Prédiction photo-z: CNN face à l'adversité*

J.E Campagne
(IJCLab/Orsay)
LSST-France feb 2020

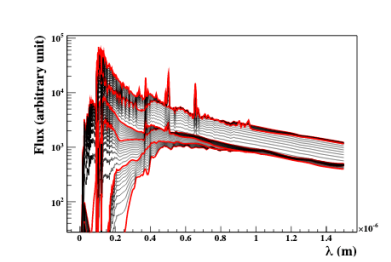
*: nothing to do with Trump ☺

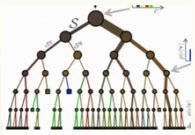
Outline

- Photo-z methods brief summary
- Case of Inception developed by J. Pasquet et al.
- Adversarial Samples
- Some results towards robustness
- Summary/Outlooks

Methods for Photo-z

Since the pioneering work in the 60's, several methods have been developed to estimate the redshift from the multi-bands photometric measurements, basically:





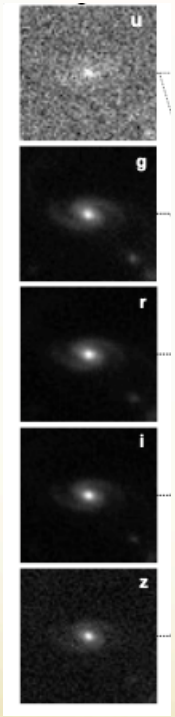
Possible combinaison

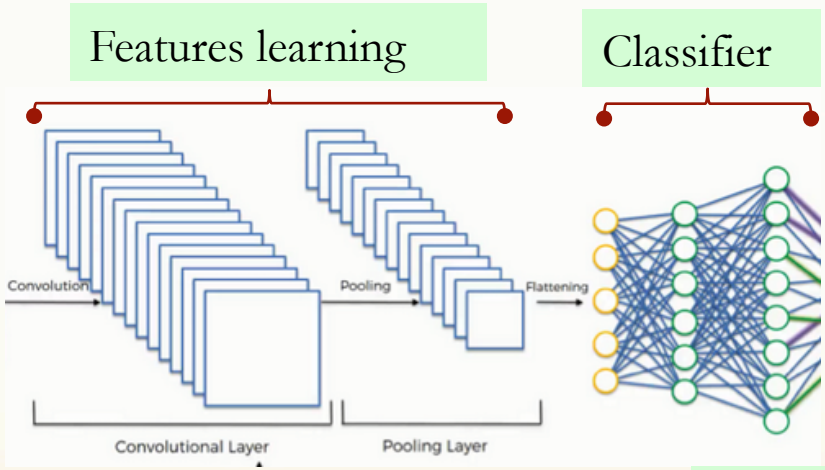
- template-fitting
  - Uses the SED and a method of fit
  - since Loh & Spillar 1986 ~30 galaxies in cluster 0024+1654,…, Beck et al 2016...
  - for LSST eg. Gorecki et al 2014 and Ansari et al 2019
- feature based Machine Learning
  - Uses a certain number of predefined features extracted from the measurements and feed to an engine as k-NN, NN/MLP, Decision Tree, BDT or Random Forest
  - Eg. Csabai et al. 2007 (k-NN) used by Beck et al 2016, Gorecki et al 2014 (NN) ,Ansari et al 2019 (BDT)...
- image based Deep Learning

Nb. Absolutely non exhaustive list of contributions.

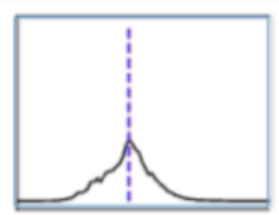
DL: what is promised?

Input images → Output Photo-z p.d.f





Cascade of Convolutional layers



Output neurons

Pasquet J., Bertin E., Treyer M., Arnouts S., Fouchez D., 2019, A&A, 621, A26
arXiv:1806.06607v2

1) Variation: D'Isanto & Polsterer (2018) with a Gaussian Mixture Model as output
2) CNN architecture is used in other context: eg. g-g lens finding algo (Lanusse et al 2018), deblending (Burke et al 2019), objects classification (Gonzales et al 2018),…
… Non exhaustive list !

« *Inception* » for photo-z

J. Pasquet et al. (2019)

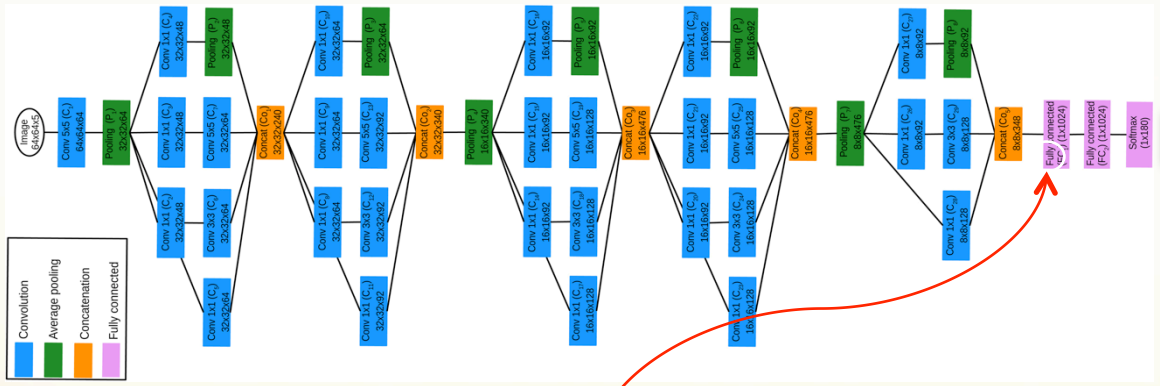
Inspired from GoogleLeNet with multi-levels of conv-layers (Szegedy et al. 2014)

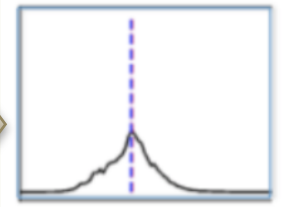
27.5M parameters

~30Layers

Convolution (6.5% param.)

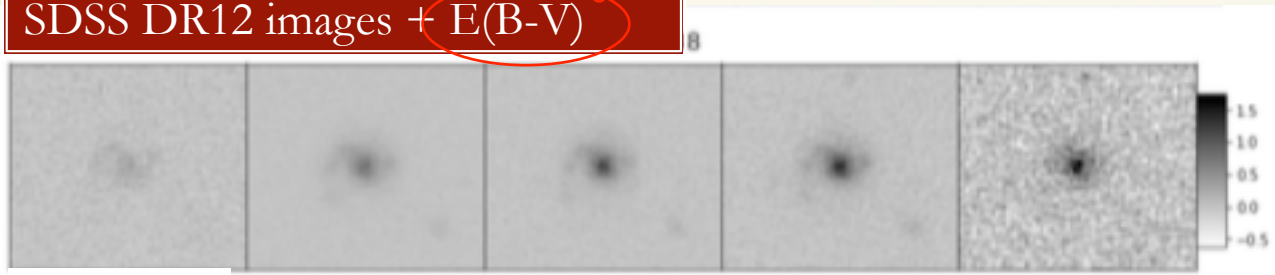
FC (93.5% param.)





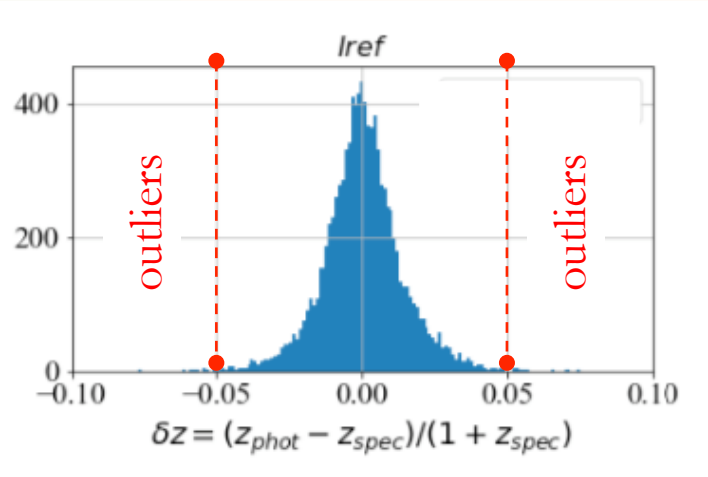
180 bins in z

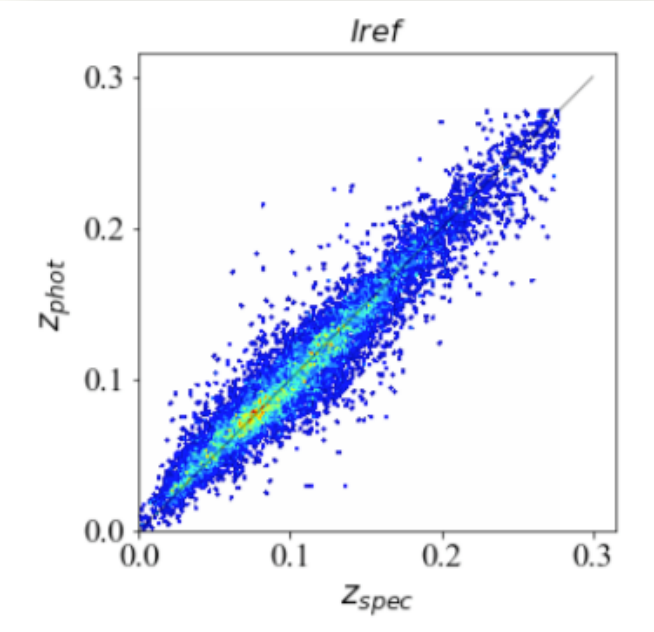
SDSS DR12 images + E(B-V)



5x64x64

Results from *Inception*





Training/Test samples :100k/100k
from a total of ~ 600k input dataset

I have refactored the original code in PyTorch latest 1.3.1 version and run @ CCIn2P3 GPU farm (mostly on V100)

- the bias defined as the mean of the δz distribution;
- the $\sigma_{\rm MAD} = 1.4826 \times |\delta z - {\rm Median}(\delta z)|$;
- and the fraction η of outliers such that $|\delta z| > 0.05$.

bias ($\times 10^{-4}$)	$\sigma_{\rm mad}$ ($\times 10^{-3}$)	η (%)
0.3	11	1

Results totally in agreement with
J. Pasquet et al. (2019)

Some syst. studies

These are a very short summary of the J. Pasquet et al. thorough study.

Item	Comments
Galactic reddening (extra features added at the level of the FC part)	a strong reddening-dependent bias is observed If the information is not provided
Galaxy inclination	the CNN is very robust: large sample & data augmentation
Neighboring galaxies	The CNN learn how to improve redshift with neightboors at z>0.1
Variations throughout the surveyed area	Deviations in the SZ and Strip 82 of the SDSS dataset
PSF	induce a small but measurable amount of systematics on the estimated redshifts. Info can be added at the FC input (not done).

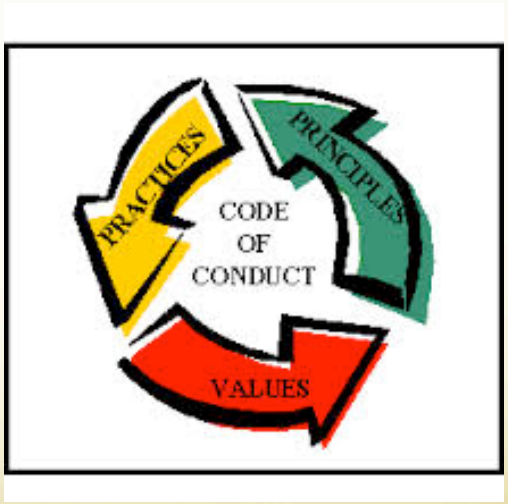
So far so good!

Teams have spend some times to:

- Elaborate ML/DL architectures
- Apply some ML paradigm to tune the hyper-parameters using for instance: the triptych Training/Testing/Validation sets, Under/Over fitting aspects
- Compare their results against "State-of-the-art" competitors
- Perform systematics studies on the Input Data: eg. are they representative of the use-case, what about their quality…

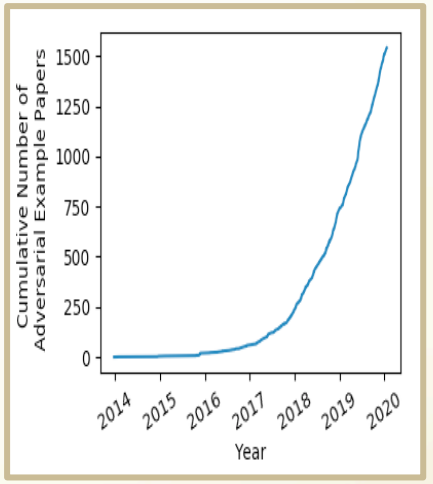
But, haven't we forgotten something ?

WARNING: The following slides contain images that may be disturbing to some readers.



Adversarial samples: brief history

- After "AlexNet" the winner of ImageNet competition 2012
- Topic rising since Szegedy et al. (2013): "**Intriguing properties of neural networks**"
- 1st explanation Goodfellow et al. (2014) : "**Explaining and Harnessing Adversarial Examples**"
- Part of the NIPS '17 Competition
- Kurakin et al @ ICRL 17: "**Adversarial Machine Learning at Scale**"
- Ilyas et al (2019): "**Adversarial Examples Are Not Bugs, They Are Features**"
- Madry et al (2017-19) @ ICLR 18: "**Towards Deep Learning Models Resistant to Adversarial Attacks**"
- **… Towards a deeper understanding of what is going on and how to overtake this intrinsic problem.**



NSIP: Neural Information Processing Systems
ICLR: International Conference on Learning Representations

Empirical risk/adversarial sample

$\{x_i, z_i\}_{i \leq N} \in D_{train}$ Eg. x_i: images, z_i: spectro-z

Classical Empirical risk

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left\{ \frac{1}{|D_{train}|} \sum_{(x,z) \sim D_{train}} \ell(f_\theta(x), z) \right\}$$

Adversarial perturbation

$$\delta^* = \max_{\|\delta\| \leq \varepsilon} \ell(f_\theta(x + \delta), z)$$

1) Min-max/saddle point problem: no general solution in non-convex problem
2) Which norm $||.||$, which value of ε ?

One-step perturbation

$$\delta^*(x) \equiv \underset{\|\delta\| \leq \varepsilon}{\text{argmax}}\ \ell(f_\theta(x+\delta), z)$$

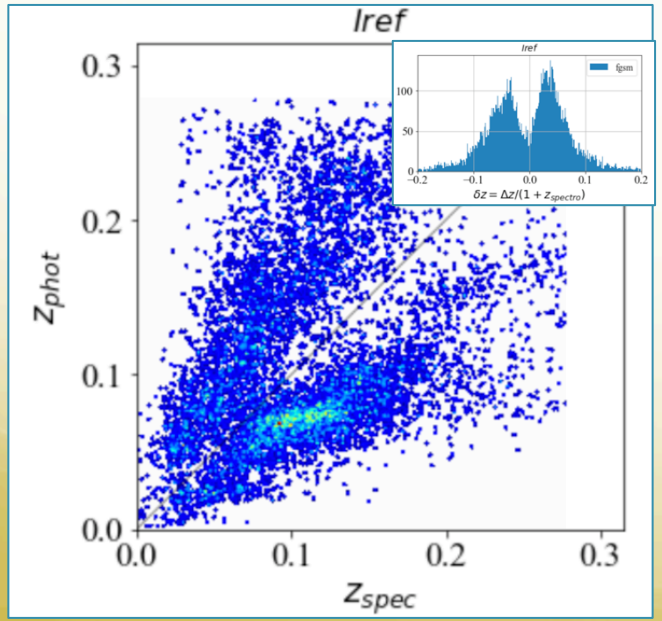
Fast Sign Gradient Method

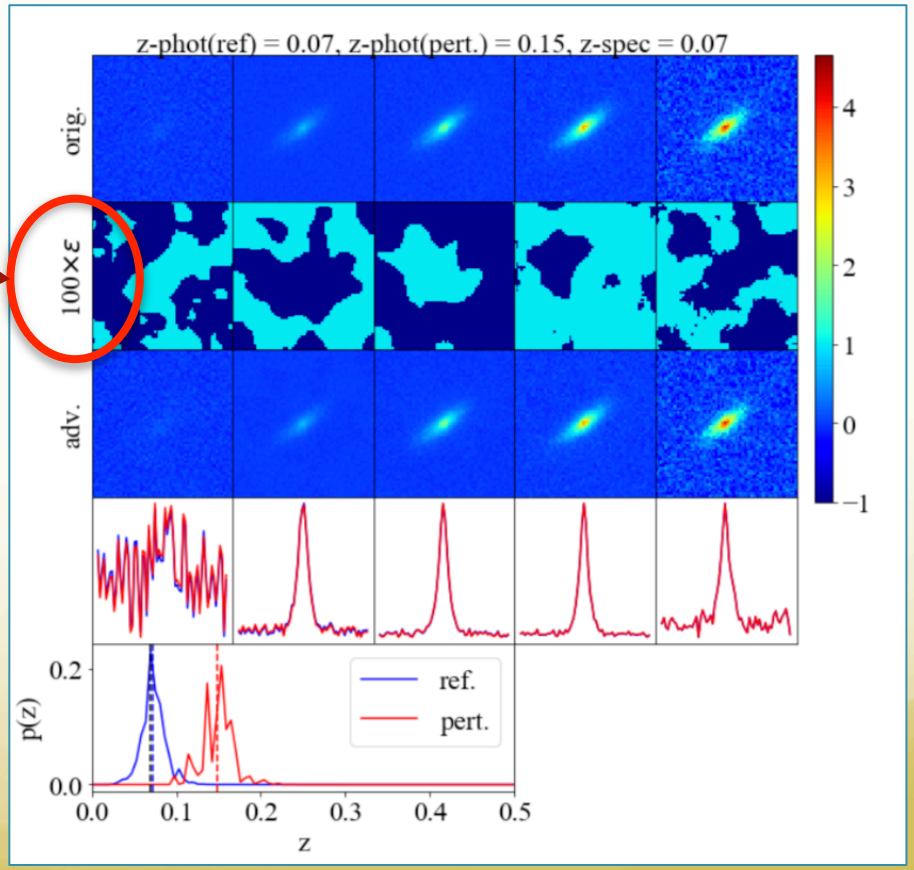
$$f_\theta \text{ "linear"} + \|\delta\|_\infty \leq \varepsilon \Rightarrow \delta^*(x) = \varepsilon \times \text{sign}\left(\nabla_\delta \ell(f_\theta(x+\delta), z)\right)$$

Goodfellow et al. 2014

$\varepsilon = 10^{-2}$

(no effect with random noise)





Multi-steps perturbations

$$\delta^*(x) \equiv \underset{\|\delta\| \leq \varepsilon}{\operatorname{argmax}}\ \ell(f_\theta(x+\delta), z)$$

Non-linear case

$$\delta \leftarrow \delta + \underset{\|u\| \leq \alpha}{\operatorname{argmax}} \left[u^T . \nabla_\delta \ell(f_\theta(x+\delta), z) \right]$$

Kurakin et al 2016

$$\|\delta\|_\infty \leq \varepsilon$$

Projected Gradient Descend

$$\delta \leftarrow \mathcal{P}_\varepsilon^\infty \left[\delta + \alpha\ \operatorname{sign}(\nabla_\delta \ell(f_\theta(x+\delta), z)) \right]$$

clip learning rate

One can also use GAN
Jang et al. 2019
Ensemble Adv. Training
Tramer et al. 2018

Different Inception networks and CNN architectures trained independently and tested with the same adversarial samples.

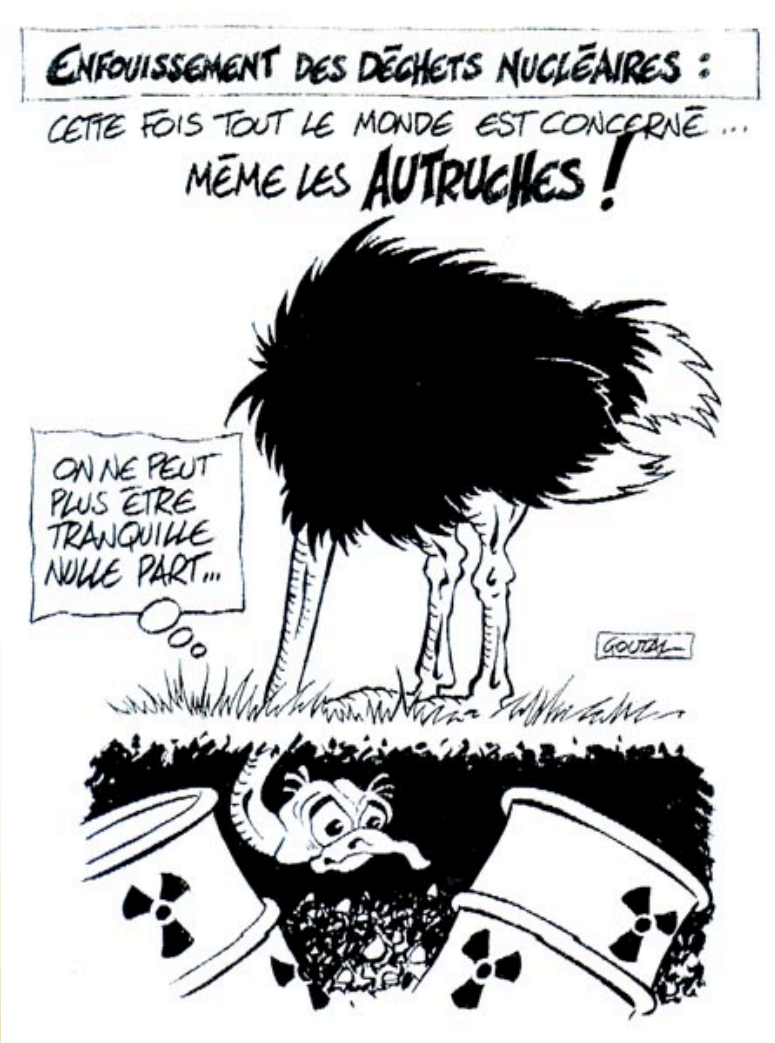
Models (images)	bias ($\times 10^{-4}$)	σ_{mad} ($\times 10^{-3}$)	η (%)
Iref (non perturbed)	0.3	11	1
$\varepsilon = 10^{-2}$, Single FSGM			
Iref	–	66	42
I0-I3 (*Iref* adv)	–	[63, 68]	[40, 43]
CNN (*Iref* adv)	–	76	49
$\varepsilon = 10^{-2}$, PGD, $\alpha = 10^{-3}$, $n_{iter} = 10$			
Iref	–	82	59
I0-I3 (*Iref* adv)	–	[76, 78]	[52, 55]

What to do?

- What's the problem?

What to do?

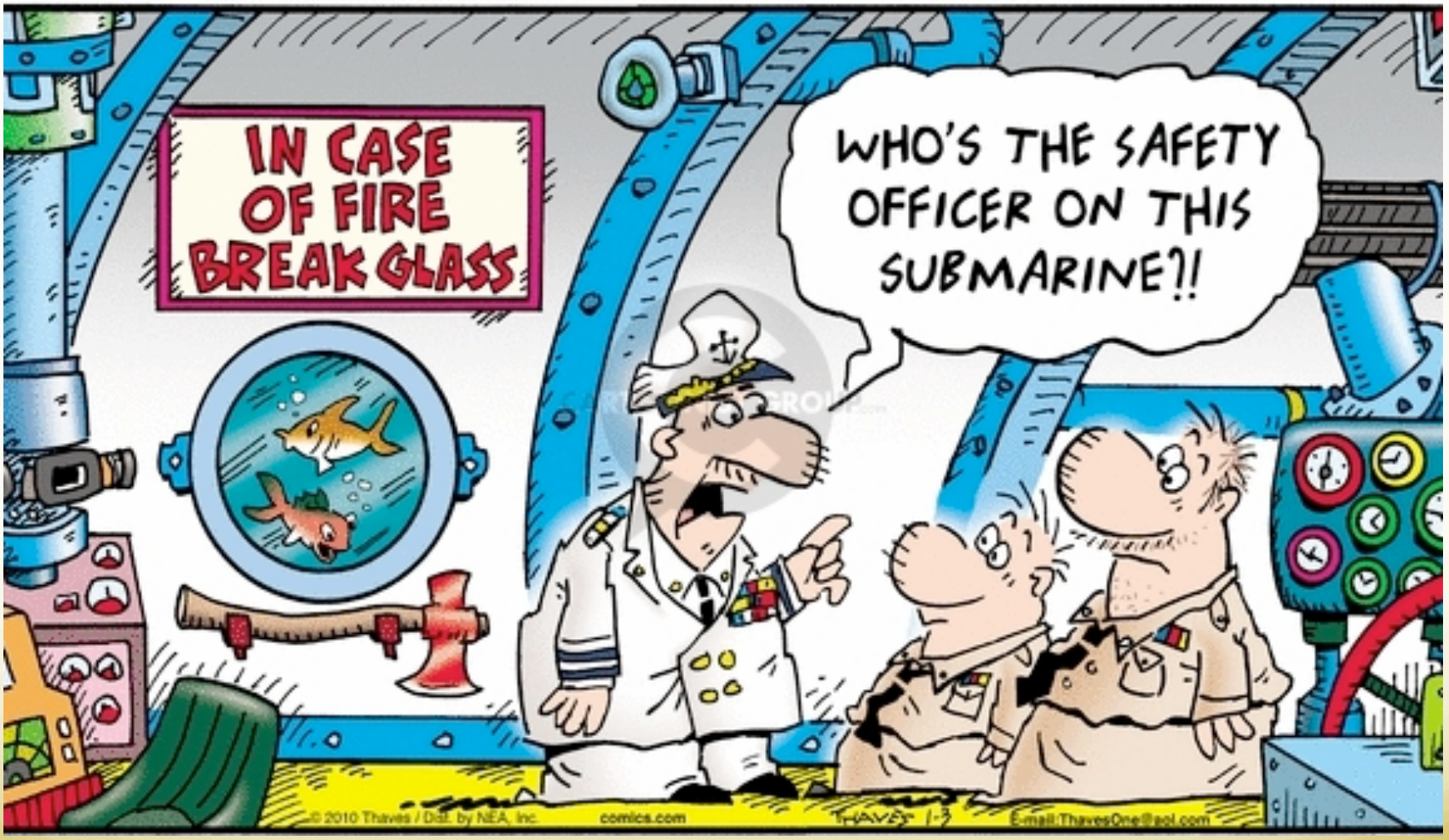
- What's the problem?
- Bury one's head in the sand…



What to do?

- What's the problem ?
- Bury one's head in the sand…
- « These kind of perturbations will never append ! »: are you sure ?
- Take it seriously as a sign of a certain (intrinsic) weakness:
  - Training ?
  - Architecture ?
  - Both ?

Countermeasures ?



Countermeasures ?

What about the training?

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{|B_{train}|} \sum_{(x,z)\sim B_{train}} \nabla_\theta \left[\max_{\|\delta\|\leq\varepsilon} \ell(f_\theta(x+\delta), z) \right]_{\theta=\theta_t}$$

1) Solution not known in the general case.

$$\nabla_\theta \, \ell(f_\theta(x+\delta^*), z)$$

J. Danskin 1966
Convex case

2) Mix up normal images & adversarial ones acts as regularisation terms.

Finlay et al. 2018;
Bietti et al. 2018

Adversarial training/results

Mix up Normal/Adv.

1: Choose adversarial samples fraction and the attack generator (FSGM/PGD, ε, α, number of iterations)
2: Do θ (model weights) initialisation
3: **for all** mini-batch B_{train} **do**
4: $g \leftarrow 0$ ▷ loss gradient w.r.t θ
5: **for all** $(x, z) \in B_{train}$ **do**
6: **if** x counts for an adversarial sample **then**, according to initial generator choice, find δ^*:
7: $\delta^* \leftarrow \varepsilon \operatorname{sign}(\nabla_\delta \ell(f_\theta(x+\delta), z))]$ ▷ (FSGM)
8: $\delta \leftarrow \mathcal{P}_\varepsilon^\infty [\delta + \alpha \operatorname{sign}(\nabla_\delta \ell(f_\theta(x+\delta), z))]$ ▷ (PGD)
9: **else**
10: $\delta^* \leftarrow 0$
11: $g \leftarrow g + \nabla_\theta \ell(f_\theta(x+\delta^*), z)$ ▷ Update loss gradient
12: $\theta \leftarrow \theta - \alpha \frac{g}{|B_{train}|}$ ▷ Update model weights

There exist several alternatives & improvements

After adv-training (FSGM): results on "un-perturbed / perturbed" images

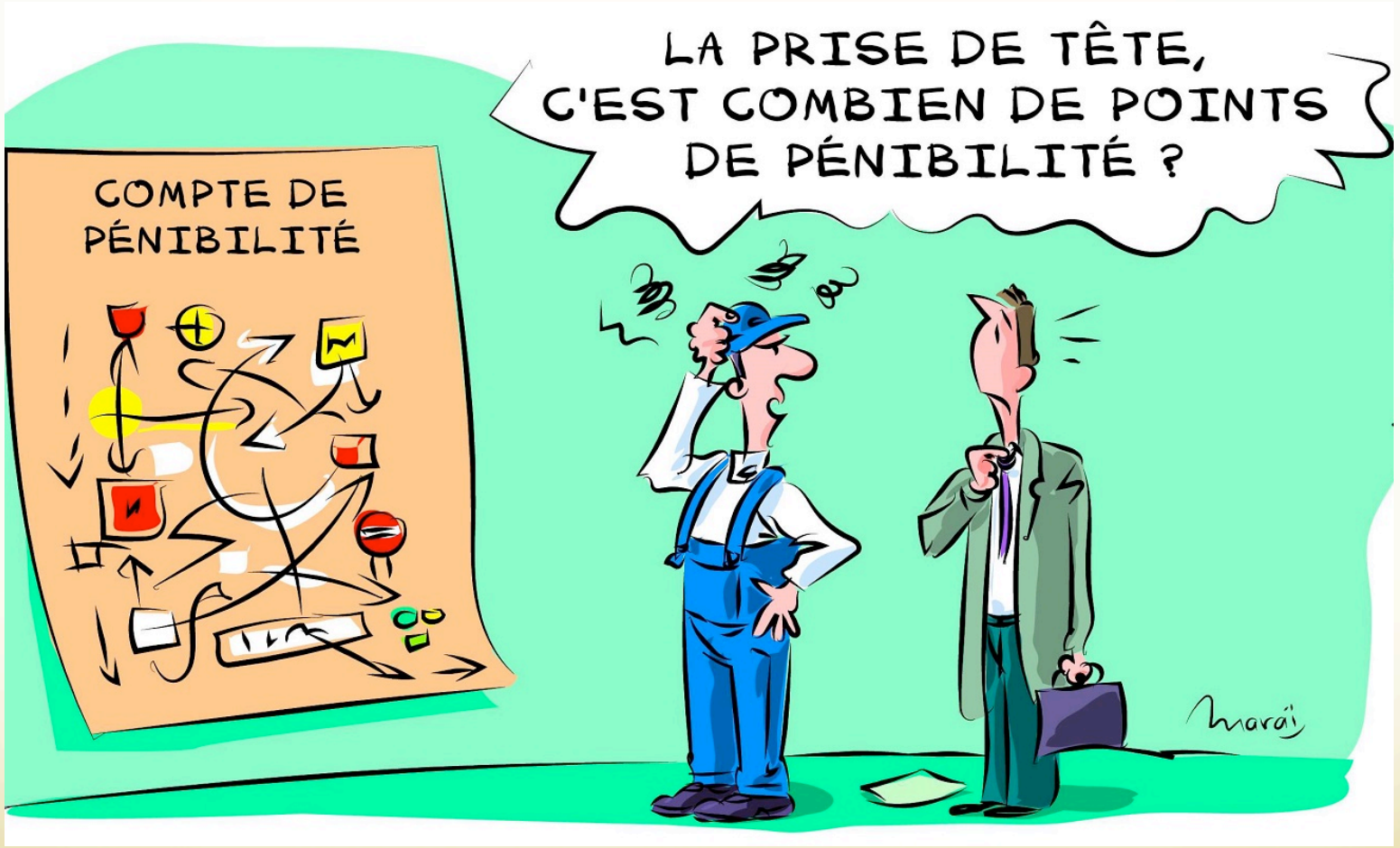
fraction of adv.	bias ($\times 10^{-4}$)	σ_{mad} ($\times 10^{-3}$)	η (%)
0%	−0.3/−105	11/66	1/42
5%	−20/−40	11/9	1/4
10%	40/−25	11/8	1/2
20%	−6/23	11/8	1/1
Iref (non perturbed)	0.3	11	1

We retrieve the un-perturbed classical result.

Summary/outlooks

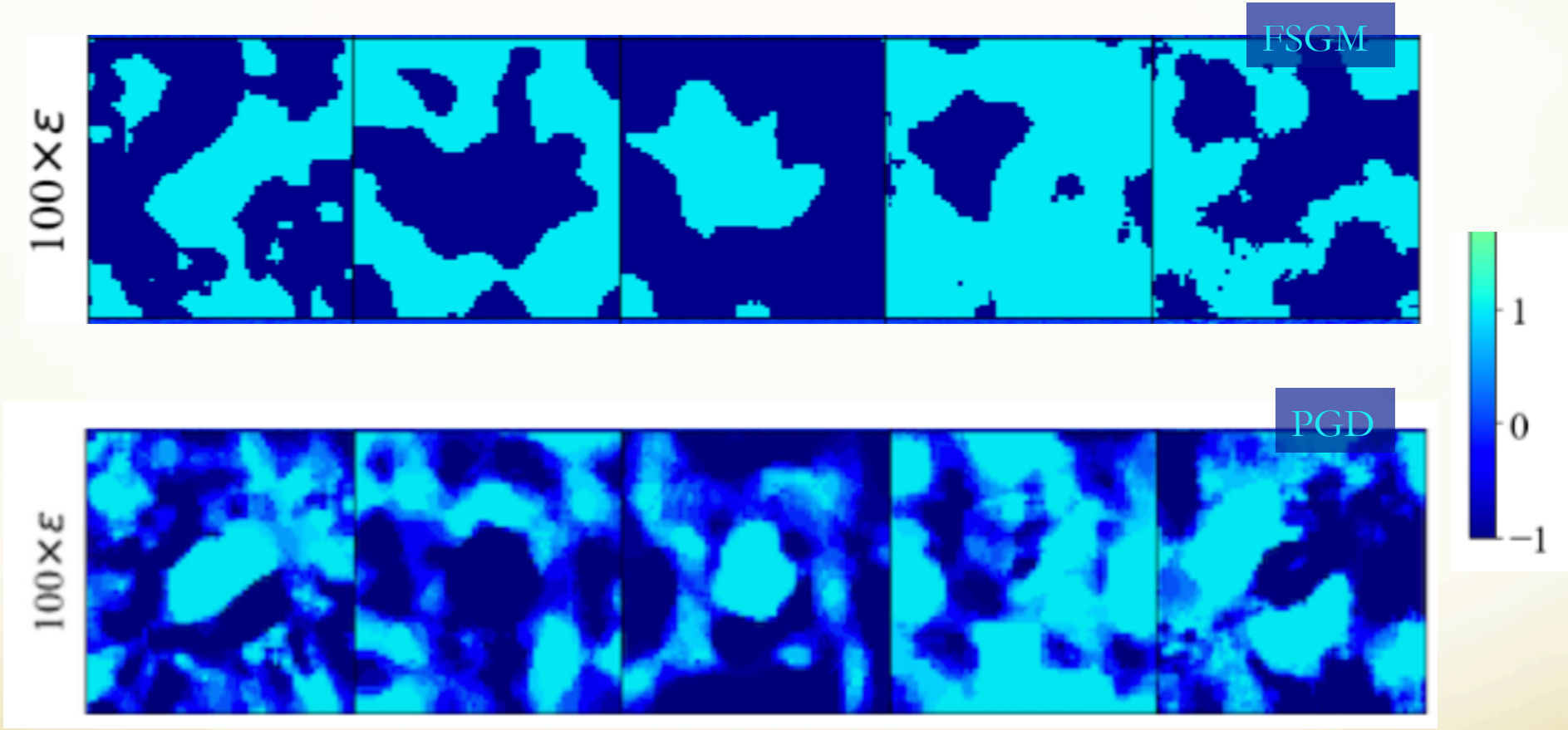
- The **classical** training/testing/validation triptych is **not enough to guarantee the generalisation power** of a network. Notice that **the problem in more general** than CNN (ie. DT, Gradient Boosted DT, R may also be affected as described in reference (Chen et al. 2019)).
- Some countermeasures have been elaborated but still it is a very active research domain as no satisfactory solution exists yet
- I've shown that mix up normal images with FSGM perturbed images gives some good results for Inception **robustness**
- But this is **not the end of the story**: **Inception is not immune** against more aggressive perturbations (eg. PGD) even if one uses above method and increases the capacity of the network
- So, what next?
  - **compare with other architectures**: eg. throw loss surface sensitivity against input modifications (Yu et al. 2018)..
  - **Change the training method** : eg. Lipschitz regularisation (Finlay et al. 2018), Kernel perspective for regularization (Bietti est al 2019)…

I've submitted a paper to MNRAS "Adversarial training applied to Convolutional Neural Network for photometric redshift predictions". Stay tuned.


COMPTE DE PÉNIBILITÉ
LA PRISE DE TÊTE, C'EST COMBIEN DE POINTS DE PÉNIBILITÉ ?

French joke

Back-up


FSGM
100×ε
PGD
100×ε
1
0
−1

If we train the Inception model against FSGM perturbation, it has no power against PGD perturbations.

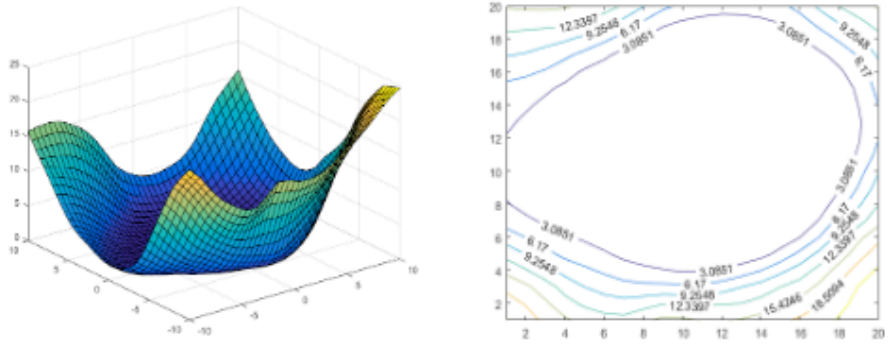
If we increase the #features at the input of the classifier part of Inception with fa= 50% we gain in robustness but wo recovering the classical training with no perturbed images.

Model	bias ($\times 10^{-4}$)	σ_{mad} ($\times 10^{-3}$)	η (%)
Iref	0.3/–/–	11/66/82	1/42/59
I(modified)	−21/ − 32/ − 32	15/24/25	2/6/6

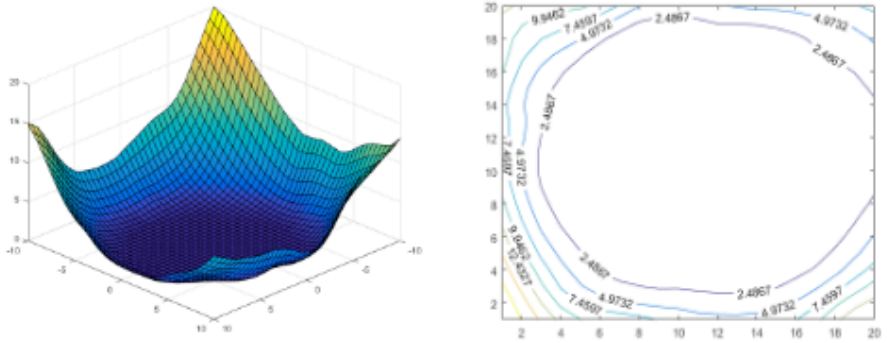
Test inputs: Non-perturbed/FSGM/PGD

Loss in Parameter space

No real difference between Robust training or not

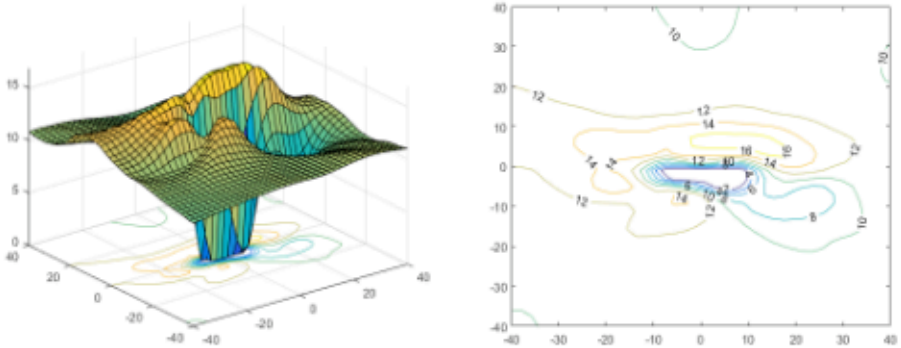


(a) Loss Surface of Natural Model

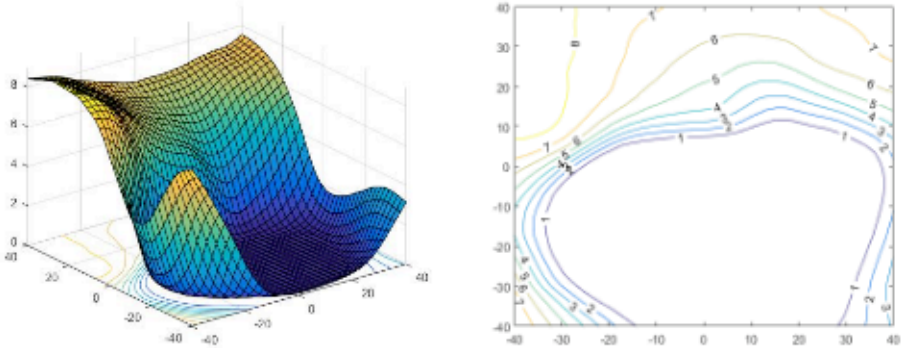


(b) Loss Surface of Robust Model

Loss in **Input space**

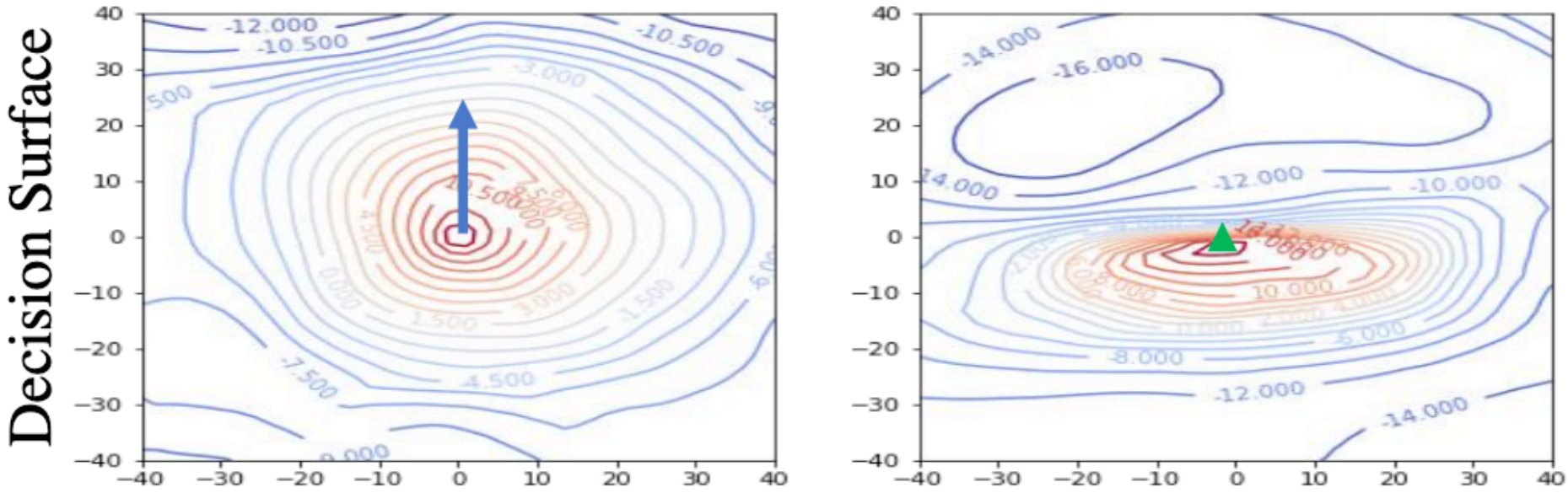


(a) Loss Surface of Natural Model



(b) Loss Surface of Robust Model

Why random noise is ok while adv-pertubation is efficient ?



(a) Random Direction

(b) Cross Entropy Loss

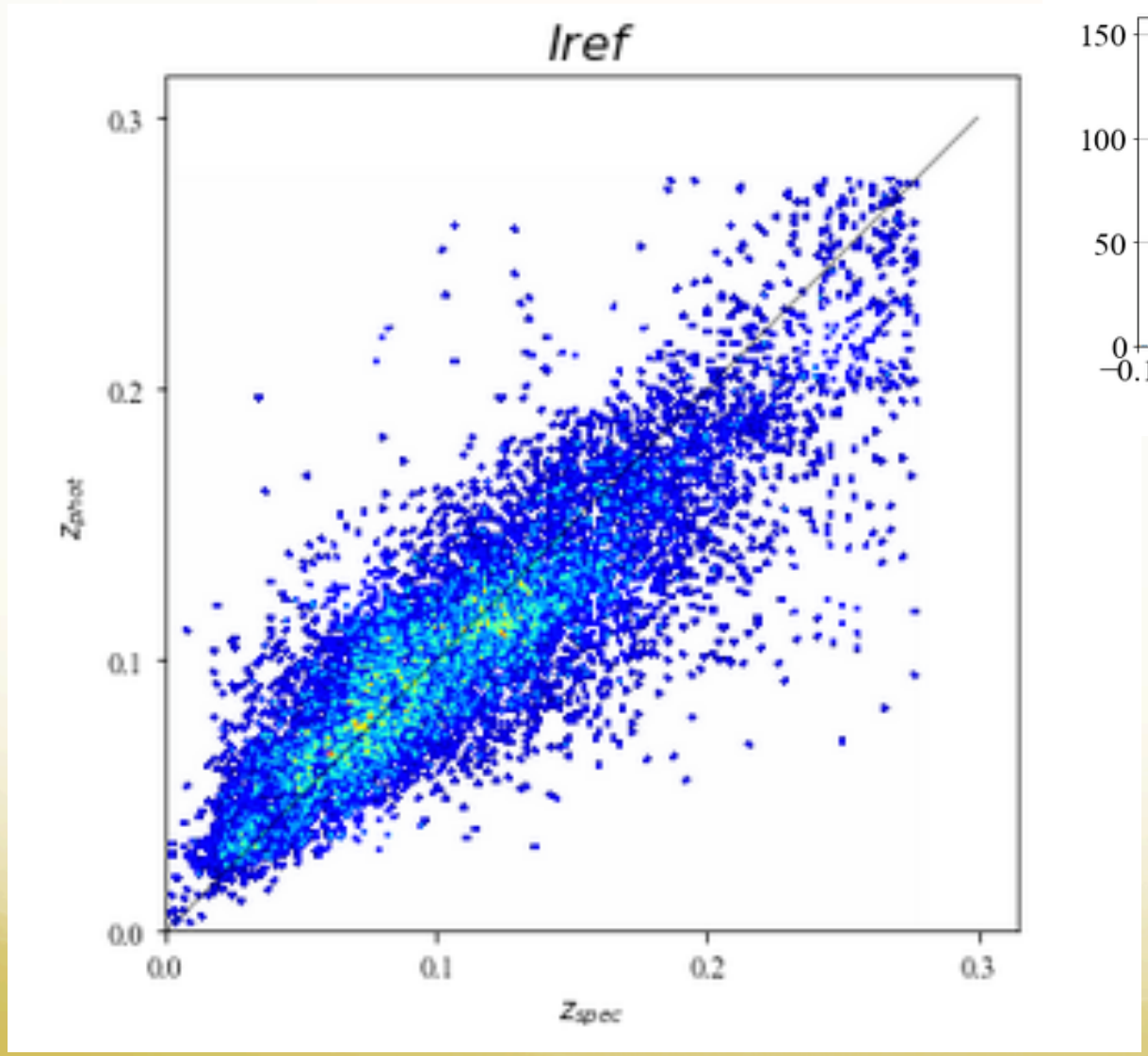
Fuxun Yu et al. 2019

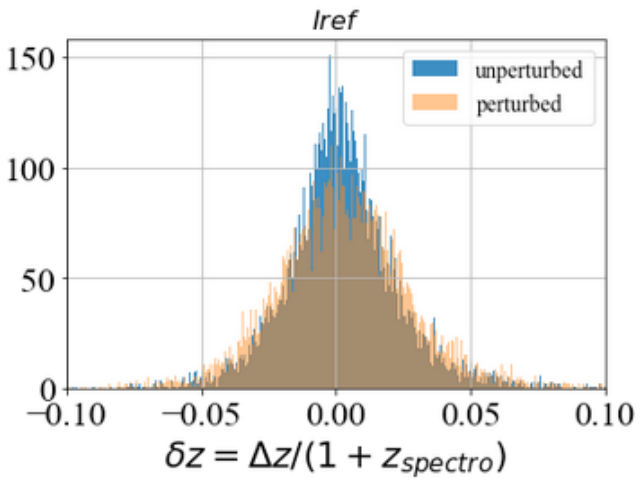
Weak models may fail to learn non-trivial classifiers. In the case of small capacity networks, attempting to train against a strong adversary (PGD) prevents the network from learning anything meaningful. The network converges to always predicting a fixed class, even though it could converge to an accurate classifier through standard training. The small capacity of the network forces the training procedure to sacrifice performance on natural examples in order to provide any kind of robustness against adversarial inputs.

The value of the saddle point problem decreases as we increase the capacity. Fixing an adversary model, and training against it, the value of (2.1) drops as capacity increases, indicating the the model can fit the adversarial examples increasingly well.

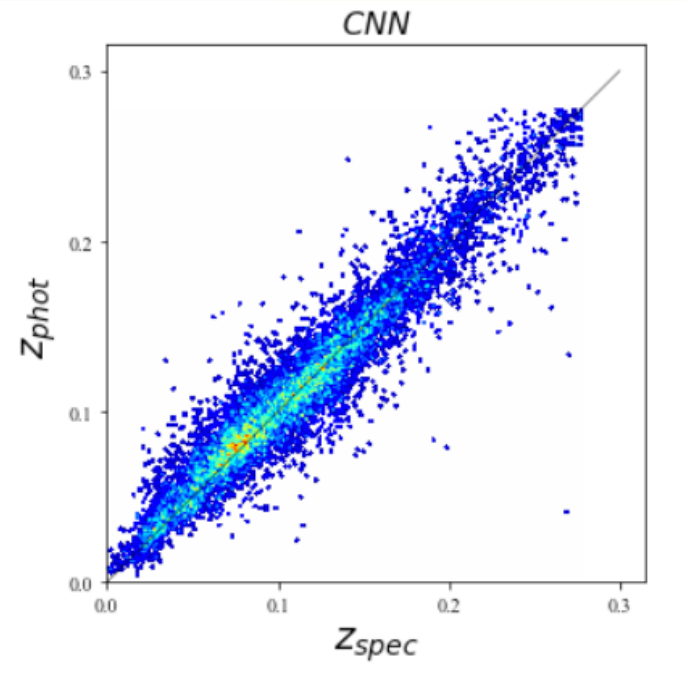
Madry et al

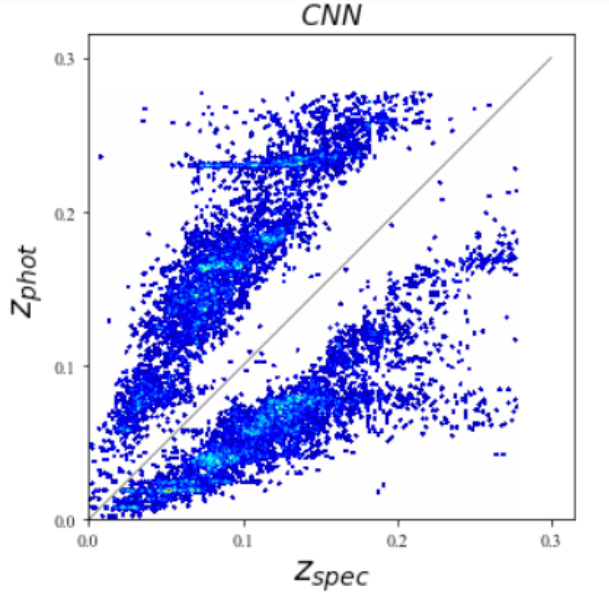
Inception trained with PGD L_∞ $\varepsilon=10^{-2}$ and attack similar

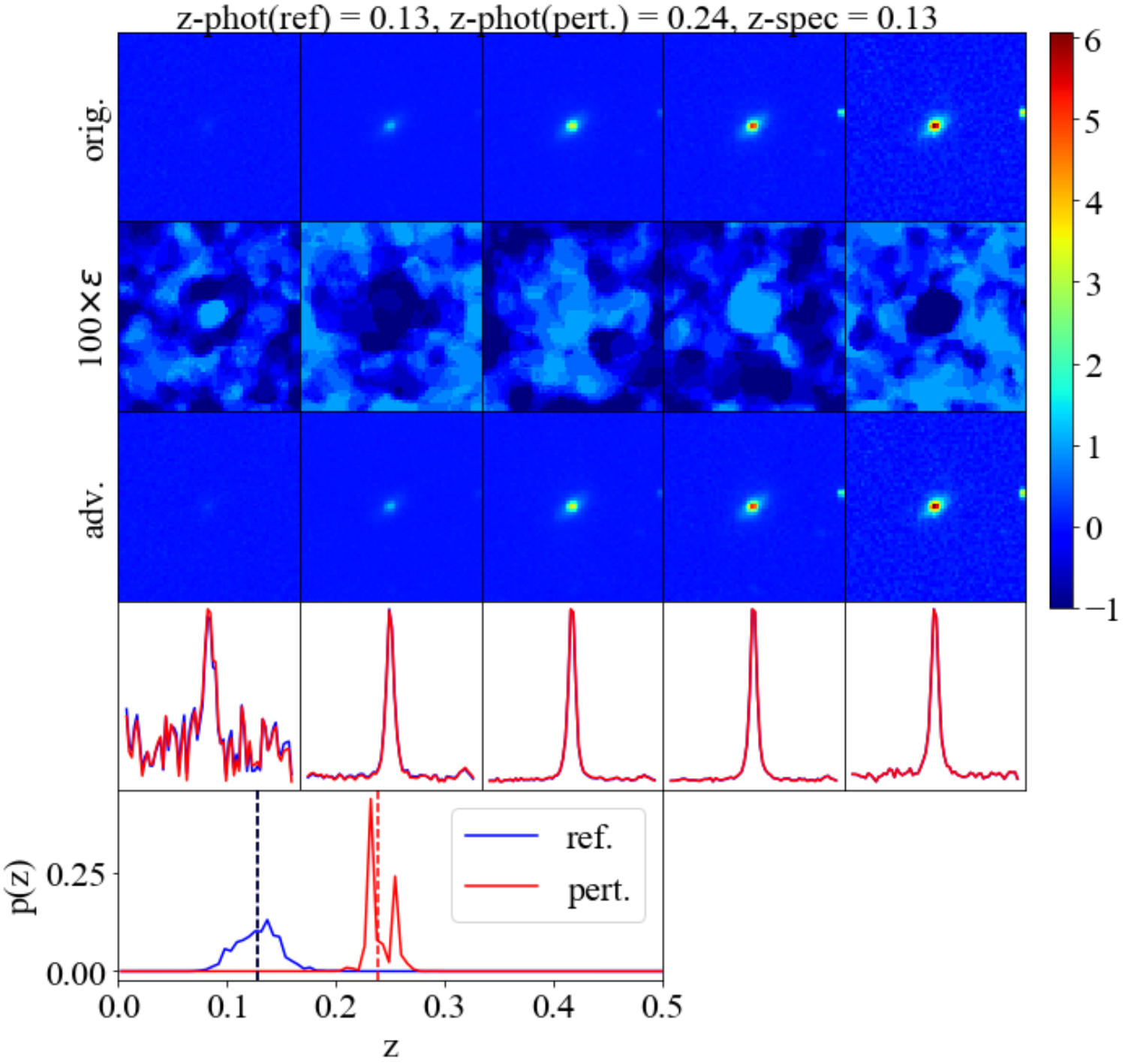




Simpler CNN


CNN
0.3
0.2
0.1
0.0
z_{phot}
0.0
0.1
0.2
0.3
z_{spec}


CNN
0.3
0.2
0.1
0.0
z_{phot}
0.0
0.1
0.2
0.3
z_{spec}


z-phot(ref) = 0.13, z-phot(pert.) = 0.24, z-spec = 0.13
orig.
100×ε
adv.
6
5
4
3
2
1
0
−1
p(z)
0.25
0.00
ref.
pert.
0.0
0.1
0.2
0.3
0.4
0.5
z