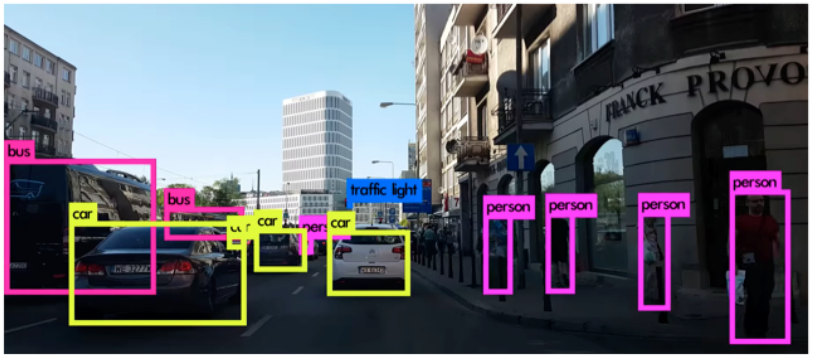
On **Robustness** and **Efficiency** of Machine Learning Systems

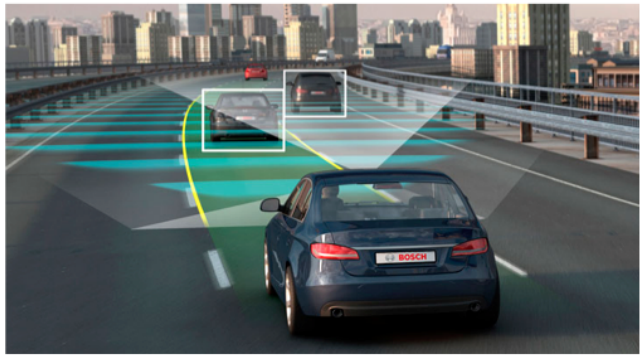
Yao Li
Department of Statistics and Operations Research
UNC Chapel Hill

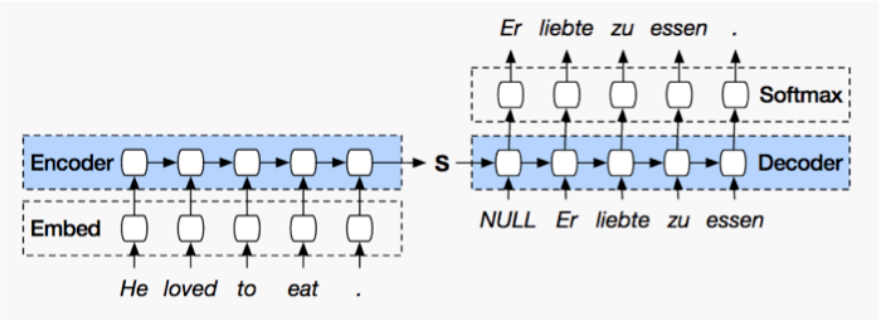
Thomas

Cho

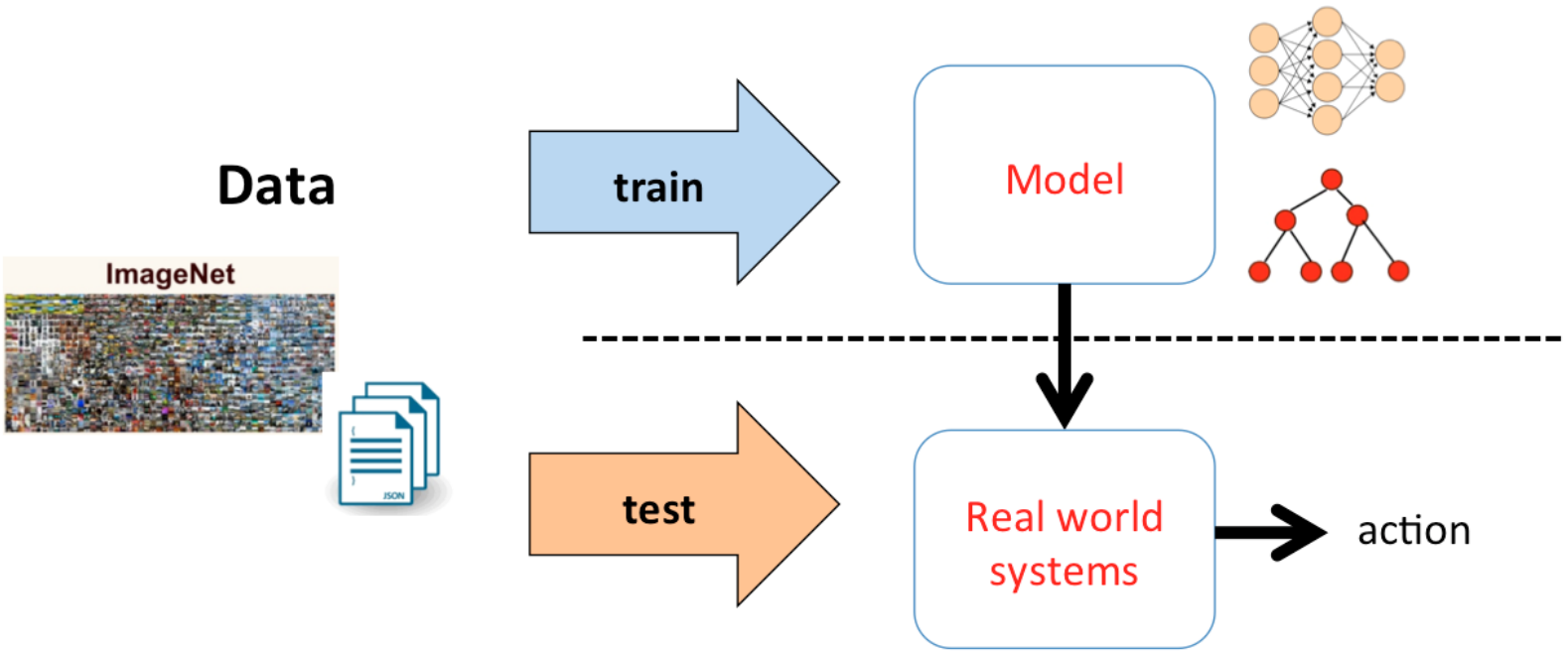
Machine Learning





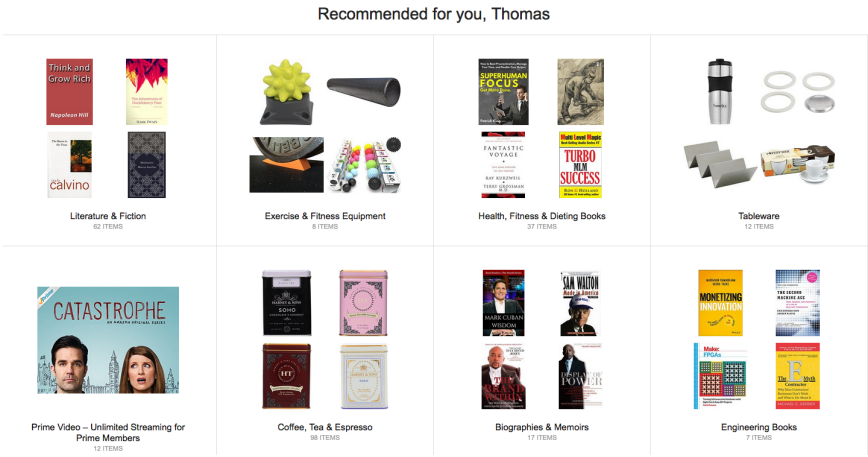


Machine Learning Systems



Training: Scalable and efficient algorithms

- Utilize more information
- Need for Scalabe algorithms

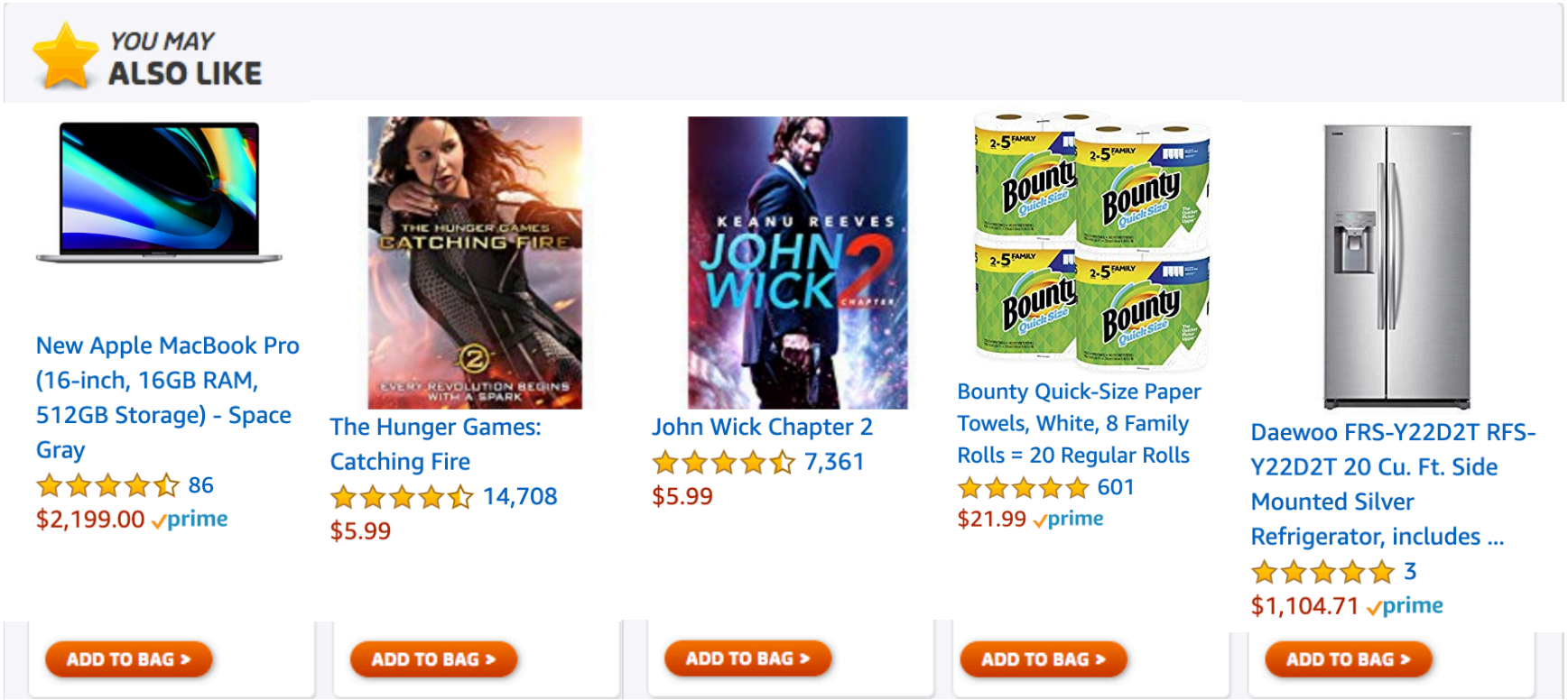


(Recommendation system for e-commerce)

(Game result prediction)

Scalable Demand-Aware Recommendation

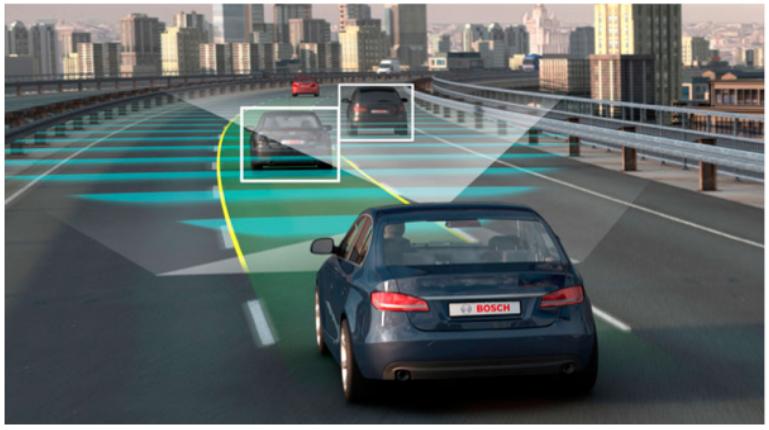
- Durable vs. Non-durable goods



- Form utility: the item is desired as it is manifested
- Time utility: the item is desired at the given point in time

Prediction: Robustness and Safety

ML systems need to interact with real world



- Robustness and Safety

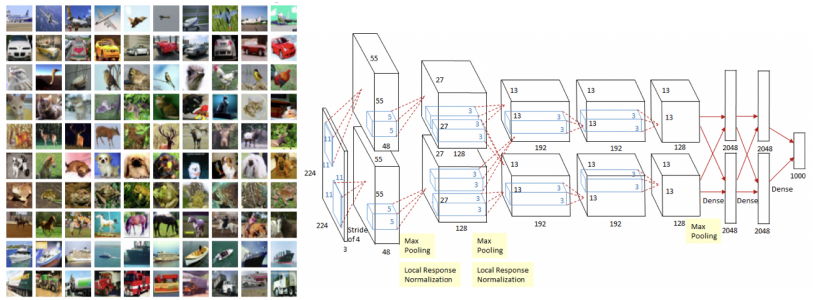
Today's Outline

1. Robustness and Safety of ML Systems
   - Background
   - Related Work
   - Proposed Defense Methods

2. Efficient Machine Learning Algorithms
   - Background
   - Previous Work
   - Proposed Method

Outline

Deep Neural Network



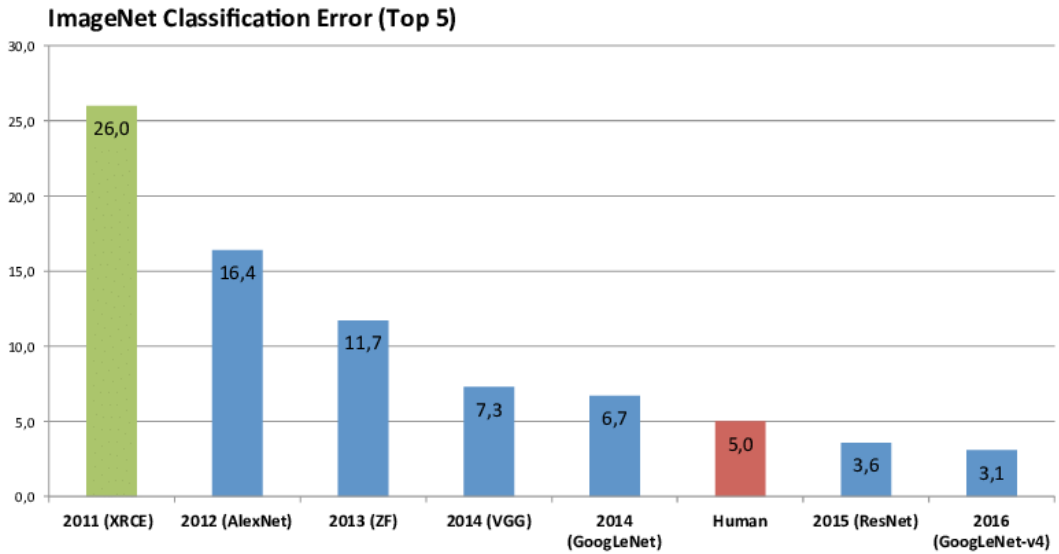
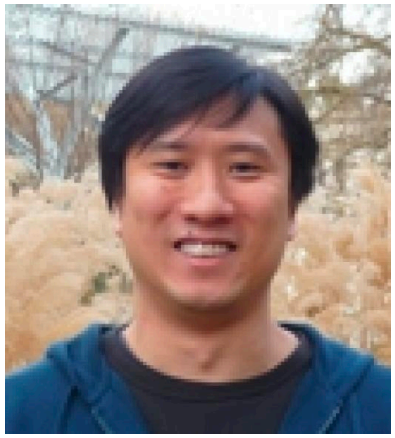
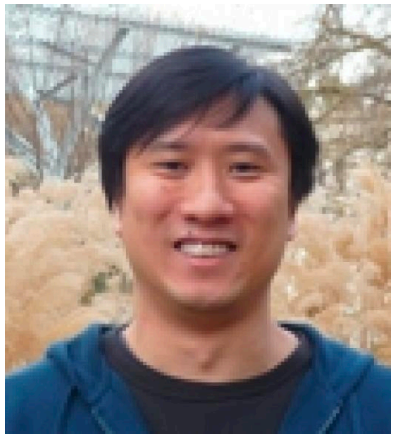


Figure: Winner results of the ImageNet large scale visual recognition challenge (LSVRC) of the past years on the top-5 classification task

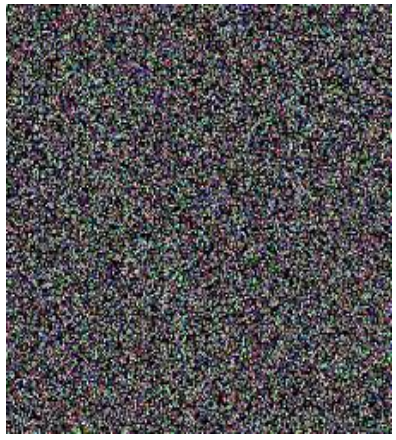
Adversarial Examples



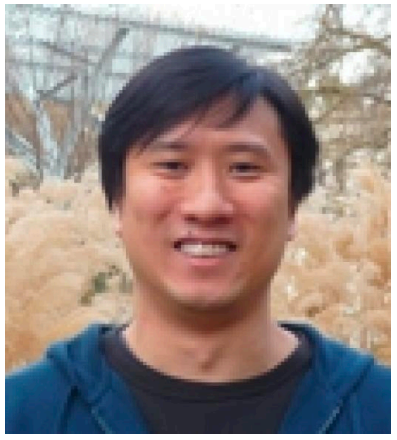
Adversarial Examples



$+0.0001\times$



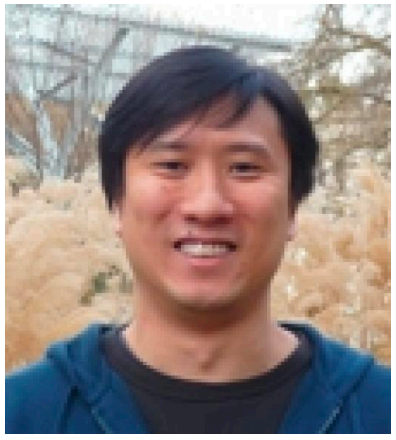
$=$



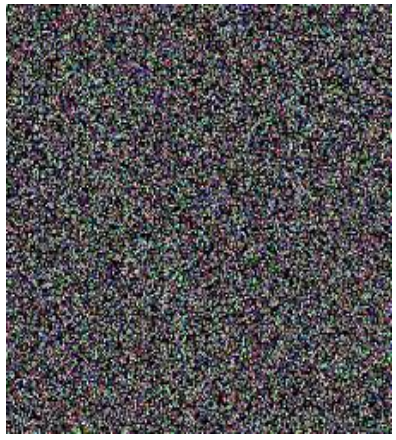
DNN

DNN

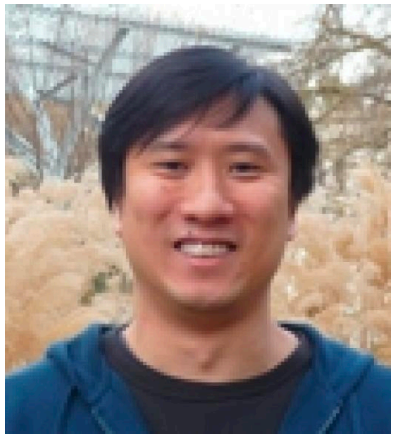
Adversarial Examples



$+0.0001\times$



$=$

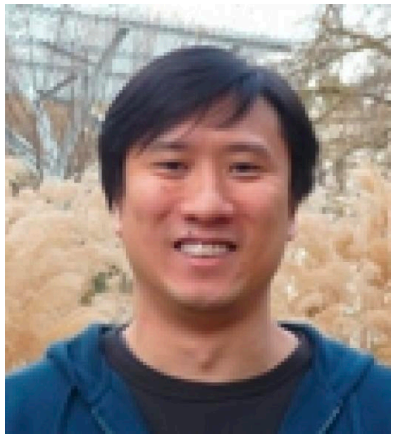


DNN

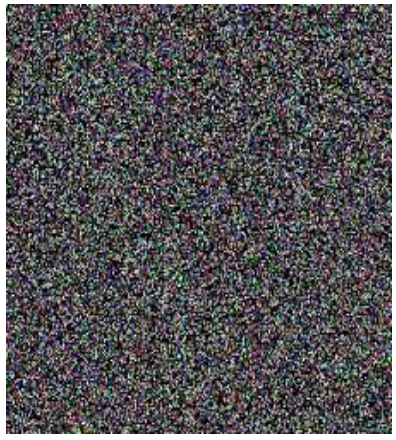
DNN

Thomas

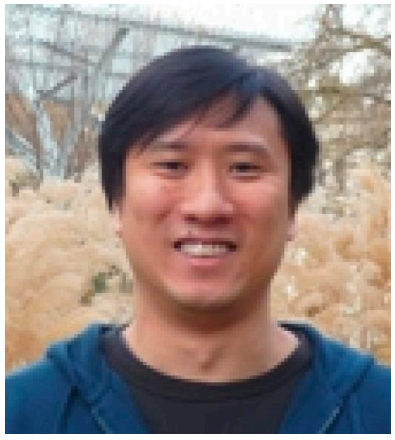
Adversarial Examples



$+0.0001\times$



$=$



DNN

↓

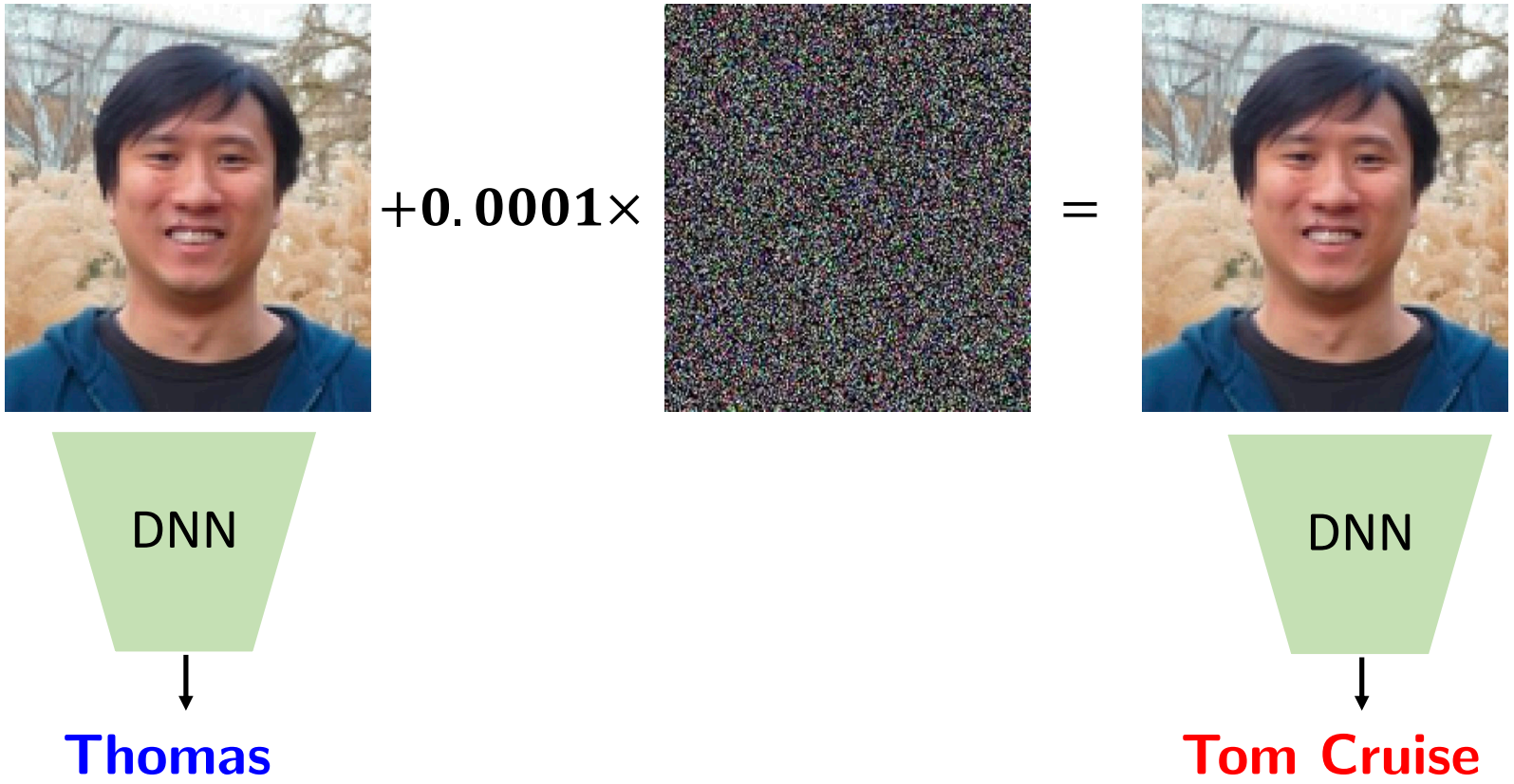
Thomas

DNN

↓

Tom Cruise

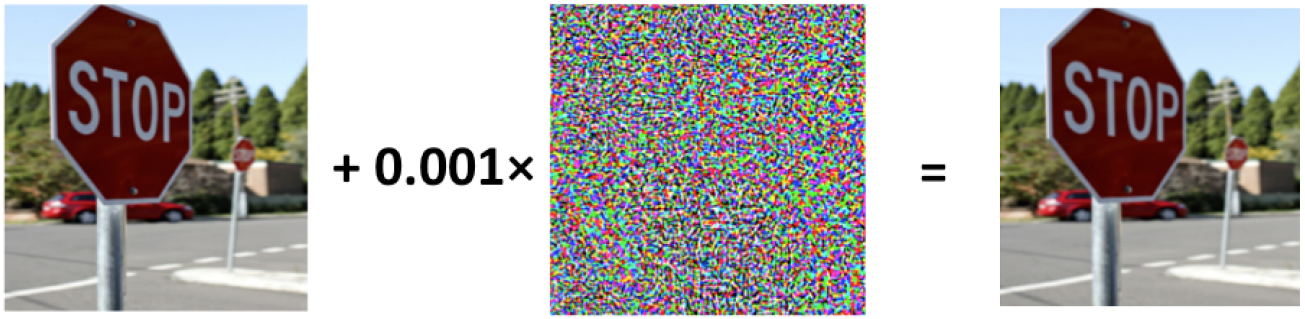
Adversarial Examples



A carefully crafted adversarial example can easily fool a deep network

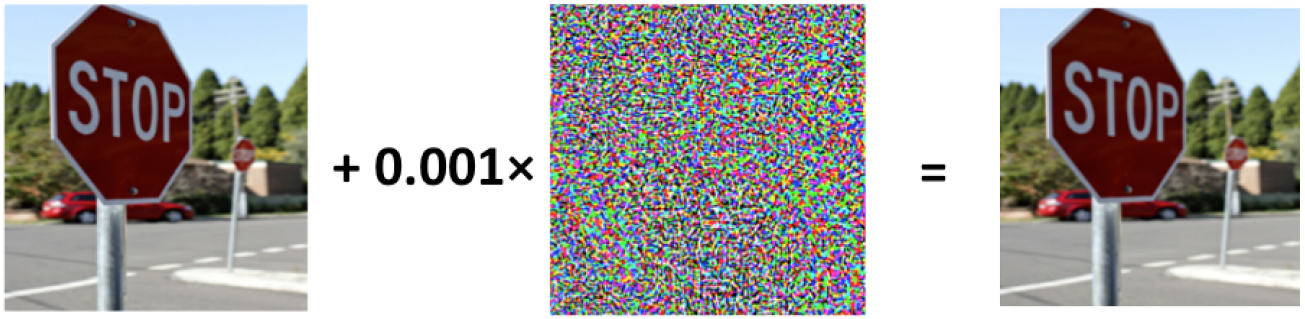
Adversarial Examples

- Robustness is critical in real systems



Adversarial Examples

- Robustness is critical in real systems



Not safe!

Research Questions

Attack: How to craft adversarial example?

Defense: How to **improve** robustness?

Outline

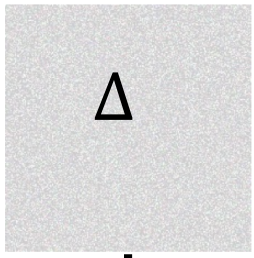
Notations and Attack Procedure

Model f

$\mathbf{y_0} = \mathbf{f}(\mathbf{x_0})$

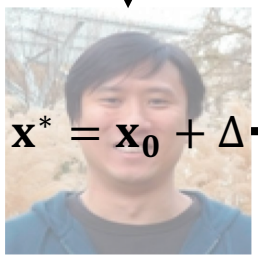
Attack: small perturbation

$+$



$\neq$

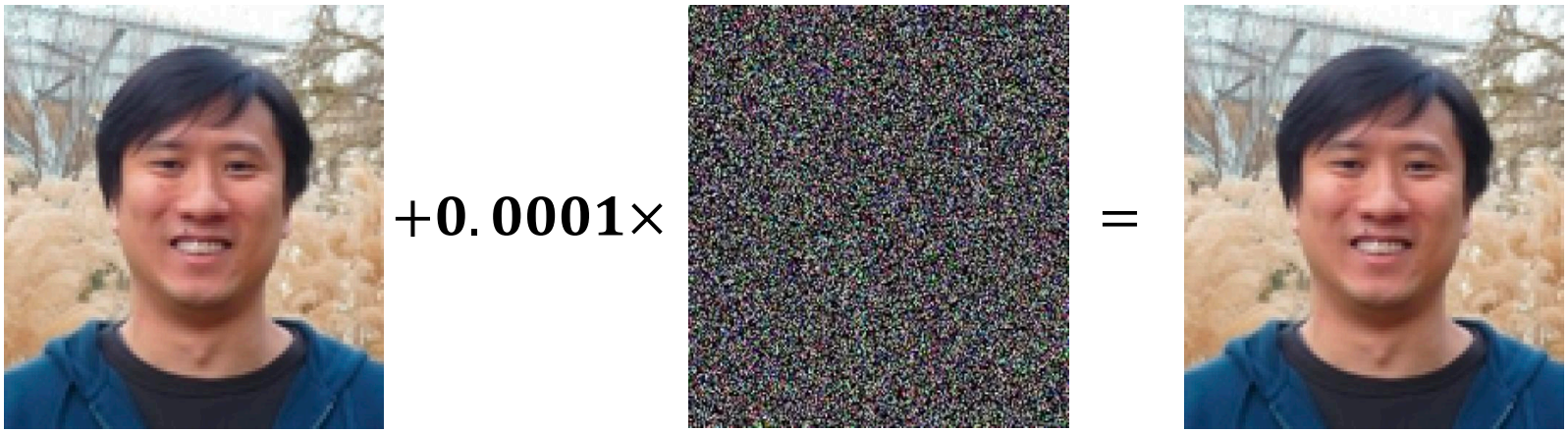
Adversarial example



Model f

$\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*)$

Type of Attacks



Thomas $\mathbf{y}^*$

1. Untargeted attack: $y^* \neq$Thomas
2. Targeted attack: For target class $t =$Tom Cruise, the attacker wants $y^* =$Tom Cruise

Projected Gradient Descent Attack (PGD)

Attack as an optimization problem:

- Given prediction model with fixed parameter $\boldsymbol{\theta}$
- $(\boldsymbol{x}_0, y_0)$: input image and label

$$\boldsymbol{\delta} = \arg\max_{\boldsymbol{\delta} \in \mathcal{S}} L(\boldsymbol{\theta}, \boldsymbol{x}_0 + \boldsymbol{\delta}, y_0)$$

$$\boldsymbol{x}^* = \boldsymbol{x}_0 + \boldsymbol{\delta}$$

L: loss function training the classifier

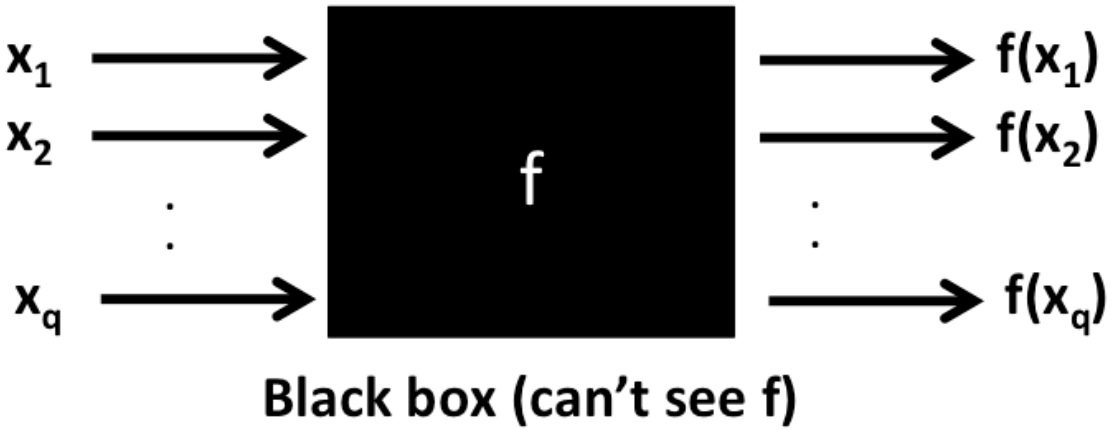
$\boldsymbol{\delta}$: adversarial perturbation

$\mathcal{S} \in \mathbb{R}^d$: allowed perturbations, usually chosen to be $\{\boldsymbol{\delta} | \|\boldsymbol{\delta}\|_\infty \leq \epsilon\}$

(Madry et al., 2018)

Adversarial Attack: Other setting

- Black-box setting: Only part of the information is available



- Distortion can be measured by other norms:
  (e.g., ℓ_2, Elastic net, ...)
  (Carlini et al., 2017; Chen et al., 2018; Brendel et al. 2017)

Attack vs. Defense

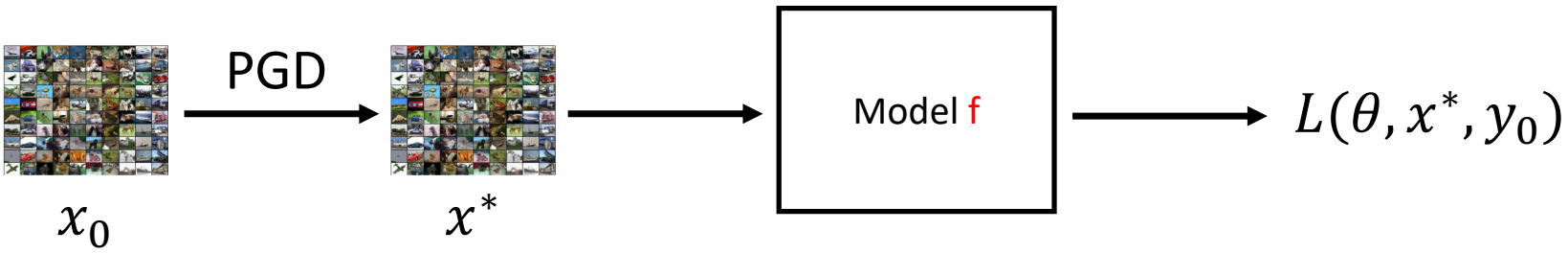
Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

Nicholas Carlini David Wagner
University of California, Berkeley

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye [*1] **Nicholas Carlini** [*2] **David Wagner** [2]

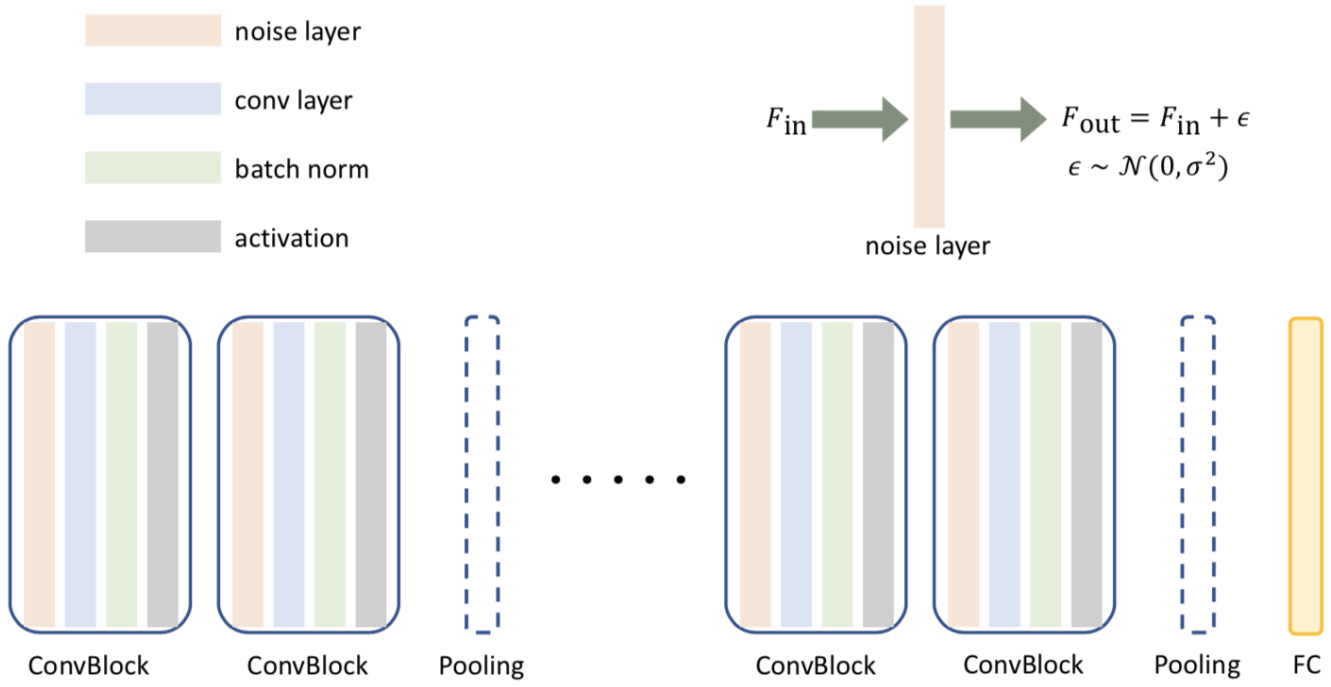
Previous Method 1: Madry's adversarial training



- **Madry's adversarial training:** Madry et al. (2018) proposed to incorporate the adversarial search inside the training process, by solving the following robust optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} L(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\delta}, y) \right\}$$

Previous Method 2: Random Self-Ensemble



- **Random Self-Ensemble:** Liu et al. proposed a "noise layer", which fuses output of each layer with Gaussian noise.

Outline

Proposed Defense Methods

1. Adversarial Bayesian Neural Network (Adv-BNN)
2. Embedding Regularized Classifier (ER-Classifier)

Adversarial Bayesian Neural Network (Adv-BNN)

Observation 1

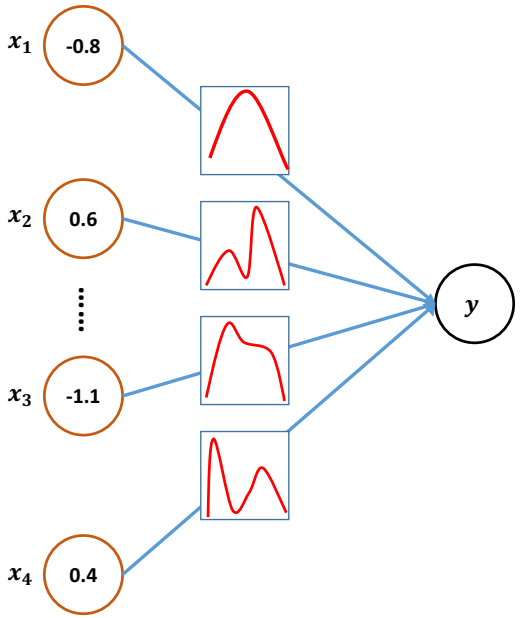
Liu et al., 2018: Adding stochastic components in the neural network to hide gradient information from attacker improves robustness.

Observation 2

Madry et al., 2018: Min-max formulation of training DNN classifier improves the robustness against adversarial examples.

- Our approach: combining adversarial training with Bayesian neural network
  - Bayesian neural network
  - Min-max optimization

Bayesian Neural Network (BNN)



- All weights are represented by probability distributions over possible values
- BNN: a probabilistic model $p(y|\boldsymbol{x}, \boldsymbol{w})$

(Blundell et al., 2015)

Adv-BNN: Objective Function

Recall BNN objective function:

$$\arg\max_{\boldsymbol{\mu},\boldsymbol{s}} \Big\{ \sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}} \mathbb{E}_{\boldsymbol{w}\sim q_{\boldsymbol{\mu},\boldsymbol{s}}} \log p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) - \mathrm{KL}(q_{\boldsymbol{\mu},\boldsymbol{s}}(\boldsymbol{w}) \parallel p(\boldsymbol{w})) \Big\}$$

Adv-BNN objective function:

$$\arg\max_{\boldsymbol{\mu},\boldsymbol{s}} \Big\{ [\sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}} \min_{\|\boldsymbol{\delta}\|_\infty\le\epsilon} \mathbb{E}_{\boldsymbol{w}\sim q_{\boldsymbol{\mu},\boldsymbol{s}}} \log p(y_i|\boldsymbol{x}_i+\boldsymbol{\delta},\boldsymbol{w})] - \mathrm{KL}(q_{\boldsymbol{\mu},\boldsymbol{s}}(\boldsymbol{w}) \parallel p(\boldsymbol{w})) \Big\} \tag{1}$$

- $\mathcal{D}$: data distribution
- $\boldsymbol{\delta}$: adversarial distortion
- ϵ: maximum ℓ_∞ distortion
- $p(\boldsymbol{w})$: prior distribution, $\mathcal{N}(0_d, s_0^2 \boldsymbol{I}_{d\times d})$.

Embedding Regularized Classifier (ER-Classifier)

Observation 1

Levina et al., 2005: The only reason any methods work in very high dimensions is that, in fact, the data are not truly high-dimensional.

Observation 2

Tanay et al., 2016: Adversarial samples do not lie on the data manifold.

- How to use the two findings to improve robustness of Deep Neural Network?

Deep Classifier: Framework

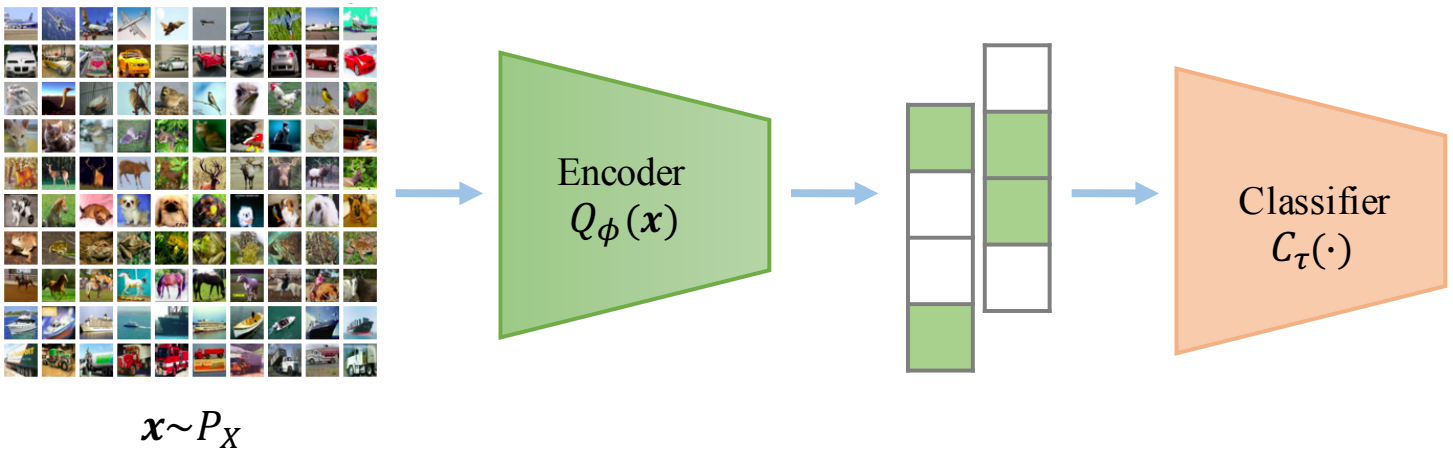


Figure: Deep Classifier

- P_X: Data distribution
- $\boldsymbol{Q}_\varphi$: Encoder (neural network)
- $\boldsymbol{C}_\tau$: Classifier (neural network)

ER-Classifier: Framework

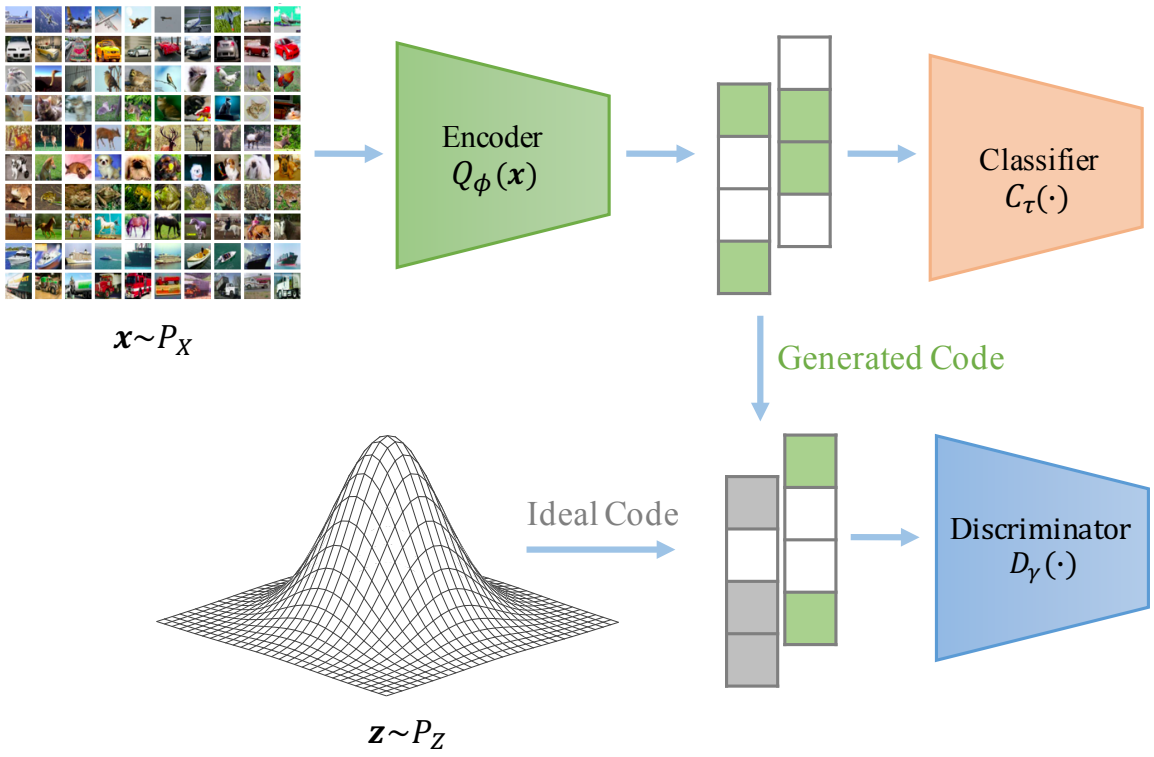


Figure: Embedding Regularized Classifier

- P_Z: Prior distribution
- $\boldsymbol{D}_\gamma$: Discriminator (neural network)

  (discriminator $\boldsymbol{D}$ used to separate "true" $\boldsymbol{z} \sim P_Z$ and "fake" $\tilde{\boldsymbol{z}} \sim \boldsymbol{Q}_Z$)

Embedding Visualization

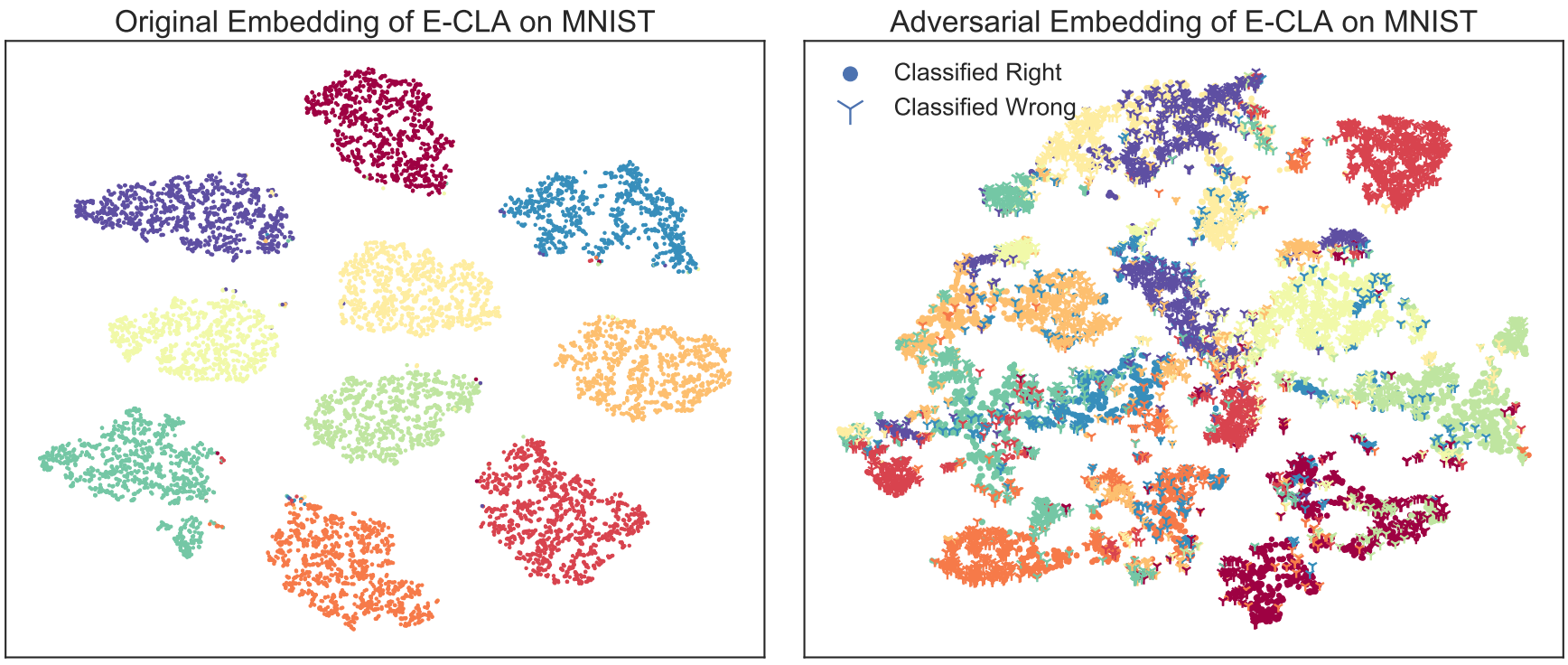


Figure: 2D embeddings for E-CLA on MNIST.

Embedding Visualization

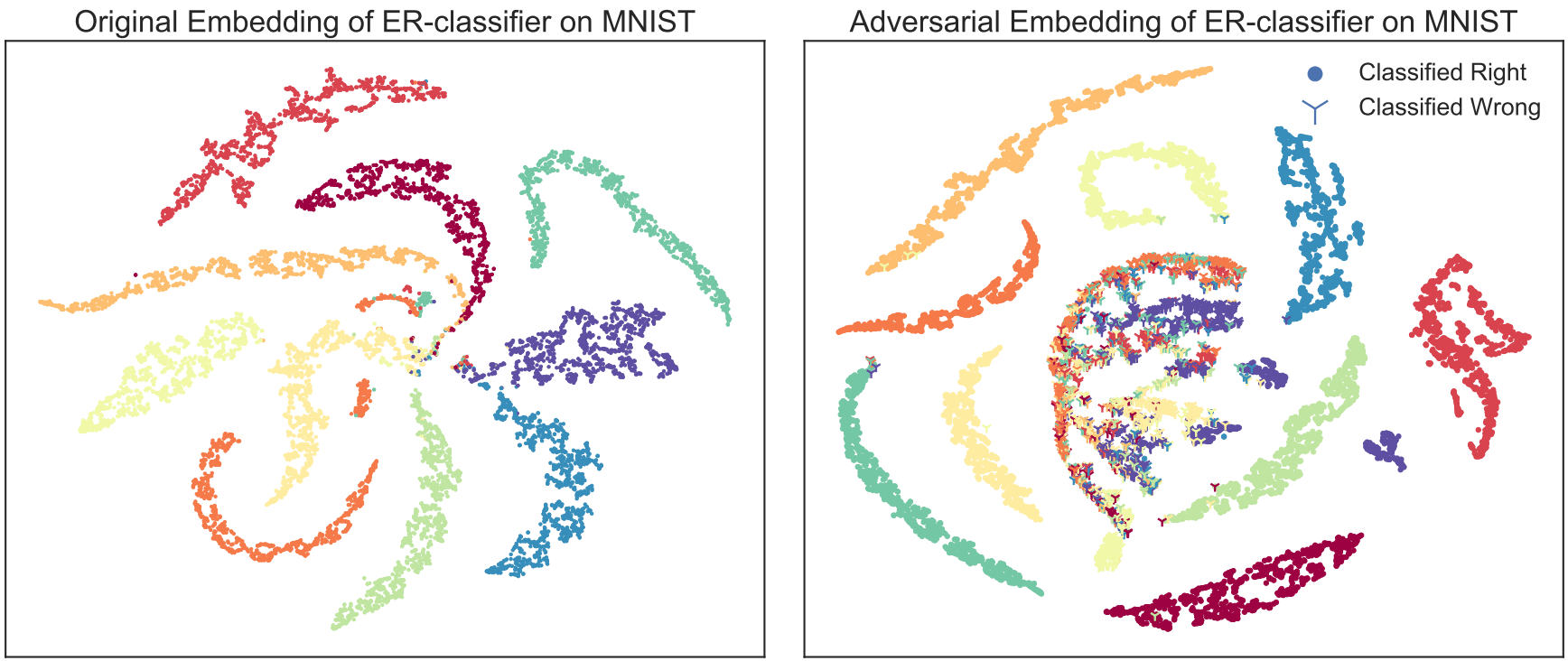


Figure: 2D embeddings for ER-Classifier on MNIST.

Future Work

- Provide uncertainty for label prediction
- Detecting adversarial examples
- Better evaluation methods
- Robustness for statistical model

Outline

Introduction: Group Comparison

Figure: NBA and Dota 2

- Given the results of previous games, how to predict the outcome of an unseen game?

Problem Setting

- Number of individuals: n
- Number of observed comparisons: T
- Each game involves two disjoint teams: I_t^+ and I_t^-, each of them is a subset of $\{1, \cdots, n\}$
- Possible number of teams: N, where $N = \binom{n}{L}$ and $L = |I_t^+| = |I_t^-|$.
- The outcome of game t: o_t.
  - Measured outcomes (scores): $o_t \in \mathbb{R}$
  - Binary indicator outcomes (wins/losses): $o_t \in \{+1, -1\}$.

Example: only players 2,3,4 are on the winning team, $I_t^+ = \{2, 3, 4\}$

Outline

Assumptions of Previous Work

Assumption 1

The team's score is the sum over team members' scores: $s_t^+ = \sum_{i \in I_t^+} w_i$, *where* w_i *is the ability of player* i. *The observed outcome is determined by* $s_t^+ - s_t^-$.

Logistic Regression Model

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} \sum_{t=1}^{T} \ell(\boldsymbol{w}^T \boldsymbol{x}_t, o_t) + R(\boldsymbol{w}), \tag{2}$$

- ℓ is logistic loss.
- $\boldsymbol{w}$ are the individual scores we want to learn.
- $\boldsymbol{x}_t \in \mathbb{R}^n$ is the indicator vector, where $(\boldsymbol{x}_t)_j = 1$ if $j \in I_t^+$, $(\boldsymbol{x}_t)_j = -1$ if $j \in I_t^-$, and $(\boldsymbol{x}_t)_j = 0$ for all other elements.

Limits of these Models

- Common weakness: they cannot evaluate the player-interaction effects.
- Solution: include player-interaction effects in the model
- Two questions regarding including player-interaction effects:
  - Can incorporating player-interaction effects improve the prediction accuracy of the model?
  - When n is large, how can our algorithm scale up?

Outline

Assumptions of New Model

Assumption 2

Assume each user has their individual score w_{jj}, and for each pair of users there is a pair-wise score w_{ij}. A team's ability is modeled by

$$s_t^+ = \sum\nolimits_{i,j \in I_t^+} w_{ij}.$$

Goal: learn the model so that the score s_t^+ is larger than s_t^- for each game.

Basic Model for Higher Order Interactions

- **Basic HOI**:

$$\min_{W \in \mathbb{R}^{n \times n}} \sum_{t=1}^{T} \ell((\boldsymbol{e}_t^+)^T W(\boldsymbol{e}_t^+) - (\boldsymbol{e}_t^-)^T W(\boldsymbol{e}_t^-), o_t) + \lambda \|W\|_F^2. \quad (3)$$

where $\boldsymbol{e}_t^+$ is the indicator vector for I_t^+

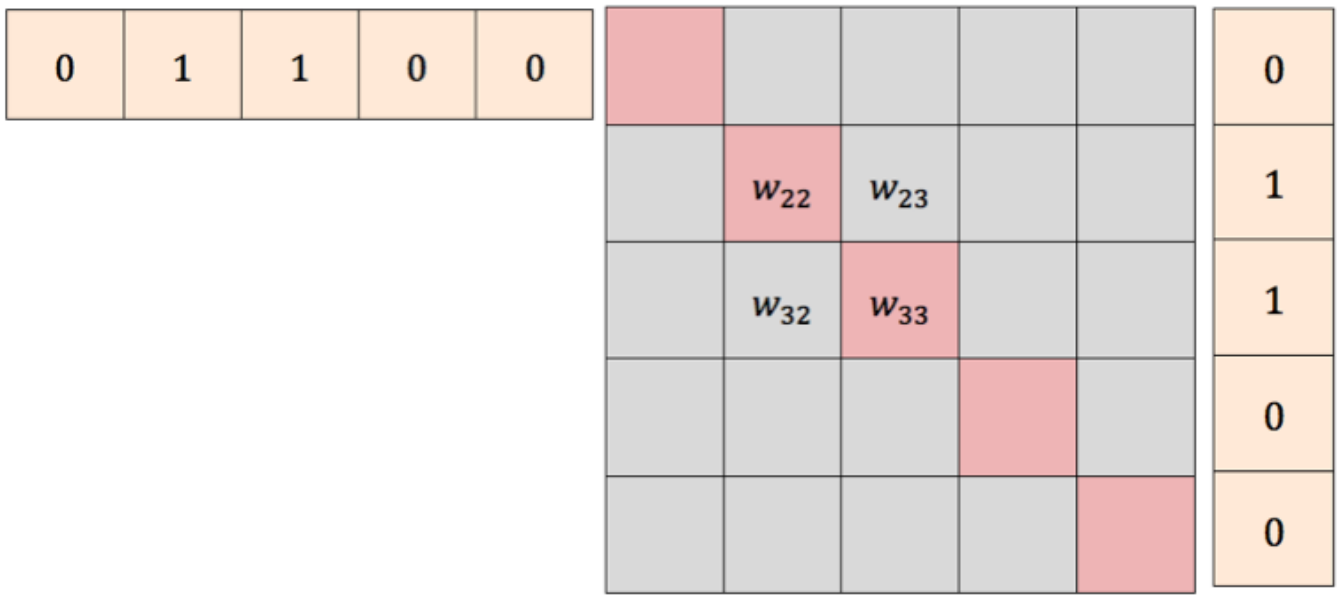


Figure: Example of $(\boldsymbol{e}_t^+)^T W(\boldsymbol{e}_t^+)$, and W is the matrix we want to learn

Limits of Basic HOI

- Basic HOI can be reformulated as logistic regression and solved by standard methods.
- Sample complexity: Problem (3) has $\frac{n(n+1)}{2}$ parameters, so the degree of freedom grows with $O(n^2)$.
- Computing: Problem (3) requires $O(n^2)$ memory to store the W matrix. Therefore, a standard solver will be hard to scale beyond $30,000$ players.
- Conclusion: When n is very large, Basic HOI doesn't work.

How to handle pair-wise effects when n is large?

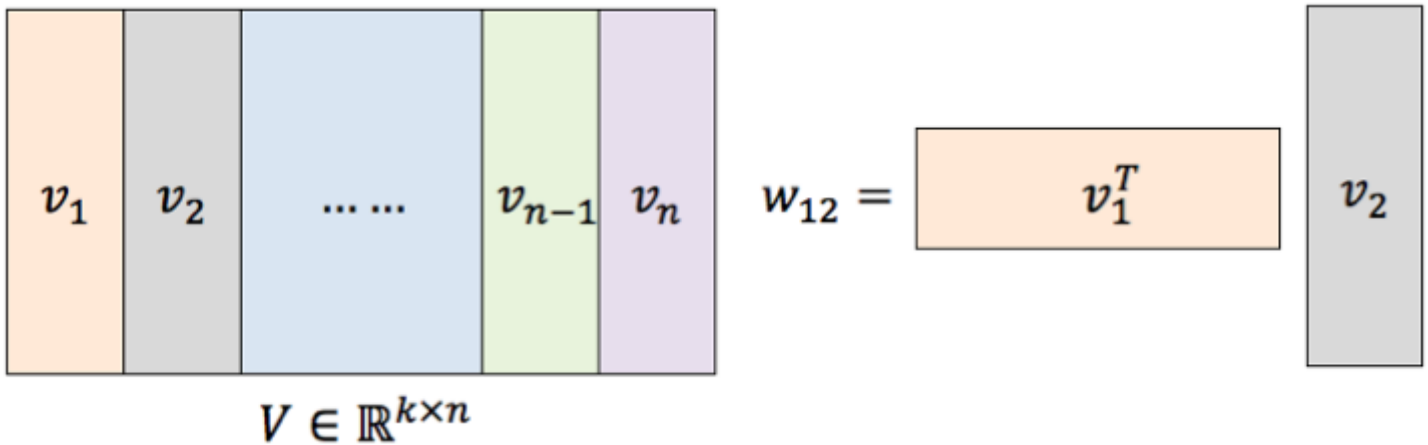
Assumptions of Factorization Model

Assumption 3

We assume a team's score can be written as

$$s_t^+ = \sum_{i \in I_t^+} w_i + \sum_{i \in I_t^+} \sum_{j \in I_t^+} \boldsymbol{v}_i^T \boldsymbol{v}_j$$

- Model parameters that have to be estimated are $\boldsymbol{w} \in \mathbb{R}^n$ and $V \in \mathbb{R}^{k \times n}$, each $\boldsymbol{v}_j$ is the j-th column of V.
- The hyper-parameter k defines the dimensionality of the factorization.



Factorization Model for Higher Order Interactions

- **Factorization HOI**:

$$\begin{aligned} & \underset{\boldsymbol{w}\in\mathbb{R}^n, V\in\mathbb{R}^{k\times n}}{\operatorname{argmin}} \sum_{t=1}^{T} \ell(s_t^+ - s_t^-) + \frac{\lambda_w}{2}\|\boldsymbol{w}\|_2^2 + \lambda_V \|V\|_F^2 \\ = & \underset{\boldsymbol{w}\in\mathbb{R}^n, V\in\mathbb{R}^{k\times n}}{\operatorname{argmin}} \sum_{t=1}^{T} \ell(\boldsymbol{w}^T(\boldsymbol{e}_t^+ - \boldsymbol{e}_t^-) + \sum_{i,j\in I_t^+} \boldsymbol{v}_i^T \boldsymbol{v}_j - \sum_{i,j\in I_t^-} \boldsymbol{v}_i^T \boldsymbol{v}_j) \\ & + \frac{\lambda_w}{2}\|\boldsymbol{w}\|_2^2 + \lambda_V \|V\|_F^2, \end{aligned} \tag{4}$$

where λ_w and λ_V are the regularization parameters.

Data Statistics

Figure: Character vs. Character, Player vs. Player

Datasets	HotS Tournament (Character)	HotS Tournament (Player)	HotS Public (Character)	HotS Public (Player)	Dota 2 (Character)	Dota 2 (Player)
Number of Games (T)	9,610	9,610	139,462	139,462	46,459	46,459
Number of Individuals (n)	54	3,470	62	7,251	113	30,452

Table: Dataset Statistics

Methods

- Basic HOI: the proposed basic model using pairwise information.
- Factorization HOI: the proposed latent factor model, which approximates the pairwise interaction by a factor form.
- Trueskill: the state-of-the-art algorithm used in all the online game matching engines.
- Bradley-Terry model: the generalized Bradley-Terry model for group comparison data.
- Logistic Regression (LR): another baseline for individual score model using logistic loss.

Experiment Results

Table: Performance of the proposed algorithm and other algorithms. The numbers are prediction accuracy (%), and "oom" indicates out of memory here.[4]

Datasets	LR	Trueskill (1)	Trueskill (10)	Bradley-Terry	Basic HOI	Factorization HOI
HotS Tournament (C)	59.73	62.90	58.48	59.52	80.59	77.84
HotS Tournament (P)	83.45	80.02	84.50	84.18	83.89	85.17
HotS Public (C)	54.34	53.36	53.06	53.50	54.45	54.75
HotS Public (P)	54.01	53.64	53.87	53.92	53.39	55.76
Dota 2 (C)	61.64	52.50	52.61	61.37	65.34	63.72
Dota 2 (P)	65.98	62.16	64.26	62.72	oom	68.25

Patterns Learned by Factorization HOI

Table: Top-5 pairs and bottom-5 character pairs learned by our model on Heroes of the storm tournament data.

Top 5 pairs	Bottom 5 pairs
(Lunara, Leoric)	(Raynor, Zeratul)
(Kerrigen, Sylvanas)	(Illidan, Thrall)
(Reghar, Illidan)	(Sonya, Zeratul)
(Chen, Jaina)	(Muradin, Lt. Morales)
(Thrall, Valla)	(Anub'arak, Illidan)

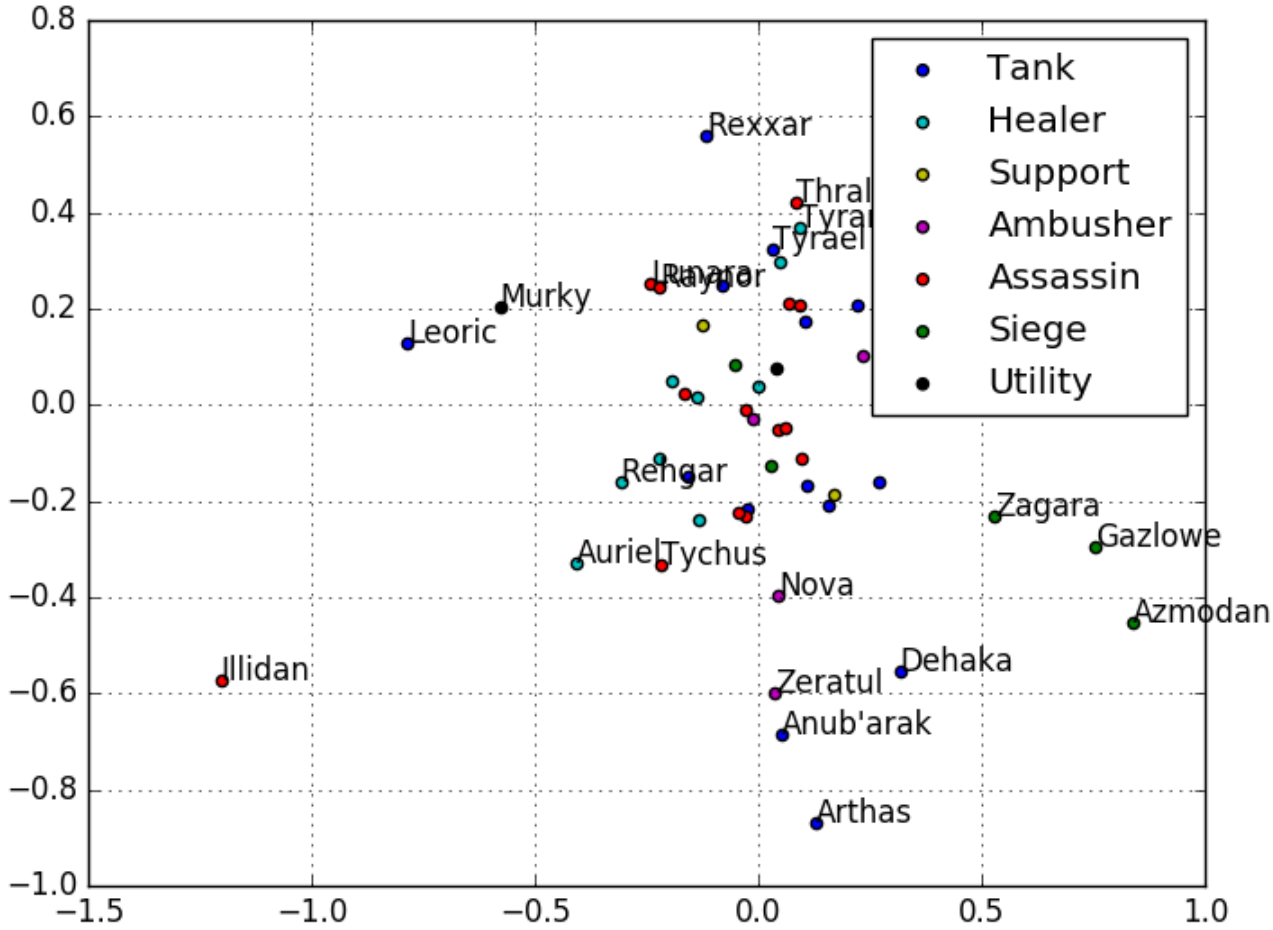


Figure: Projection of interaction features for each character ($\boldsymbol{v}_i$) to 2-D space. Colors represents the official categorization for these characters.

References

[1] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. *Weight uncertainty in neural networks.* ICML, 2015

[2] Tanay, T., and Griffin, L. *A boundary tilting persepective on the phenomenon of adversarial examples.* ArXiv, 2016

[3] Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. *Synthesizing robust adversarial examples.* ICML, 2018.

[4] Carlini, N., and Wagner, D. *Towards evaluating the robustness of neural networks.* SC, 2017.

[5] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. *Towards deep learning models resistant to adversarial attacks.* ICLR, 2018.

[6] Levina, E., and Bickel, P. J. *Maximum likelihood estimation of intrinsic dimension.* NIPS, 2005.

[7] Simon-Gabriel, C. J., Ollivier, Y., Bottou, L., Schölkopf, B., and Lopez-Paz, D. *Adversarial vulnerability of neural networks increases with input dimension.* ICML, 2019.

References

[8] Liu, X., Cheng, M., Zhang, H., and Hsieh, C. J. *Towards robust neural networks via random self-ensemble.* ECCV, 2018.

[9] Li, Y., Cheng, M., Fujii, K., Hsieh, F., and Hsieh, C. J. *Learning from group comparisons: exploiting higher order interactions.* NIPS, 2018.

[10] Liu, X., Li, Y., Wu, C., and Hsieh, C. J. *Adv-bnn: Improved adversarial defense through robust bayesian neural network.* ICLR, 2019.

[11] Yi, J., Hsieh, C. J., Varshney, K. R., Zhang, L., and Li, Y. *Scalable demand-aware recommendation.* NIPS, 2017.

[12] Li, Y., Min, M. R., Yu, W., Hsieh, C. J., Lee, T., and Kruus, E. *Defending against adversarial examples by regularized deep embedding.* ICML (to be submitted), 2020.

[13] Li, Y., Yu, W., Min, M. R., Lee, T., Kruus, E., Wang, W., and Hsieh, C. J. *Detecting adversarial examples with regularized deep embedding.* IJCAI (to be submitted), 2020.

Embedding Visualization

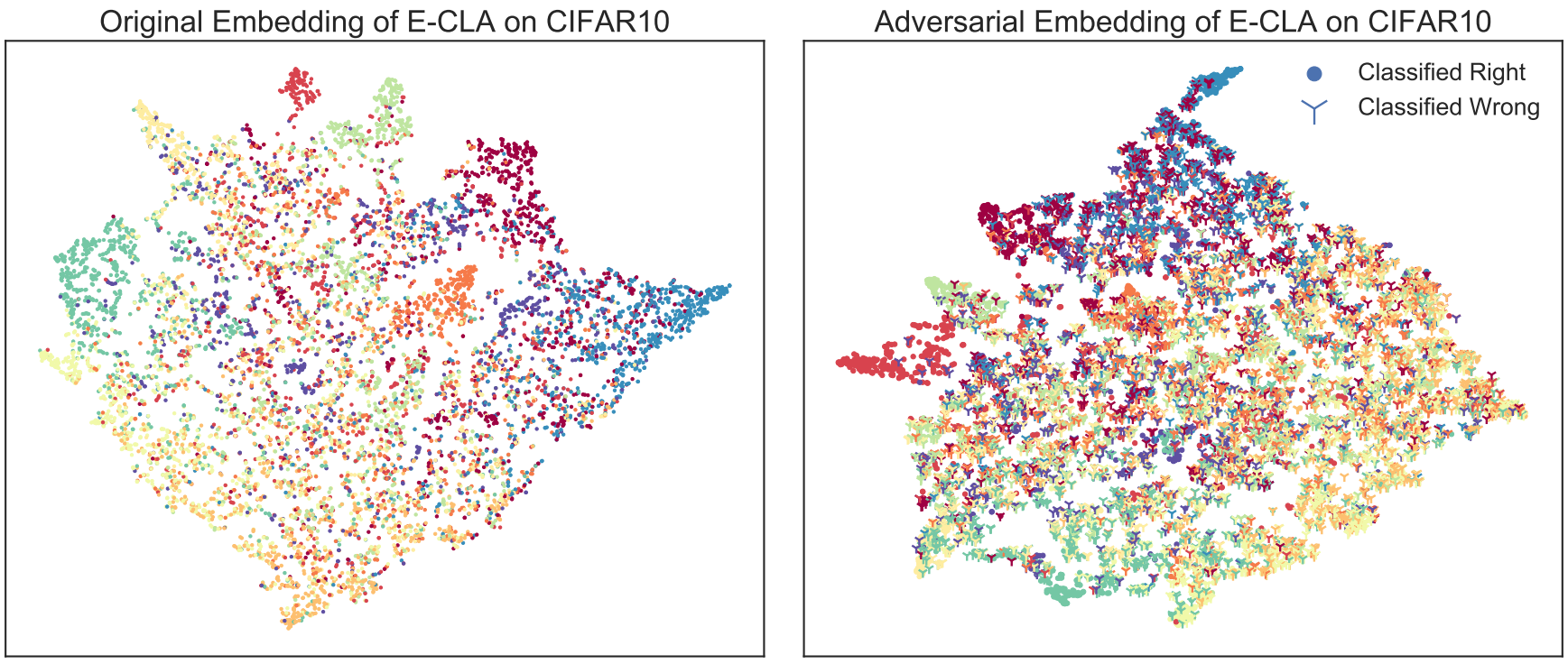


Figure: 2D embeddings for E-CLA on CIFAR10.

Embedding Visualization

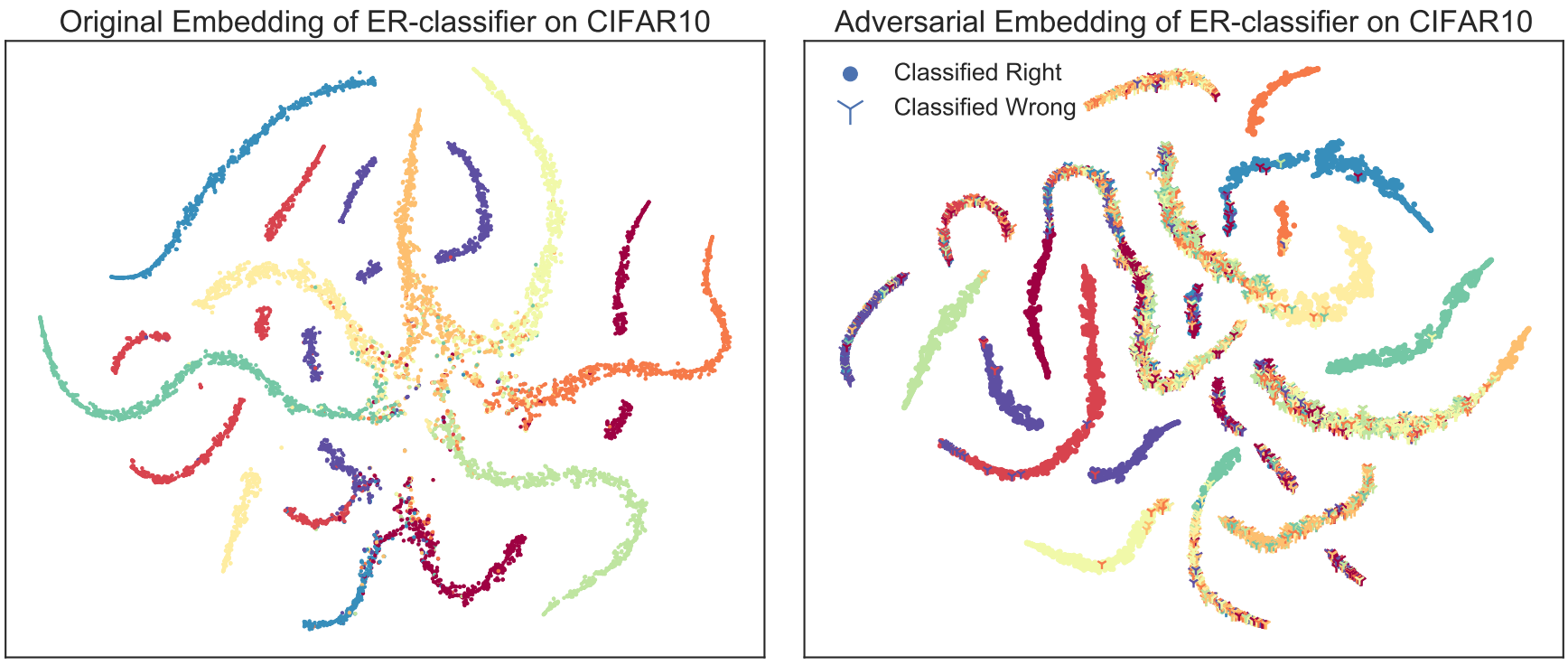


Figure: 2D embeddings for ER-Classifier on CIFAR10.