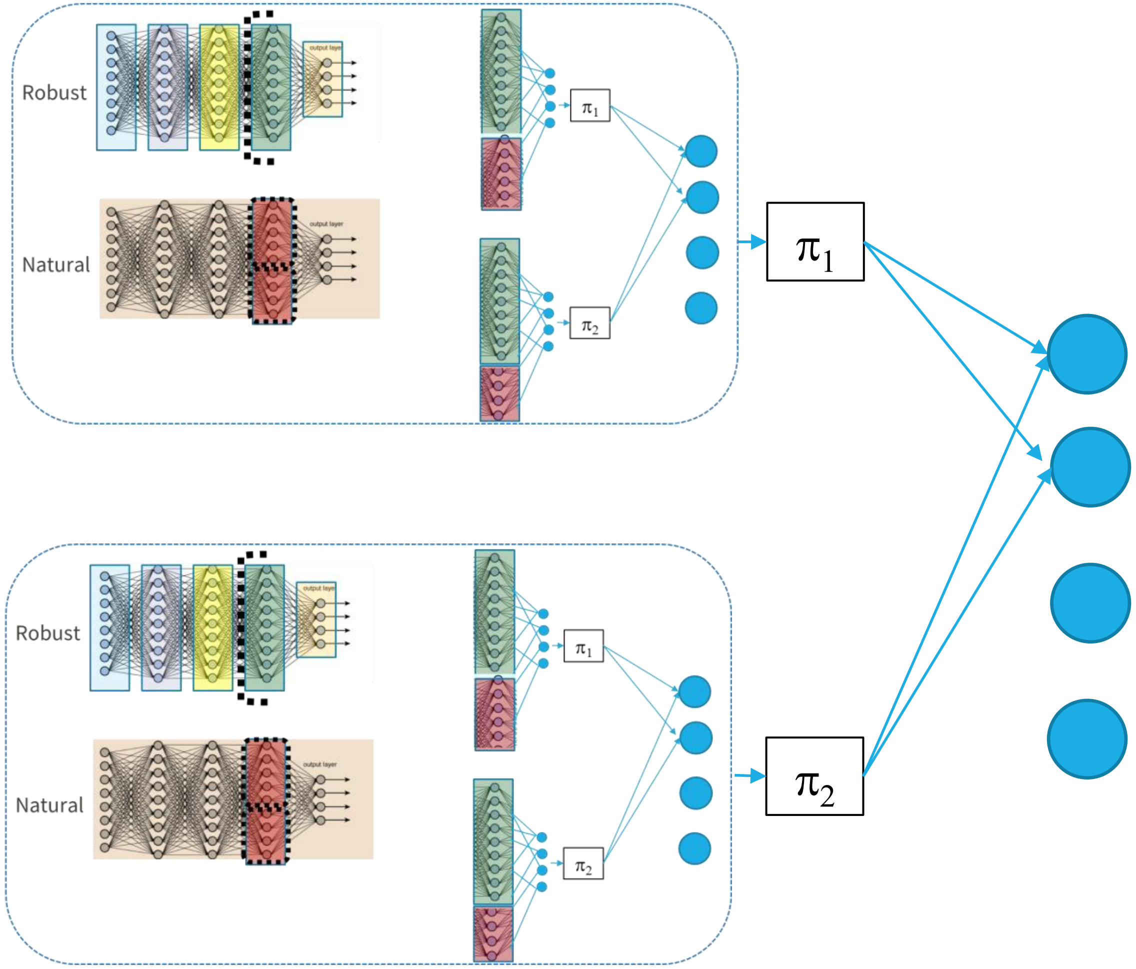


Take many composite ensembles

Meta-composite ensembling

Composite 1

Composite 2



Meta Composite prediction = ensemble of composites

Meta-composite ensembling

- Combine *M* independently trained composite models

	Single non-robust model	Single robust model (train $\varepsilon = 0.5$)	Robust ensemble (8 models, train $\varepsilon = 0.22$)	1-composite (train $\varepsilon = 0.4$, 0.05 trained at $\varepsilon = 0.4$)	2x 1-composite Weighted average
Natural test	94.6%	88.3%	94.0%	91.4%	91.6%
Adv. Test ($\varepsilon = 0.5$, $k = 7$)	0.4%	68.7%	68.8%	68.0%	70.0%
AUC(0.5) w/4 increments	0.067	0.767	0.781	0.769	0.783

Key insights and Conclusions

- AUC metric to evaluate robustness of models
  - Allows us to assess robustness at multiple attack strengths
- Robust ensembling outperforms single models
  - Choosing models randomly forces adversary to use average strategy
  - Different models may mispredict the same way, but require different perturbations
  - Allows us to decrease train ε, therefore increasing natural accuracy at a given level of robustness
- Proposed composite and meta-composite models
  - Re-incorporate non-robust features
  - Improves on AUC metric compared to single models while using less models than robust ensembling

Future work

- Validation with other adversarial attacks such as Carlini-Wagner (Carlini and Wagner 2017)
- Use meta-composite framework to improve natural accuracy outside adversarial context

Acknowledgements

- Prof. Aleksander Mądry
- Logan Engstrom, Andrew Ilyas, and the rest of Mądry Lab
- Dr. Slava Gerovitch and Prof. Srini Devadas, for supporting the PRIMES program and this research
- Prof. Nicolas Papernot and workshop organizers