Small Change Matters

Towards Robust Deep Learning with Optimal Transport

Dinh Phung
dinh.phung@monash.edu

A@M Colloquium, Melbourne University, July 2023

Robust and Trustworthy AI

- AI impacts us in a profound way
- Rapidly becomes more autonomous with fully automated critical decisions

Problem: a magnitude of order more serious than, probably, the rate of AI growth if things go wrong!

1. **Tesla Autopilot kills**

Tesla Autopilot Crashes: With at Least a Dozen Dead, 'Who's at Fault, Man or Machine?'

After a Tesla car reportedly on autopilot recently killed two people in China and many other drivers report self-driving system malfunctions, the automaker is facing increased scrutiny over its technology

by **Lauren Richards** — December 1, 2022 in **Business**, **Corporations**, **Society**, **Tech**

Robust and Trustworthy AI

- AI impacts us in a profound way
- Rapidly becomes more autonomous with fully automated critical decisions

Problem: a magnitude of order more serious than, probably, the rate of AI growth if things go wrong!

1. **Tesla Autopilot kills**
2. **IBM Watson recommends wrong cancer treatment**

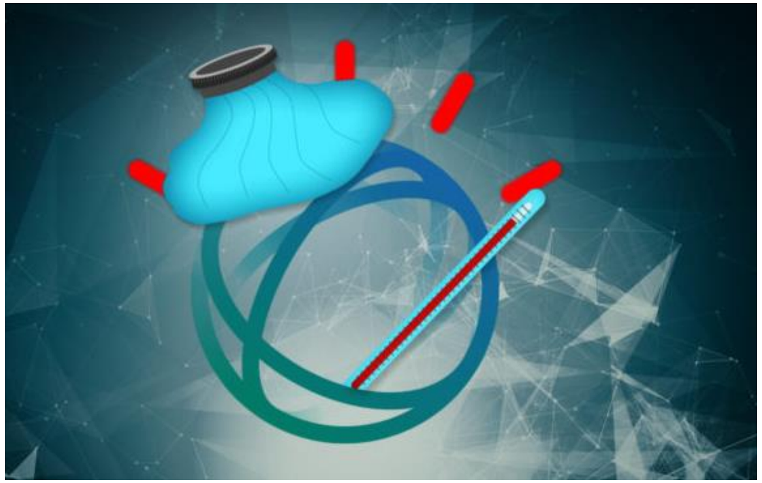
EXCLUSIVE

STAT+

IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show

By Casey Ross and Ike Swetlitz July 25, 2018

Reprints



Robust and Trustworthy AI

- AI impacts us in a profound way
- Rapidly becomes more autonomous with fully automated critical decisions

Problem: a magnitude of order more serious than, probably, the rate of AI growth if things go wrong!

1. **Tesla Autopilot kills**
2. **IBM Watson recommends wrong cancer treatment**
3. **LLM-based Chatbot [Elisa] encourages suicide**

(source- Medium)

47 Comments

WEIRD BUT TRUE

Married father commits suicide after encouragement by AI chatbot: widow

By Ben Cost

March 30, 2023 | 5:59pm | Updated

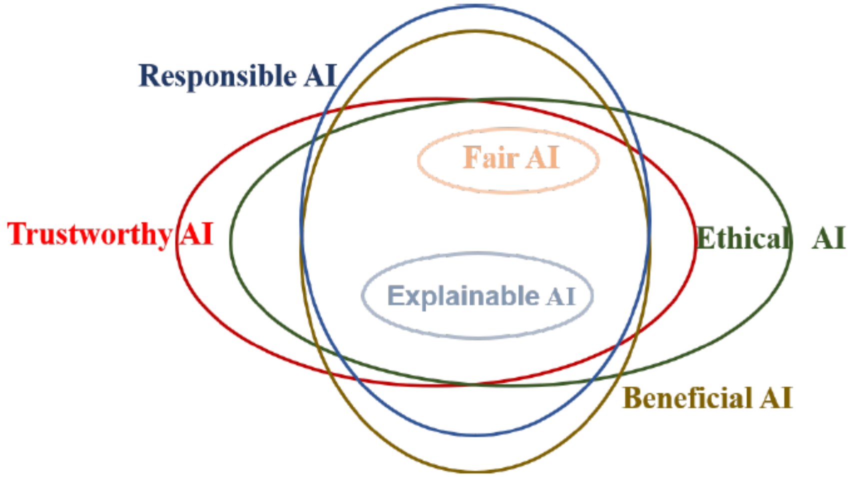
Robust and Trustworthy AI

Amazon’s AI Recruitment Tool Bias, Microsoft Chatbot Tay Offensive Tweets, Apple Card Gender Bias, Uber’s Greyball program, …. **Google Photo Misclassification**

1. **Tesla Autopilot kills**
2. **IBM Watson recommends wrong cancer treatment**
3. **LLM-based Chatbot [Elisa] encourages suicide**

Robust vs Trustworthy AI

- Robust AI: consistent performance
  - missing/incomplete data, out-of-distribution shift, noisy, unreliable scenarios, day/light, …
  - under deliberate adversarial attacks to disrupt its functioning.
- Trustworthy AI: robustness + transparent, accountable, bias-free
  - bring confidence and trust to AI adoption to everyday activities.
- Vital to (Human + AI) endeavour!



Other related concepts

Liu et al., Trustworthy AI: A Computational Perspective, *ACM Computing Survey*, 2021.

Adversarial Attack and Robustness

- Deliberately exploit loopholes in the AI system to disrupt its functions
- Deep learning: turns out, it's very easy to hack DNNs!

Heaven D., Deep Trouble for Deep Learning, Vol 574 *Nature*, 2019.

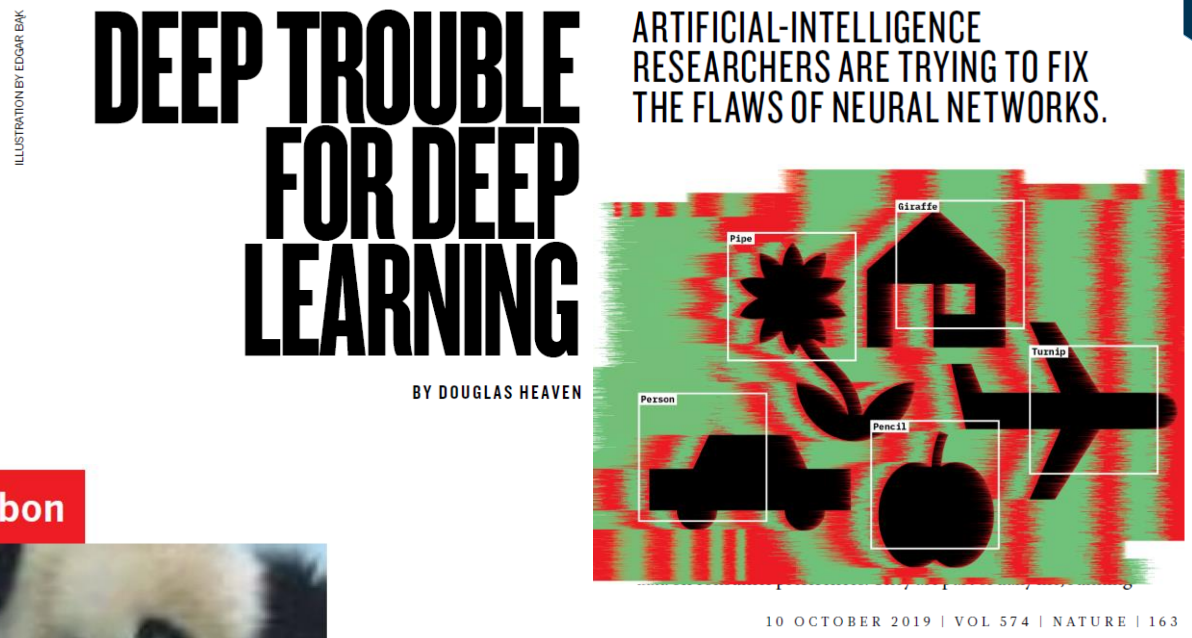
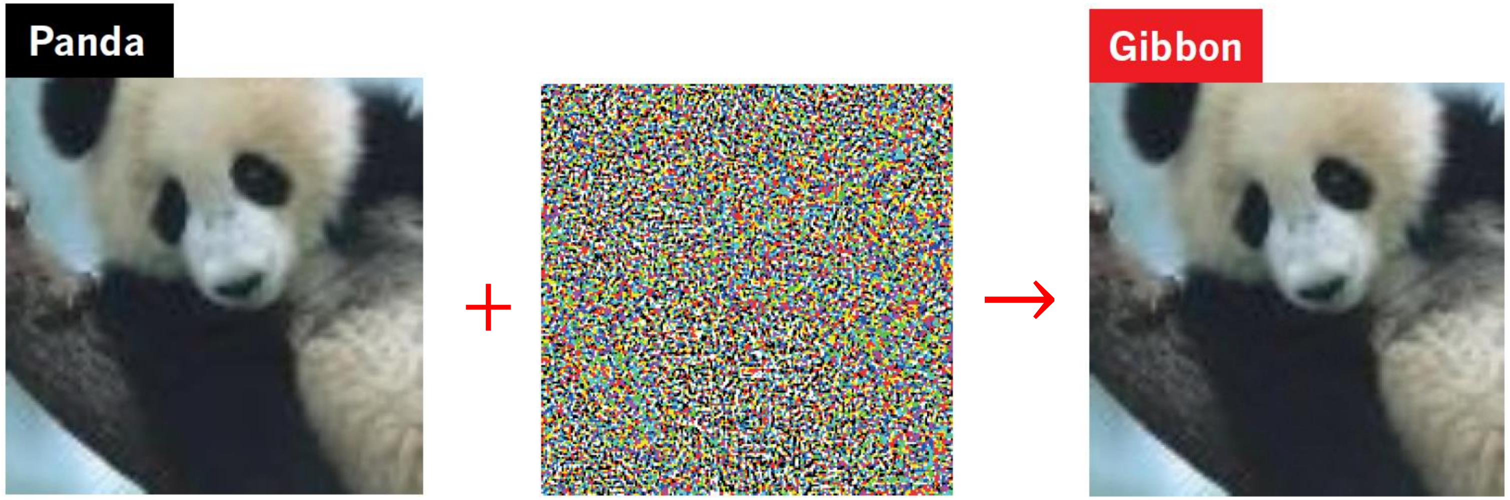


ILLUSTRATION BY EDGAR BĄK
DEEP TROUBLE FOR DEEP LEARNING
ARTIFICIAL-INTELLIGENCE RESEARCHERS ARE TRYING TO FIX THE FLAWS OF NEURAL NETWORKS.
BY DOUGLAS HEAVEN

10 OCTOBER 2019 | VOL 574 | NATURE | 163



ϵ- small perturbation

Adversarial Attack and Robustness

- Deliberately exploit loopholes in the AI system to disrupt its functions
- Deep learning: turns out, it's very easy to hack DNNs!

Heaven D., Deep Trouble for Deep Learning, Vol 574 *Nature*, 2019.

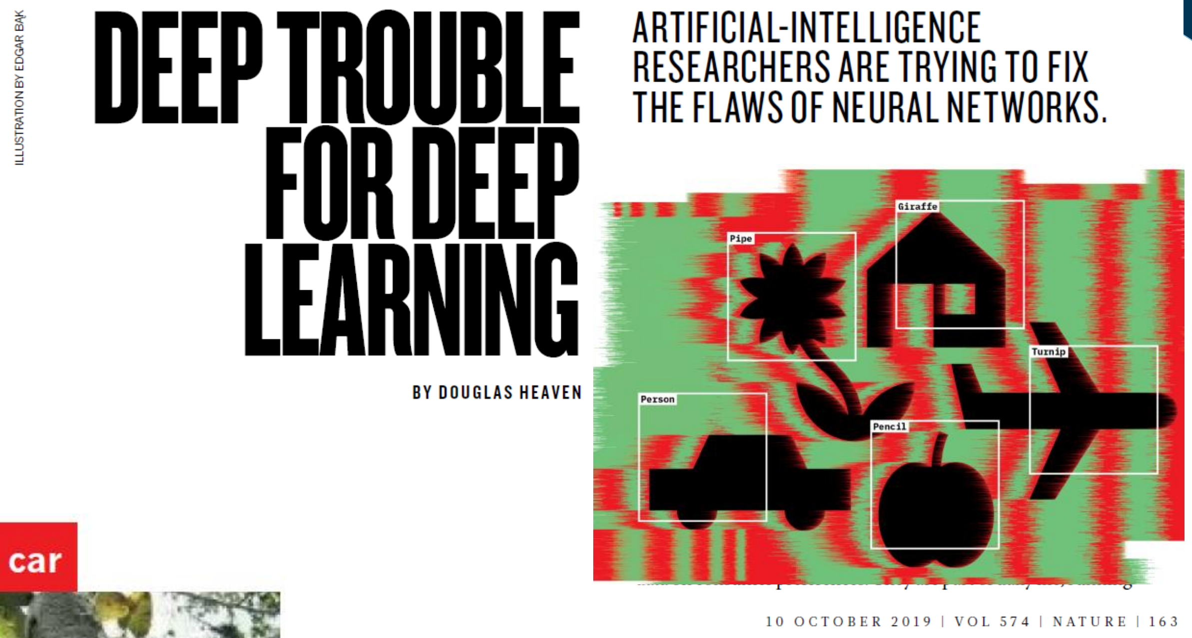


ILLUSTRATION BY EDGAR BĄK

DEEP TROUBLE FOR DEEP LEARNING

ARTIFICIAL-INTELLIGENCE RESEARCHERS ARE TRYING TO FIX THE FLAWS OF NEURAL NETWORKS.

BY DOUGLAS HEAVEN

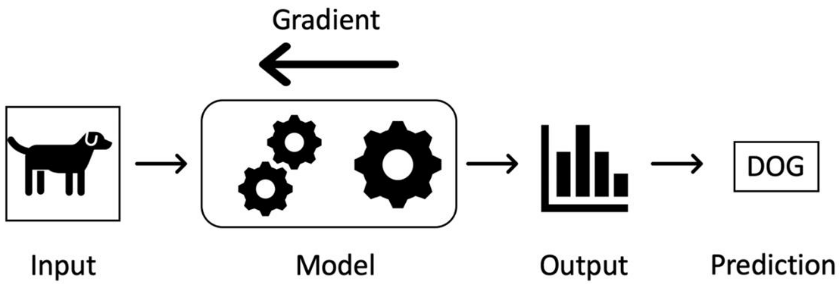
10 OCTOBER 2019 | VOL 574 | NATURE | 163

Targeted Attack

Adversarial Attacks

Whitebox, blackbox, nobox

Whitebox	Blackbox	Nobox
Has access to all model details including defending strategy	Do not have access to internal model	Access to none



Attack setting	Input	Architecture	Output	Gradient
White-box	✓	✓	✓	✓
Black-box	✓	⊘	✓	⊘
No-box	⊘	⊘	⊘	⊘

Adversarial Attacks

Training time attacks

- Training time attacks
  - Backdoor attack: Injecting backdoor into a target model
  - Poison attack: corrupting a target model

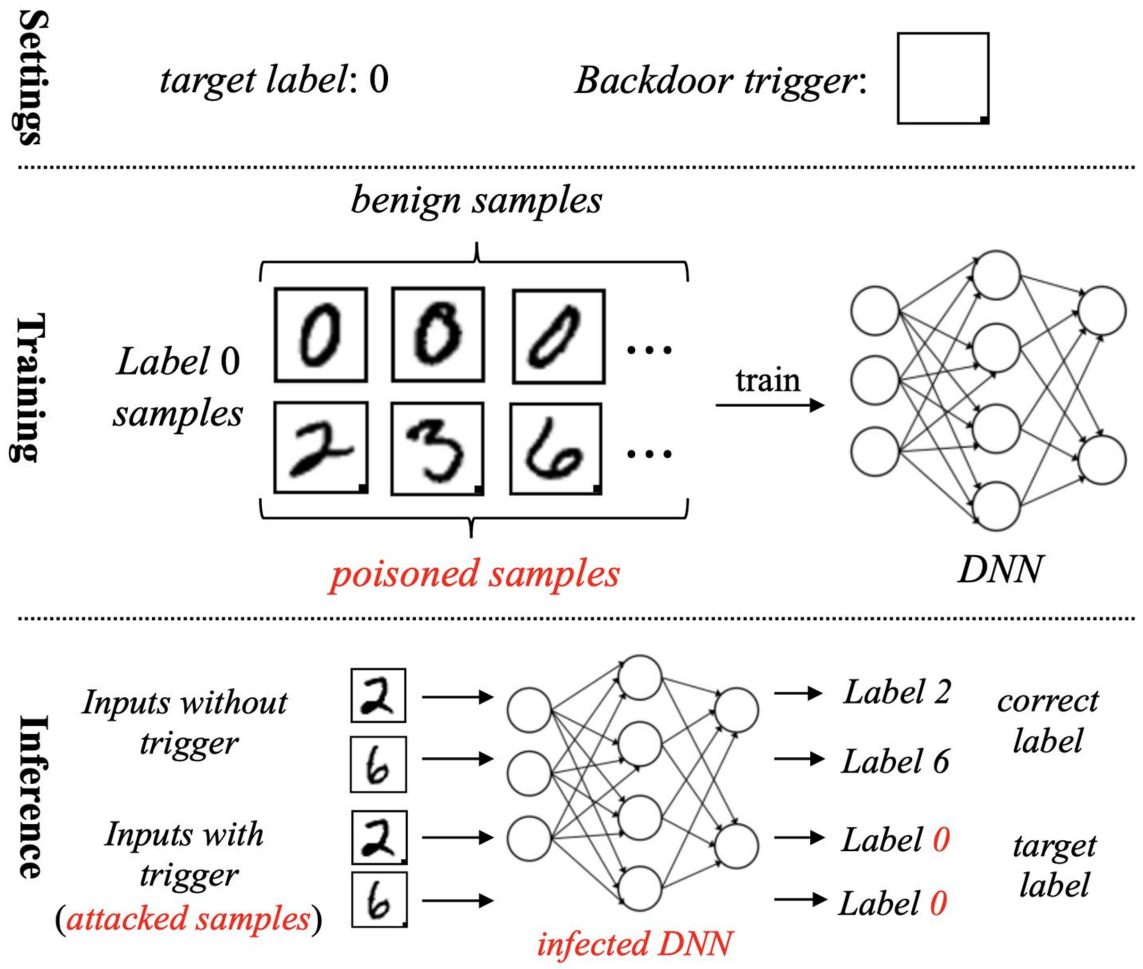
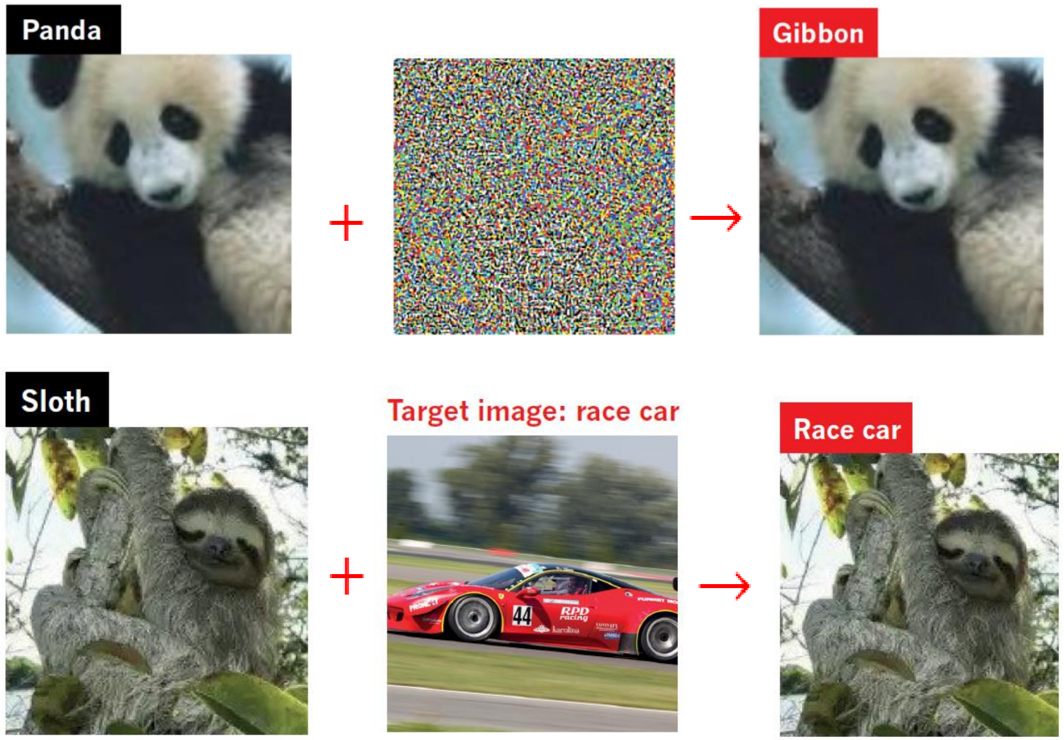


Image credit: Li, Yiming, et al. "Backdoor learning: A survey." *IEEE Transactions on Neural Networks and Learning Systems* (2022).

Adversarial Attacks

Test time attacks

- Training time attacks
  - Backdoor attack: Injecting backdoor into a target model
  - Poison attack: corrupting a target model
- Test time attacks:
  - Evasion attack: manipulating model's prediction (i.e., adversarial examples)
  - Model extraction: stealing model functionality
  - Privacy attack: extracting sensitive training data



Adversarial Attack and Defense

> **"THERE ARE SO MANY DIFFERENT WAYS THAT YOU CAN ATTACK A SYSTEM."**

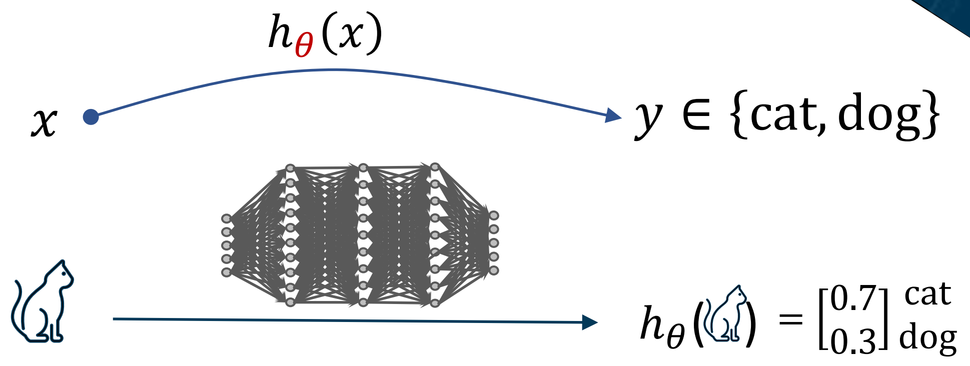
Type of Attacks	Domain Attacked
Adversarial examples	Visual: images, videos
Backdoor attacks	Auditory: speech, music
Poison attacks	Text: sentiment,
Privacy attacks	Graph
...	....

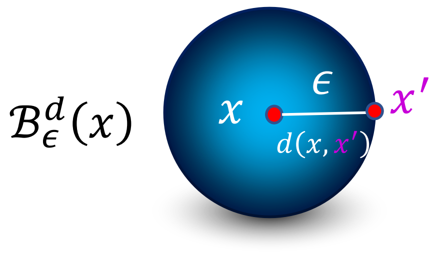
Defense: Adversarial Training, Randomized Smoothing, Adversarial Purifying, and many more.

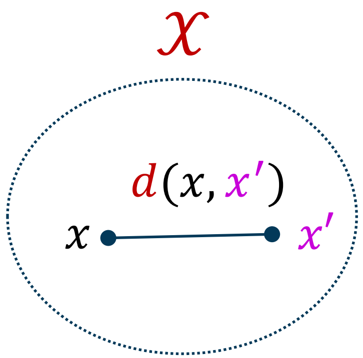
Heaven D., Deep Trouble for Deep Learning, Vol 574 *Nature*, 2019.

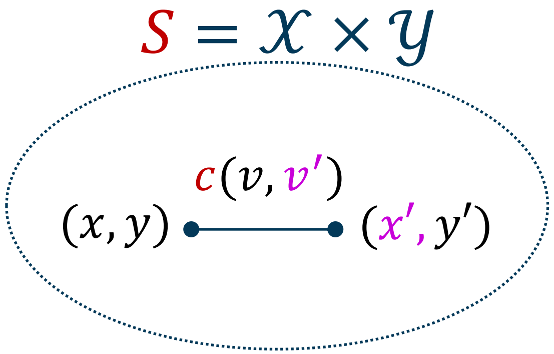
Notation

- $\mathbb{I}_{\{\text{condition}\}} = 1$ if condition is true; 0 otherwise
  - E.g. $\mathbb{I}_{\{1=1\}} = 1, \mathbb{I}_{\{1=2\}} = 0$
- Supervised learning: $h_\theta : \mathcal{X} \to \mathcal{Y}, \theta \in \Omega$
  - Input space $x \in \mathcal{X}$, output space $y \in \mathcal{Y}$
  - Prediction space:
    - $h_\theta(x) \in \Delta^{|\mathcal{Y}|-1}$ (simplex)
    - $h_\theta^j(x) = j^{\text{th}}$ element, $i.e., p(y = j|x)$
    - $\hat{y} = \operatorname{argmax}_j h_\theta(x),\ \hat{y} \in \mathcal{Y}$
- ϵ-vicinity ball, $\epsilon > 0$, $\mathcal{B}_\epsilon^d(x) = \{x' : d(x, x') < \epsilon\}$
  - centred at x induced by metric d on $\mathcal{X}$
- S: a Polish space, endowed with metric $c(v, v')$
  - $c(v, v')$: non-negative, symmetric, triangle inequality
  - We usually consider product spaces: $S = \mathcal{X} \times \mathcal{Y}$ or $S = \mathcal{X} \times \mathcal{X} \times \mathcal{Y}$
  - μ, ν : probability measures, $T: S \to S$: measurable map
  - $T_\# \mu$: push-forward measure of μ via T





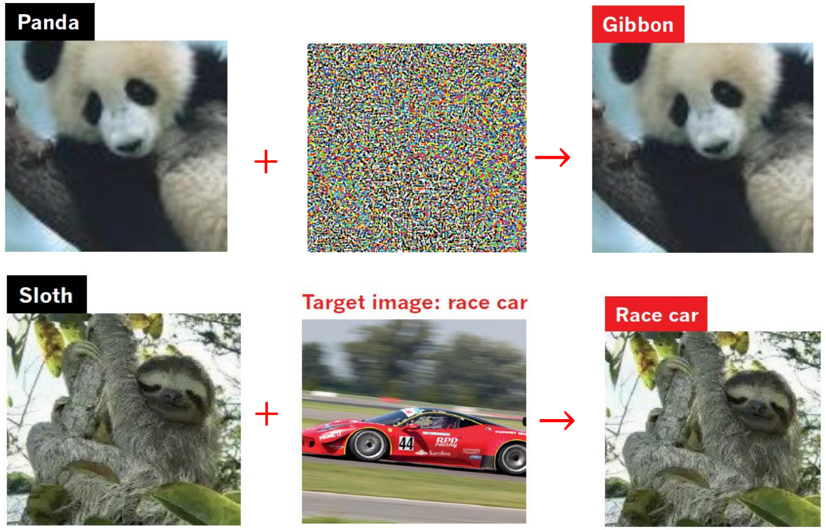




Key concepts

Given (x, y) and a classifier $\hat{y} = h(x)$

- For now, x' is said to be 'similar' to x if $x' \in \mathcal{B}_{\epsilon}^{d}(x)$
- Untargeted attack: find *adversarial* x' such that:
  - x' is similar to x, but classified differently, i.e., $h(x') \neq y$
- Targeted attack: let $y^* \neq y$, find x' such that:
  - x' is similar to x, but classified as y^* instead, i.e, $h(x') = y^*$
- Adversarial training:
  - Given training $D = \{(x_i, y_i), i = 1, \ldots, n\}$, for each x_i find its adversarial x_i' and form $D' = \{(x_i', y_i)\}$
  - Use both D and D' for training
- Defence/adversarial robustness
  - Find $h(x)$ so that $h(x)$ correctly classifies x and its adversarial x' to be in the same class y.
- Note: adversarial samples $\neq$ adversarial attacks
  - The later has a broader context as in 'adversary'



Adversarial Training (AT)

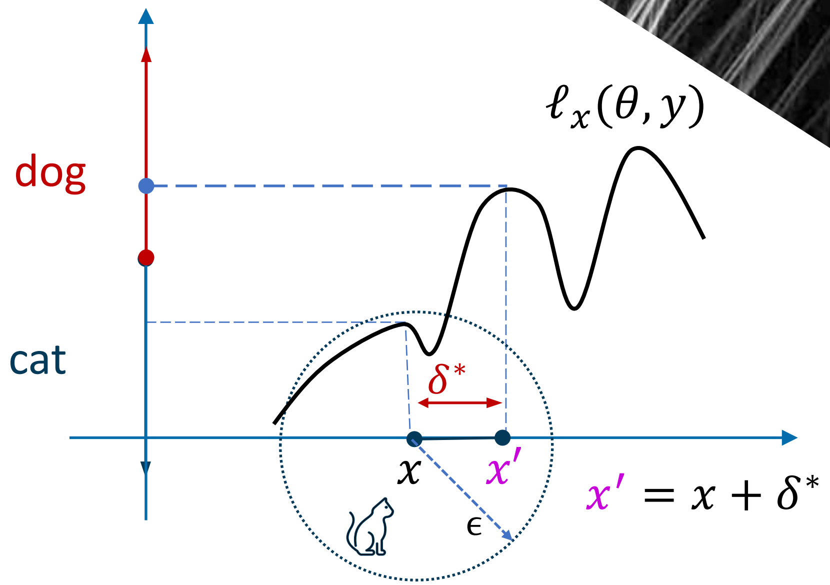
- Projected Gradient Descent (PGD)
  - Find adversarial x' $= x + \delta^*$ where $\Delta_\epsilon = \{\delta \colon \|\delta\|_\infty \leq \epsilon\}$ and:

$$\delta^* = \underset{\delta \in \Delta_\epsilon}{\operatorname{argmax}}\, \ell(h_\theta(x+\delta), y)$$

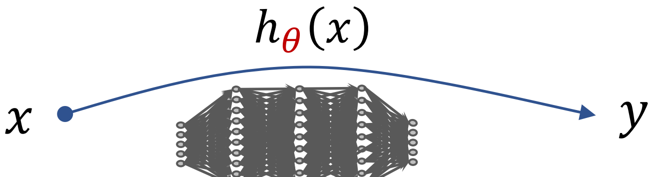
- Supervised training: let $(x, y) \sim P_{\mathcal{X}\times\mathcal{Y}}$,
  - $\mathrm{CE}(h_\theta(x), y) = \mathrm{CE}(h_\theta(x), [0, \ldots, 1, \ldots, 0]\} = -\log h_\theta^y(x)$
  - Individual loss: $\ell_{x,y}(\theta) = \ell_x(\theta, y) = \mathrm{CE}(h_\theta(x), y)$
  - Loss objective: $\ell(\theta) = \underset{(x,y)\sim P}{\mathbb{E}}\left[\ell_{x,y}(\theta)\right]$

- AT-PGD learning loss:
  - Let x' be adversarial sample of x via PGD:

$$\ell_{x,y}^{\mathrm{pgd}}(\theta) = \ell_{x,y}(\theta) + \beta \sup_{x'} \ell_{x',y}(\theta)$$
$$= \mathrm{CE}(h_\theta(x), y) + \beta \sup_{x' \in \mathcal{B}_\epsilon(x)} \mathrm{CE}(h_\theta(x'), y)$$



Input rate η and number of steps k:
- $x_0 = x + \mathrm{unifom}(-\epsilon, \epsilon)$
- $\tilde{x}_t = x_{t-1} + \eta \nabla_x \ell(h(x), y)|_{x_{t-1}}$
- $x_t = \mathrm{Proj}_{B_\epsilon(x)}(\tilde{x}_t)$
- Run for k steps, then set $x' = x_k$



*Madry et. al., Towards Deep Learning Models Resistant to Adversarial Attacks, *ICLR*, 2018.

Three SOTA AT approaches

- AT-PGD learning objective (Madry, et al, 2019):
  - PGD-AT loss: $\ell_{x,y}^{\text{pgd}}(\theta) = \text{CE}(h_\theta(x), y) + \beta \sup_{x' \in \mathcal{B}_\epsilon(x)} \text{CE}(h_\theta(x'), y)$
  - Learning objective: $\theta^* = \underset{\theta}{\text{argmin}}\ \mathbb{E}_{\mathbb{P}}[\ell_{x,y}^{\text{PGD}}(\theta)]$, i.e,

$$\inf_\theta \mathbb{E}_P \left[\text{CE}(h_\theta(x), y) + \beta \underbrace{\sup_{x' \in \mathcal{B}_\epsilon(x)} \text{CE}(h_\theta(x'), y)}_{\text{mitigate worst-case}}\right]$$

- AT-TRADES (Zhang et. al, 2019)

$$\inf_\theta \mathbb{E}_P \left[\overbrace{\text{CE}(h_\theta(x), y) + \beta \underbrace{\sup_{x'} \text{D}_{KL}\big(h_\theta(x'), h_\theta(x)\big)}_{\text{maximise diversity}}}^{\ell_{x,y}^{\text{trades}}(\theta)}\right]$$

- AT-MART (Wang et al., 2019):
  - Define $\text{BCE}(h_\theta(x), y) = -\log h_\theta^y(x) - \log\left(1 - \max_{k \neq y} h_\theta^y(x)\right)$
  - Extend TRADES to take into account the prediction confidence

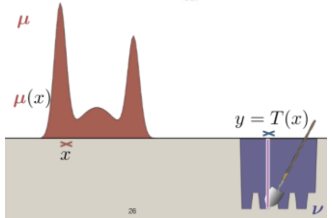
$$\inf_\theta \mathbb{E}_P \left[\overbrace{\text{BCE}(h_\theta(x), y) + \beta\big(1 - h_\theta^y(x)\big) \sup_{x'} \text{D}_{KL}\big(h_\theta(x'), h_\theta(x)\big)}^{\ell_{x,y}^{\text{mart}}(\theta)}\right]$$

Wasserstein and Optimal Transport (OT)

A (very) brief history

1781 — 150 years later — 1975 — 40 years later — 2010 — 2013 — 2017 — 2018 — OT4ML took off

G. Monge



Given μ, ν, find T s.t.

- $T_{\#}\mu = \nu$: its minimal cost
- T: (optimal) transport map

$$\inf_{T: T_{\#}\mu = \nu} \int_{\mathcal{X}} c(x, T(x)) \mathrm{d}\mu(x)$$

Dual formulation

Now computational friendly

$$W_1 = \sup_{f+g \leq c, f, g \in \mathcal{L}_1} \left\{ \mathbb{E}_{\mu}[f(x)] + \mathbb{E}_{\nu}[g(y)] \right\}$$

L. Kantorovich

(Nobel prize, economics)

Define coupling Π whose marginals are μ and ν

$$\pi^* = \inf_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\pi$$

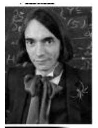
π^*: (optimal) transport plan

Wasserstein distance

$$W_p = \inf_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} [\|x - y\|^p]^{1/p} \, \mathrm{d}\pi$$

V. Villani

Field Medal



2013: Sinkhorn (NeurIPS'13)

2017: Wasserstein GAN (ICML'17)

Ours: WMeans (ICML'17, JMLR'21)

A. Figalli

Field Medal

- Possess a different geometry from standard divergences such KL or Euclidean

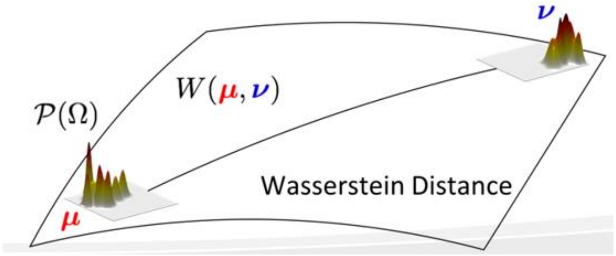


Image credit: Cuturi., *Computational Optimal Transport*, NeuRIPS 2021.

Wasserstein Risk Minimization (WRM)

- Distributional Robustness (Blanchet et al' 19) DRO = optimisation + stats
- General setting:
  - Let $v \sim P$ on metric space S
  - $f(v): S \to \mathbb{R}$ is a risk/reward function
  - Seek Q on S such that:

$$\sup_{Q} \mathbb{E}_{Q:\mathrm{dist}(Q,P)<\epsilon} [f(v)]$$

- **Key result**: if Wasserstein distance is used, then:

$$\sup_{Q:W_c(Q,P)<\epsilon} \mathbb{E}_{Q}[f(v)]$$

is equivalent to

$$\inf_{\lambda \geq 0} \left\{ \lambda\epsilon + \mathbb{E}_{v \sim P} \left[\sup_{v'} \left(f(v') - \lambda c(v, v') \right) \right] \right\}$$

Blanchet et al., Quantifying distributional model risk via optimal transport, Mathematics of Operations Research, 2019.

- WRM (Sinha et al'18) = DRO + ML
  - Consider a typical supervised setting:

$$\mathcal{X} \ni x \xrightarrow{h_\theta(x)} y \in \mathcal{Y}$$

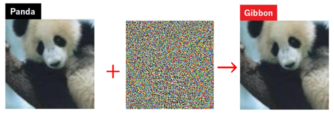
  - Now let $S = \mathcal{X} \times \mathcal{Y}$ and $v = (x, y)$, $v' = (x', y')$ on S
  - Define metric: $c(v, v') = d(x, x') + \infty \times \mathbb{I}_{[y \neq y']}$
  - And risk: $f(v) = \ell_{x,y}(\theta) = \ell(h_\theta(x), y)$
  - Then learning θ under DRO becomes (WRM)

$$\inf_{\theta} \sup_{Q:W_c(Q,P)<\epsilon} \mathbb{E}_{Q} [\ell(h_\theta(x), y)]$$

Sinha et al., Certifying Some Distributional Robustness with Principled Adversarial Training, *ICLR 2018*.

From AT to Distributional AT

- Recall: standard AT looking for pairwise (x, x') to improve robustness.
  - e.g., for PGD:



$$\inf_{\theta} \mathbb{E}_{P} \underbrace{\left[\mathrm{CE}(h_\theta(x), y) + \beta \sup_{x' \in \mathcal{B}_\epsilon(x)} \mathrm{CE}(h_\theta(x'), y)\right]}_{\ell_{x,y}^{\mathrm{pgd}}(\theta)}$$

- DRO/WRM looks for the entire adversarial distribution Q in the vicinity of data distribution P, i.e.,

$$\inf_{\theta} \sup_{Q:W(Q,P)\leq\epsilon} \mathbb{E}_{Q} \left[\ell(h_\theta(x), y)\right]$$

Is there a theoretical tool to provide a connection between them?

- First attempt using WRM for AT-PGD:
  - $S = \mathcal{X} \times \mathcal{Y}, \mathrm{c}(v, v') = d(x, x') + \infty \times \mathbb{I}_{[y \neq y']}$
  - Let $f(v) = f(x, y) = \ell_{x,y}^{\mathrm{pgd}}(\theta)$, WRM becomes:

$$\inf_{\theta} \sup_{Q:W_c(Q,P)<\epsilon} \mathbb{E}_{Q} \left[\ell(h_\theta(x), y)\right]$$

- Not quite, but almost, by letting $\epsilon \to 0$.
- And fail to solve for more complex AT methods, such as ℓ_x^{trades} and ℓ_x^{mart}

Our Unified Distribution Robustness (UDR)

Bui, et. al, ICLR 2022


Tony Bui


Dr Trung Le

- Solution sketch:
  - Let $S = \mathcal{X} \times \mathcal{X} \times \mathcal{Y}$:
    - space of x, space of its adversarial x' and output
  - Use $p(x,y) = p(y|x)p(x)$, write $P_{\mathcal{X}\times\mathcal{Y}} = P_{\mathcal{X}} \times P_{\cdot|\mathcal{X}}$
  - Denote P^* the distribution over specific configuration (x, x, y) where $x \sim P_{\mathcal{X}}$ and $y \sim P_{\cdot|\mathcal{X}}$.
  - P^* is a distribution on S, let seek Q on S such that $W_{c^*}(Q, P^*) < \epsilon$.
    - Let $v = (x, x, y) \sim P^*$ and $v' = (x', x'', y') \sim Q$, metric $c^*(\cdot)$ deliberately designed:
    - $c^*(v, v') = d(x, x') + \infty \times d(x, x'') + \infty \times \mathbb{I}_{[y = y']}$
    - $c^*(v, v') < \infty$, then $x'' = x, y' = y$ and $x' \to x$
  - Define a unified risk function $g_\theta(v')$ for UDR-PGD, URD-TRADES and URD-MART respectively:

$$= \begin{cases} \mathrm{CE}(h_\theta(x''), y') + \beta \cancel{\sup_{x' \in \mathcal{B}_\epsilon(x)}} \mathrm{CE}(h_\theta(x'), y') \\ \mathrm{CE}(h_\theta(x''), y') + \beta \mathrm{D}_{KL}\big(h_\theta(x'), h_\theta(x'')\big) \\ \mathrm{BCE}(h_\theta(x''), y') + \beta\big(1 - h_\theta^y(x'')\big)\mathrm{D}_{KL}\big(h_\theta(x'), h_\theta(x'')\big) \end{cases}$$

- Key results:
  - The primal DRO $\inf_\theta \sup_{Q: W_c(Q, P^*) < \epsilon} \mathbb{E}_Q[g_\theta(v')]$ becomes

$$\inf_{\theta, \lambda \geq 0} \left\{ \lambda\epsilon + \mathbb{E}_{v \sim P^*}\left[\sup_{v'} \left(g_\theta(v') - \lambda c^*(v, v') \right) \right] \right\}$$

  - With specific $c^*(v, v')$, this is the same as

$$\inf_{\theta, \lambda \geq 0} \left\{ \lambda\epsilon + \mathbb{E}_{x \sim P}\left[\sup_{x' \in \mathcal{X}} \left(g_\theta(x', x, y) - \lambda d(x, x') \right) \right] \right\}$$

  - Theorem: let $d^*(x, x') = d(x, x')$ if $x' \in \mathcal{B}_\epsilon^d(x)$ and ∞ otherwise, then:

$$\inf_{\theta, \lambda \geq 0} \left\{ \lambda\epsilon + \mathbb{E}_{x \sim P}\left[\sup_{x' \in \mathcal{X}} \left(g_\theta(x', x, y) - \lambda d^*(x, x') \right) \right] \right\}$$

    is equivalent to pointwise objective:

$$\inf_\theta \mathbb{E}_P \left[\sup_{x' \in \mathcal{B}_\epsilon(x)} g_\theta(x', x, y) \right]$$

  - Claims:
    - AT-method are special cases of UDR-method
    - Richer expressive capacity
    - Substantially different from WRM (Shina etal '18, Blanchet & Murphy '19)

Learning with UDR

Bui, et. al, ICLR 2022

- Note $d^*(x, x')$ is non-differentiable outside the ball $\mathcal{B}_\epsilon(x)$, define a smoothed version $\hat{d}(x, x')$:

$$d(x, x')\mathbb{I}_{[d(x,x')<\epsilon]} + \left(\epsilon + \frac{d(x, x') - \epsilon}{\tau}\right)\mathbb{I}_{[d(x,x')\geq\epsilon]}$$

- Final optimisation form:

$$\inf_{\theta,\lambda\geq 0}\left\{\lambda\epsilon + \mathbb{E}_{x\sim P}\left[\sup_{x'\in\mathcal{X}}\left(g_\theta(x', x, y) - \lambda\hat{d}(x, x')\right)\right]\right\}$$

2 3 1

Algorithm - UDR

1. For each (x_i, y_i) learn adversarial sample:

$$x_i^{\text{adv}} = \operatorname*{argmax}_{x'}\left\{g_\theta(x', x_i, y_i) - \lambda\hat{d}(x_i, x')\right\}$$

2. Update parameter λ (take derivative, set to 0):

$$\lambda_l = \lambda_{l-1} - \eta_\lambda\left(\epsilon - \frac{1}{N}\sum_i \hat{d}(x_i^{\text{adv}}, x_i)\right)$$

3. Update model parameter θ:

$$\theta_l = \theta_{l-1} - \frac{\eta_\theta}{N}\sum_i^N \nabla g_\theta\left(x_i^{\text{adv}}, x_i, y_i\right)\Big|_{\theta_{l-1}}$$

PGD

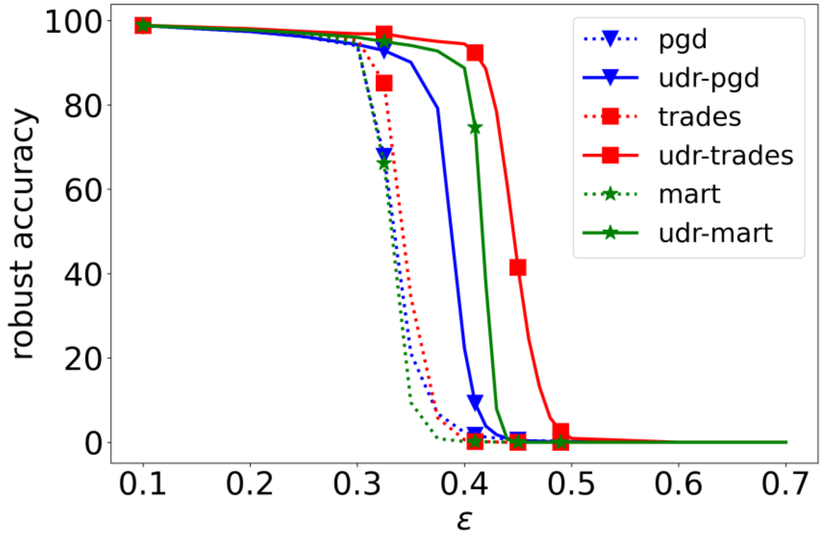
Input rate η and number of steps k:

- $x_0 = x + \text{unifom}(-\epsilon, \epsilon)$
- $\tilde{x}_t = x_{t-1} + \eta\nabla_x \ell(h(x), y)|_{x_{t-1}}$
- $x_t = \text{Proj}_{B_\epsilon(x)}(\tilde{x}_t)$
- Run for k steps, then set $x' = x_k$

Our Unified Distribution Robustness (UDR)

Bui, et. al, ICLR 2022

- Key experimental results
  - UDR-methods outperform in Whitebox Attack with fixed ϵ
  - Methods can extend beyond PGD, TRADES, MART, but also new methods, e.g., Auto-Attack and so on.
  - Consistent performance against various attack strength (e.g., varying ϵ)



	MNIST				CIFAR10				CIFAR100			
	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B
PGD-AT	99.4	94.0	88.9	91.3	**86.4**	46.0	42.5	44.2	72.4	41.7	39.3	39.6
UDR-PGD	**99.5**	**94.3**	**90.0**	**91.4**	**86.4**	**48.9**	**44.8**	**46.0**	**73.5**	**45.1**	**41.9**	**42.3**
TRADES	**99.4**	95.1	90.9	92.2	80.8	51.9	49.1	50.2	68.1	49.7	46.7	47.2
UDR-TRADES	**99.4**	**96.9**	**92.2**	**95.2**	**84.4**	**53.6**	**49.9**	**51.0**	**69.6**	**49.9**	**47.8**	**48.7**
MART	**99.3**	94.7	90.6	92.9	**81.9**	53.3	48.2	49.3	**68.1**	49.8	44.8	45.4
UDR-MART	**99.3**	**96.0**	**92.3**	**94.4**	80.1	**54.1**	**49.1**	**50.4**	67.5	**52.0**	**48.5**	**48.6**

See our poster for more details and results

Code: https://github.com/tuananhbui89/Unified-Distributional-Robustness

Some concluding thoughts

- There is a surge of interests from since Goodfellow et al.’ ICLR’15
  - Most interesting aspect: expose the ‘mysterious’ mathematical behaviours in very complex functions in high-dimensional spaces.
    - Consequences: fragility of modern DNNs
  - What caused this? –not really know!
    - Consequence of regularisation, models to be too linear in the last layer, so easy to manipulate dot product in high-dimensional space to alter production (Goodfellow ICLR’15)
    - Images have predictive features which are invisible to human (Ilyas. NeurIPS’19); Gaussian artifact (Gilmer, ICML’19), Violation of data manifold hypothesis (CVPR’19)
- Not so good news: probably 70 – 75% at best for ImageNet
- But some good news: effective ‘tools’ to understand DL models
- What’s next?:
  - Trustworthy ML has a broader context and will be truly important !!!
  - Generative AI is on the rise, so what does it mean to have Trustworthy GenAI?

THANK YOU

dinh.phung@monash.edu

Acknowledgment:
Dr Paul Montague, Dr Tamas Abraham (DST)
Next Technology Generation Scheme (2018-)
Australia Research Council Discovery Project (2023-)

Want to know more ?

Robust/Trustworthy ML

- Anh Bui et al., Generating Adversarial Examples with Task Oriented Multi-Objective Optimization, TMLR, 2023.
- Anh Bui et al., A Unified Wasserstein Distributional Robustness Framework for Adversarial Training, *ICLR*, 2022.
- Trung Le et al., A Global Defense Approach via Adversarial Attack and Defense Risk Guaranteed Bounds, *AISTATS*, 2022.
- Thanh Nguyen-Duc et al., Particle-based Adversarial Local Distribution Regularization, *AISTATS*, 2022.
- Anh Bui et al., Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness, *AAAI*, 2021.
- Anh Bui et al., Improving Adversarial Robustness by Enforcing Local and Global Compactness, *ECCV*, 2020.
- EMNLP'20, AISTATS'20, …

Selected work on Optimal Transport for ML:

- Tutorial on "Optimal Transport", ACML 2021
- Two survey papers: IJCAI'21 (for Generative AI), IJCAI'21 (for topic models)
- ICML'23, AISTAT'23, ICASSP'23
- NeuRIPS'22, ICML'22, ICLR'22, UAI'22, AISTATS'22
- JMLR'21, NeurIPS'21, ICCV'21, ICML'21, IJCAI'21, UAI'21, ICLR'21, AAAI'21
- NeurIPS'20, , ICML'20, ECCV'20,
- ICLR'19, IJCAI'19, ICML'17

MONASH University