WGAN

Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN, arXiv prepring, 2017

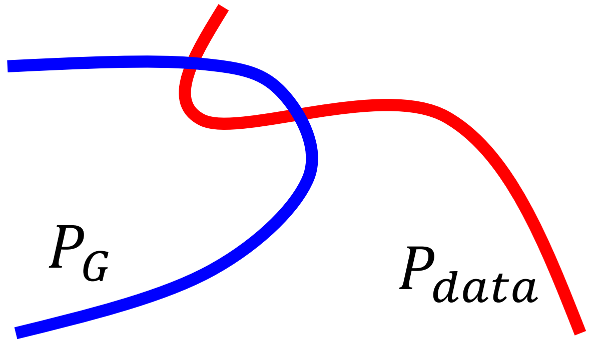
Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, “Improved Training of Wasserstein GANs”, arXiv prepring, 2017

JS divergence is not suitable

- In most cases, P_G and P_{data} are not overlapped.
- 1. The nature of data

Both P_{data} and P_G are low-dim manifold in high-dim space.

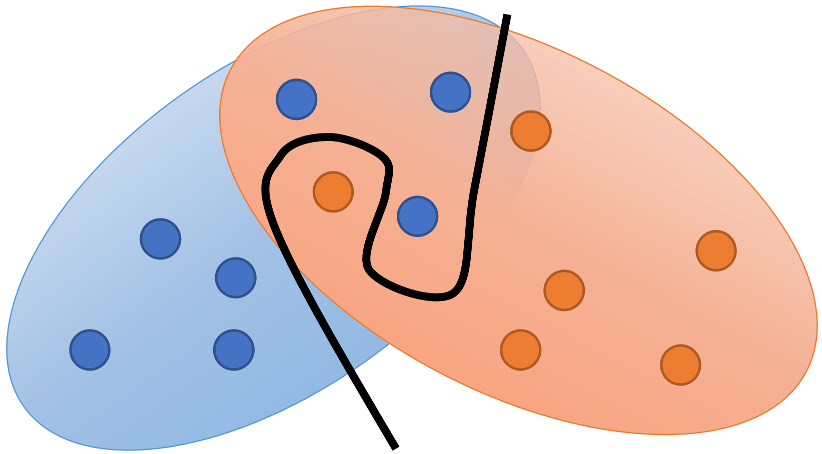
The overlap can be ignored.



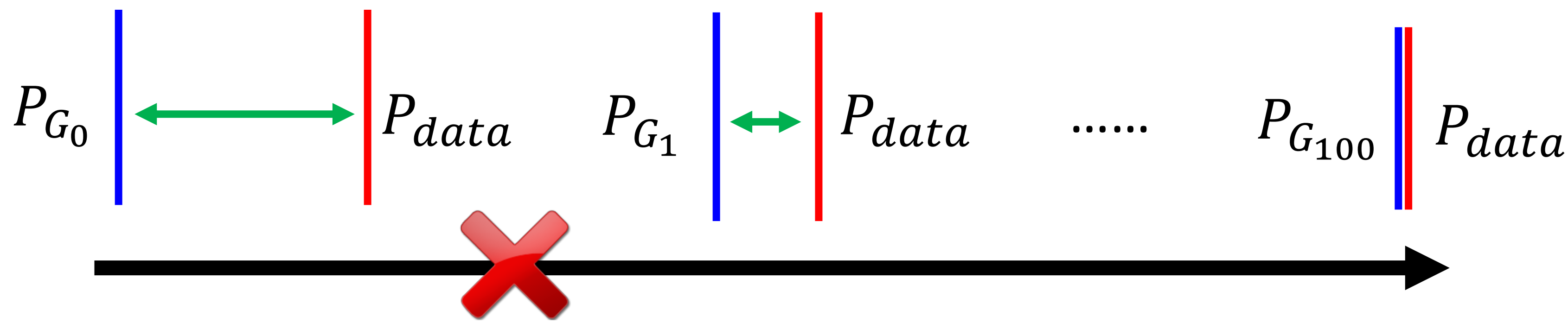
- 2. Sampling

Even though P_{data} and P_G have overlap.

If you do not have enough sampling ……



What is the problem of JS divergence?



Equally bad

$JS(P_{G_0}, P_{data}) = log2$ $\quad$ $JS(P_{G_1}, P_{data}) = log2$ $\quad$ ...... $\quad$ $JS(P_{G_{100}}, P_{data}) = 0$

在GAN中的D，如果是train binary classifier，則會發現當兩坨資料可以完全分開的時候他們的loss都是一樣的

JS divergence is log2 if two distributions do not overlap.

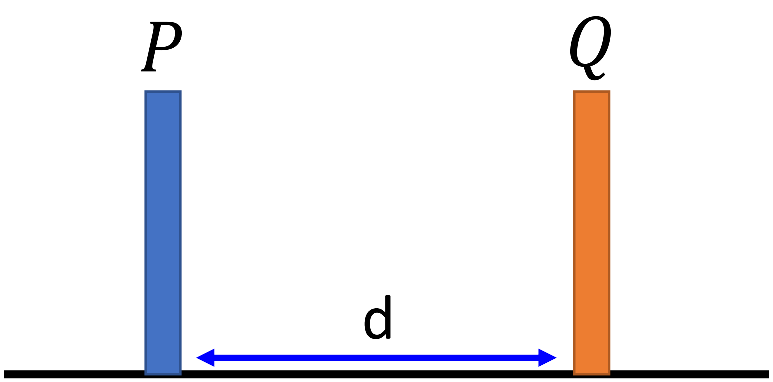
Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy

➡ Same objective value is obtained. ➡ Same divergence

可以改用LSGAN，將D的output的sigmoid拿掉，改為linear，這樣就變成regression problem

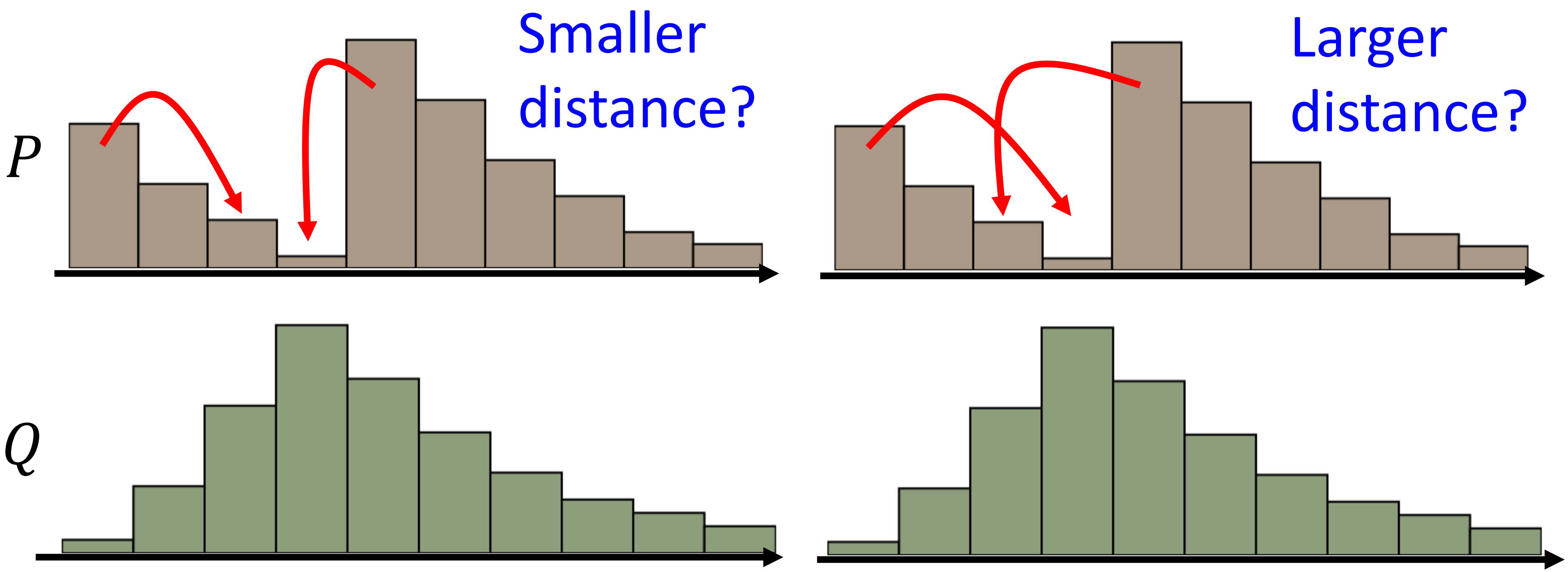
Earth Mover's Distance

- Considering one distribution P as a pile of earth, and another distribution Q as the target
- The average distance the earth mover has to move the earth.



$$W(P, Q) = d$$

Earth Mover’s Distance



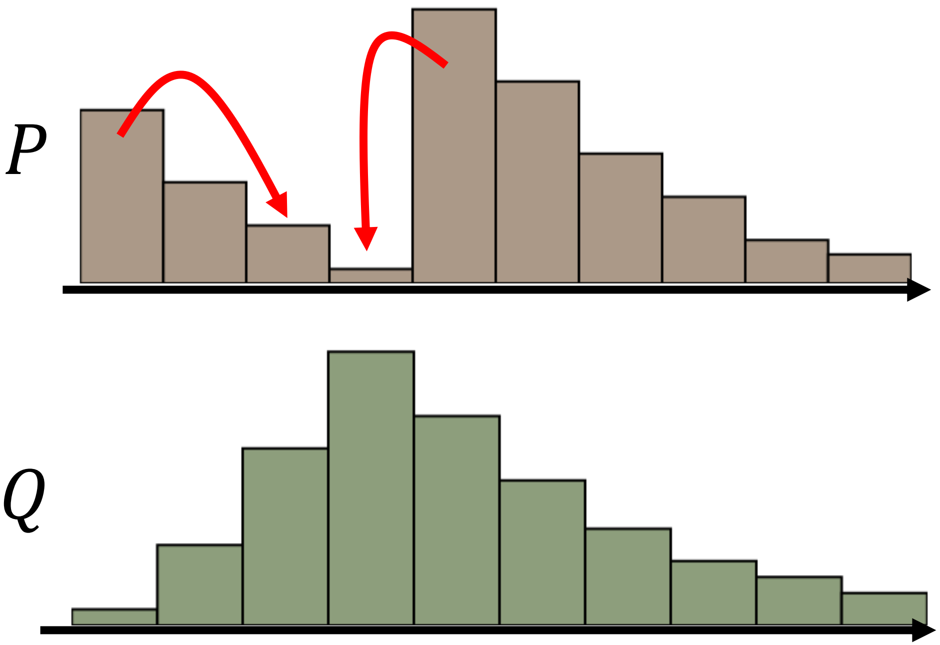
There many possible “moving plans”. 窮舉所有鏟土的方法找出最小distance

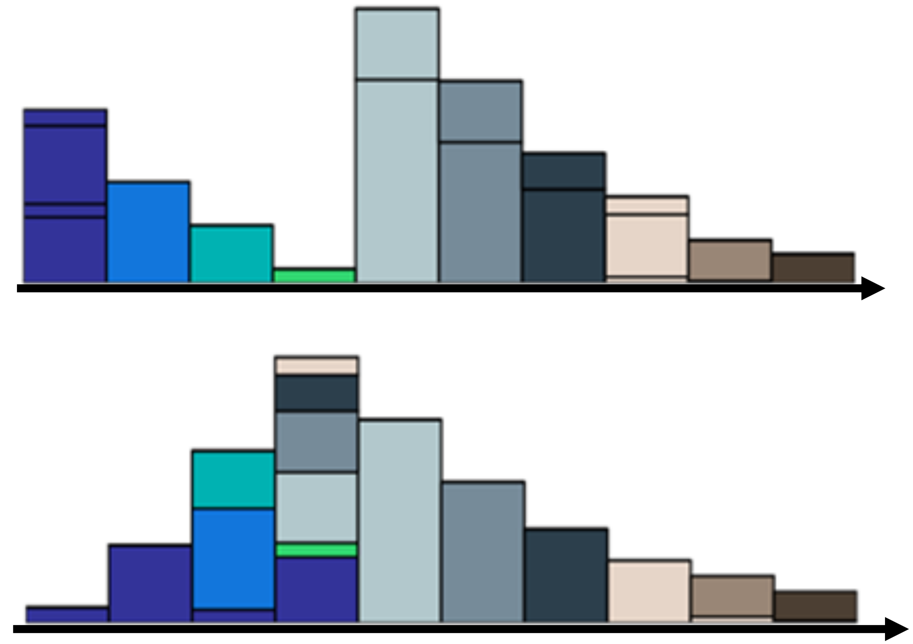
Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/

Earth Mover’s Distance

Best “moving plans” of this example



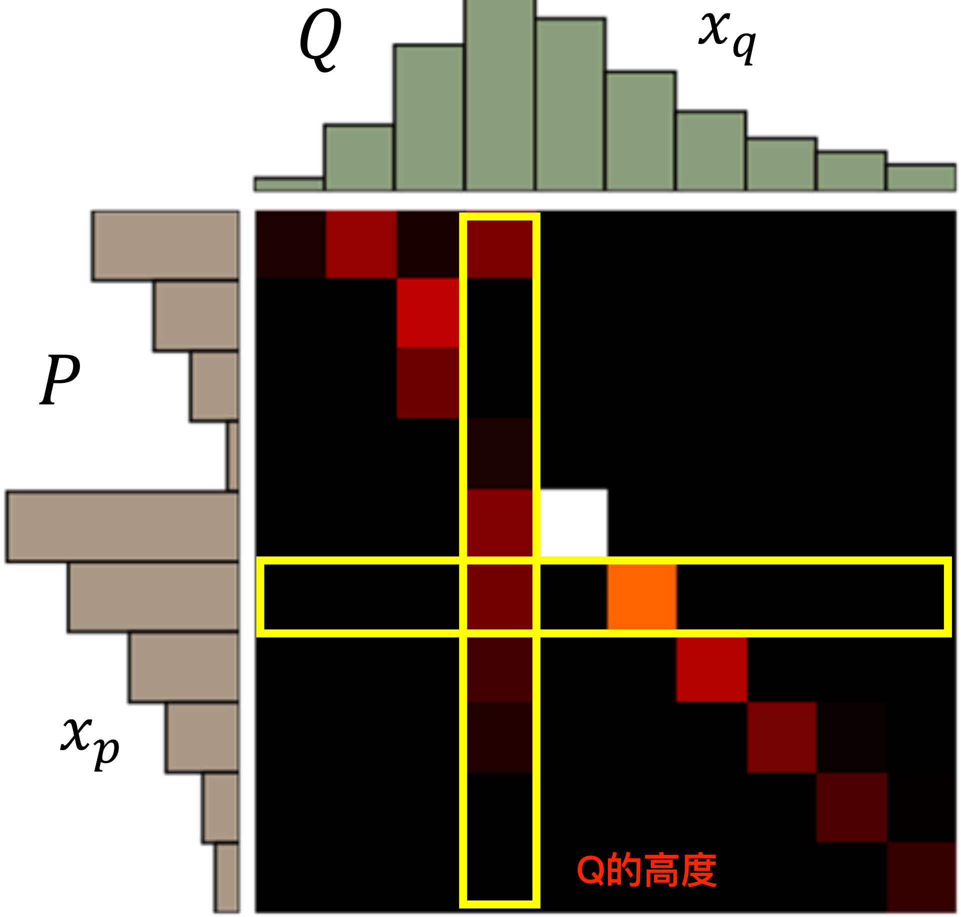


There many possible “moving plans”.

Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/

從P移動多少土到Q



moving plan γ

All possible plan Π

A “moving plan” is a matrix

The value of the element is the amount of earth from one position to another.

Average distance of a plan γ:

$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$

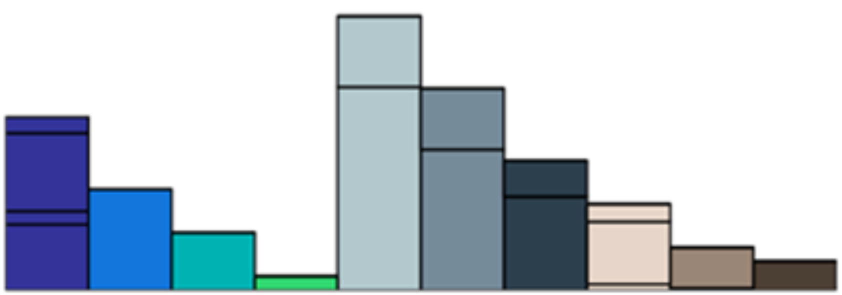
P的高度

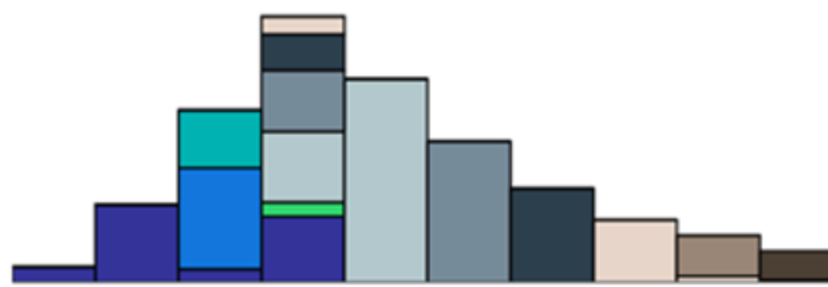
Earth Mover’s Distance:

解optimization problem

$$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$

The best plan

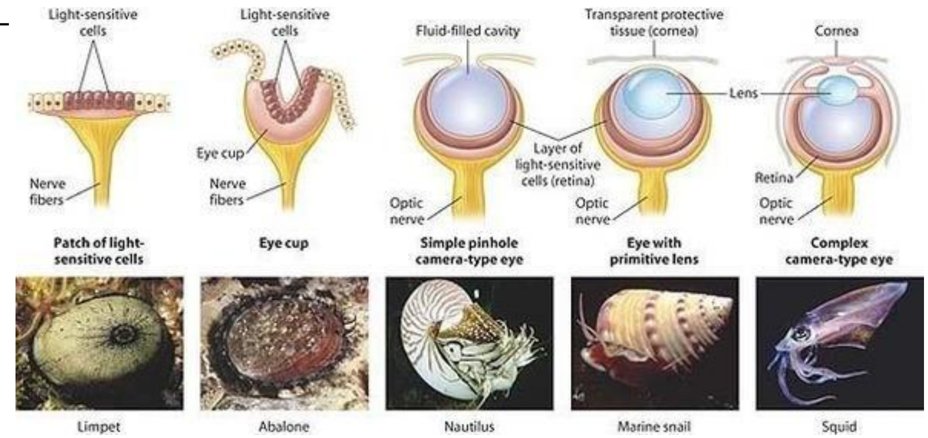




Why Earth Mover's Distance?

$$D_f(P_{data}||P_G)$$

$$W(P_{data}, P_G)$$



d_0 d_{50}

P_{G_0} $\longleftrightarrow$ P_{data} …… $P_{G_{50}}$ P_{data} …… $P_{G_{100}}$ P_{data}

每一次都要有一點點的進步才能有好的演化結果

$$JS(P_{G_0}, P_{data}) = log2 \qquad JS(P_{G_{50}}, P_{data}) = log2 \qquad JS(P_{G_{100}}, P_{data}) = 0$$

$$W(P_{G_0}, P_{data}) = d_0 \qquad W(P_{G_{50}}, P_{data}) = d_{50} \qquad W(P_{G_{100}}, P_{data}) = 0$$

Back to the GAN framework

原始的WGAN方法適用weight clipping

$$D_f(P_{data}||P_G) \Rightarrow W(P_{data},P_G)$$

$$= \max_{D} \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))]\}$$

$$W(P_{data},P_G)$$

$$= \max_{\boxed{D\in 1-Lipschitz}} \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)]\}$$

D has to be smooth enough

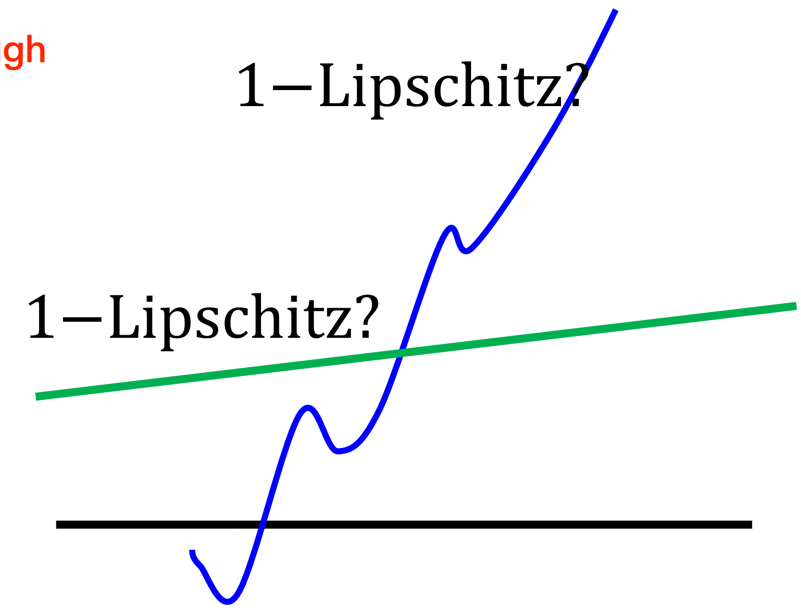
<u>***Lipschitz Function***</u>

$$\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$$

Output change　　　Input change

K=1 for "$1 - Lipschitz$"

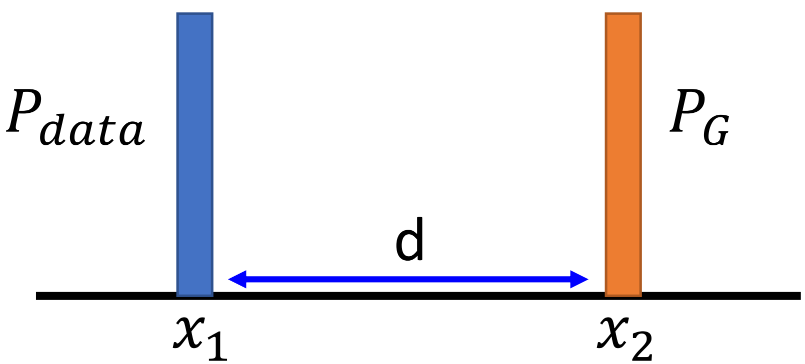
Do not change fast



Back to the GAN framework

$$W(P_{data}, P_G) = \max_{D \in 1-Lipschitz} \{ \overset{k+d}{E_{x \sim P_{data}}[D(x)]} - \overset{k}{E_{x \sim P_G}[D(x)]} \}$$

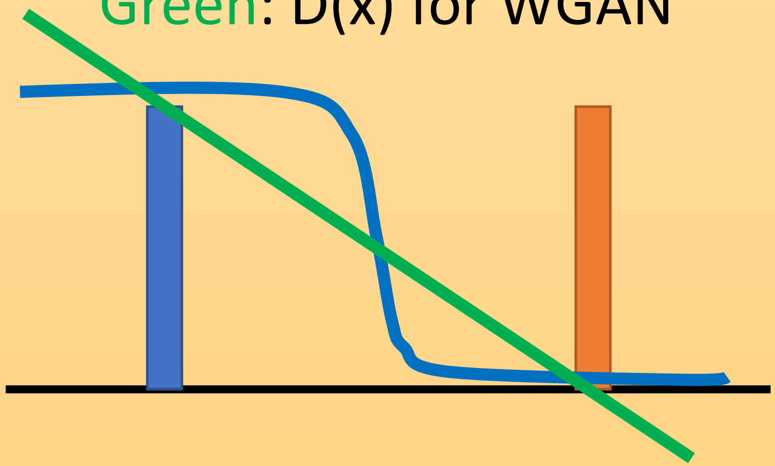
$D(x_1) = \cancel{+\infty}\ k + d$ $\qquad$ $D(x_2) = \cancel{-\infty}\ k$



$W(P_{data}, P_G) = d$

$\|D(x_1) - D(x_2)\| \leq \|x_1 - x_2\|$

Blue: D(x) for original GAN
Green: D(x) for WGAN



WGAN will provide gradient to push P_G towards P_{data}

Back to the GAN framework

K $W(P_{data}, P_G)$

$$= \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

K

How to use gradient descent to optimize?

Weight clipping:

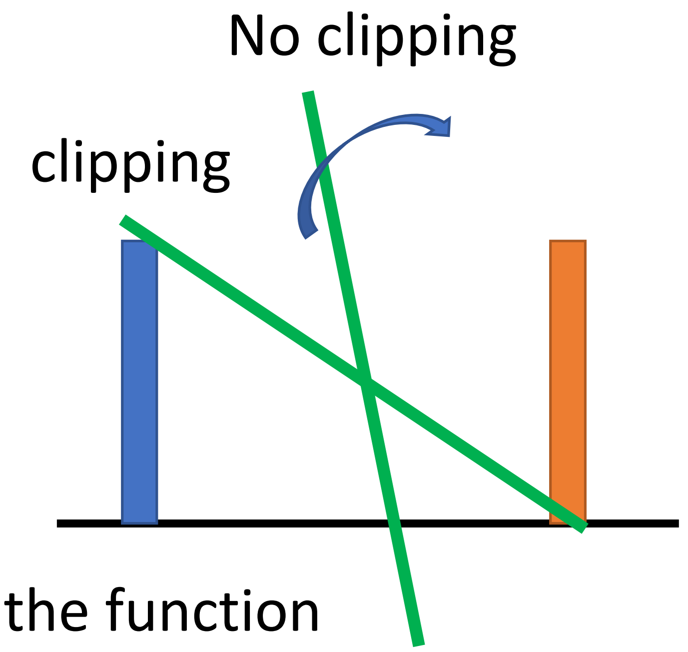
Force the weights w between c and -c

After parameter update,
if w > c, then w=c; if w<-**c**, then w=-c

We only ensure that

$$\|D(x_1) - D(x_2)\| \le K\|x_1 - x_2\|$$

For some K



Do not truly find function D maximizing the function

Algorithm of WGAN

- In each training iteration:

No sigmoid for the output of D

Learning D

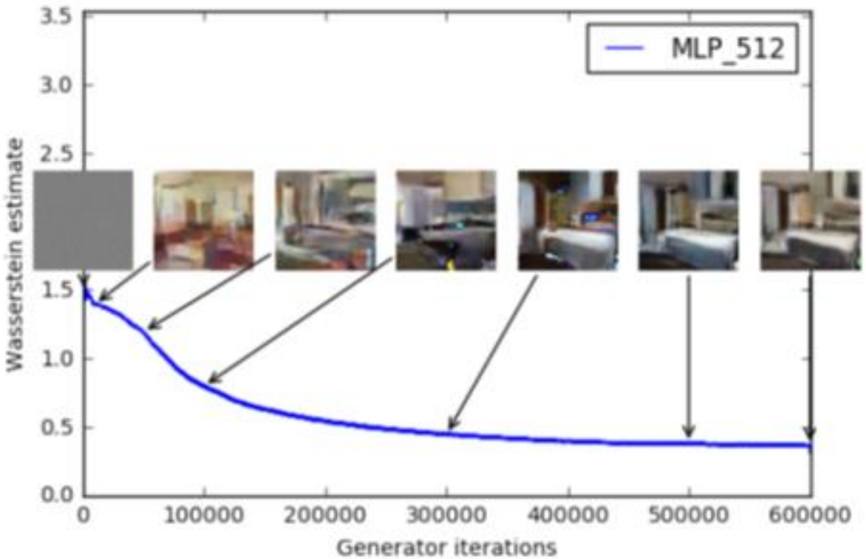
Repeat k times

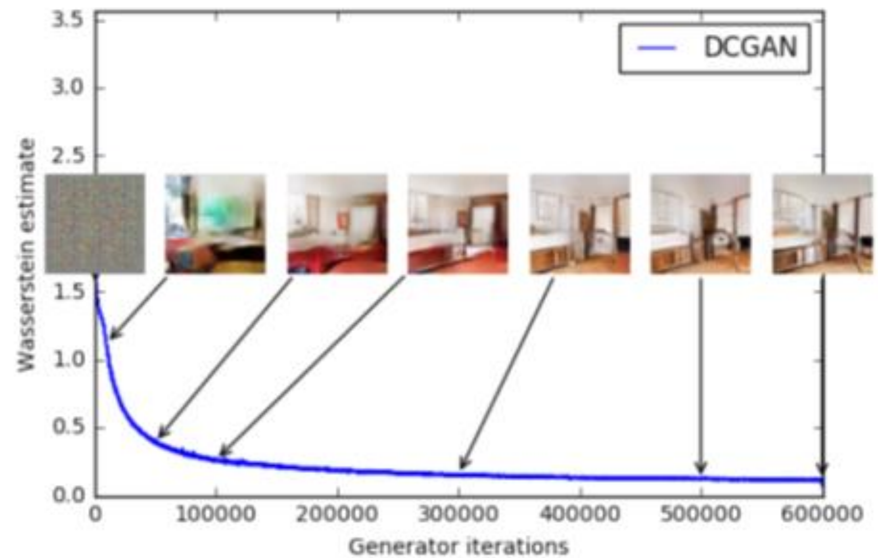
  - Sample m examples $\{x^1, x^2, \ldots, x^m\}$ from data distribution $P_{data}(x)$
  - Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from the prior $P_{prior}(z)$
  - Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
  - Update discriminator parameters θ_d to maximize
    - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} D(x^i) - \frac{1}{m}\sum_{i=1}^{m} D(\tilde{x}^i)$
    - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$ Weight clipping

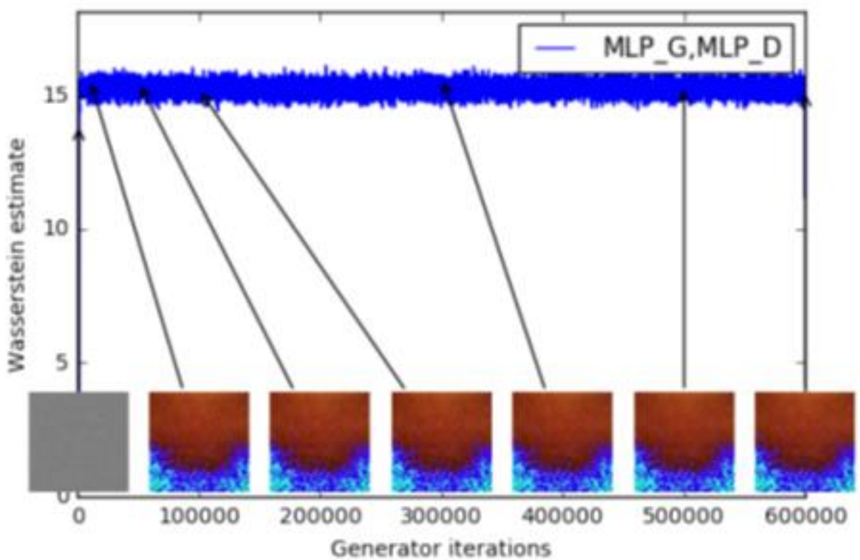
Learning G

Only Once

  - Sample another m noise samples $\{z^1, z^2, \ldots, z^m\}$ from the prior $P_{prior}(z)$
  - Update generator parameters θ_g to minimize
    - $\tilde{V} = \cancel{\frac{1}{m}\sum_{i=1}^{m} logD(x^i)} - \frac{1}{m}\sum_{i=1}^{m} D\left(G(z^i)\right)$
    - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$







Vertical

$$
\begin{aligned}
&W(P_{data}, P_G) \\
&= \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] \\
&\quad - E_{x \sim P_G}[D(x)]\}
\end{aligned}
$$

https://arxiv.org/abs/1701.07875

Improved WGAN

$$W(P_{data}, P_G) = \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

A differentiable function is 1-Lipschitz if and only if it has gradients with norm less than or equal to 1 everywhere.

$D \in 1 - Lipschitz$ $\Leftrightarrow$ $\|\nabla_x D(x)\| \leq 1$ for all x

等價的

$$W(P_{data}, P_G) \approx \max_{D} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]$$

penalty，類似regularization

$$\cancel{-\lambda \int_x max(0, \|\nabla_x D(x)\| - 1) dx\}}$$

Prefer $\|\nabla_x D(x)\| \leq 1$ for all x

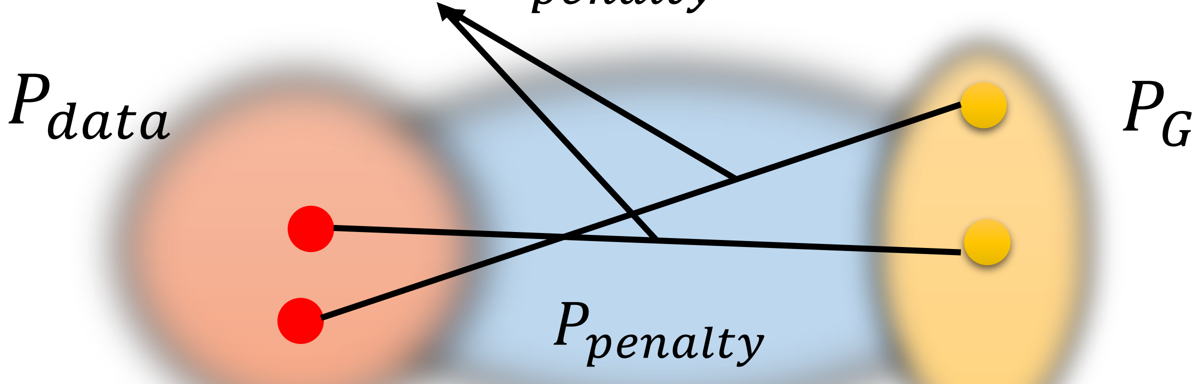
$$-\lambda E_{x \sim P_{penalty}}[max(0, \|\nabla_x D(x)\| - 1)]\}$$

但是無法真的對所有x(image)算gradient

Prefer $\|\nabla_x D(x)\| \leq 1$ for x sampling from $x \sim P_{penalty}$

Improved WGAN

$$W(P_{data}, P_G) \approx \max_D \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]$$
$$-\lambda E_{x \sim P_{penalty}}[max(0, \|\nabla_x D(x)\| - 1)]\}$$



從兩個distribution中sample各兩個點連起來做interpolation，中間的值就是penalty

"Given that enforcing the Lipschitz constraint everywhere is intractable, enforcing it ***only along these straight lines*** seems sufficient and experimentally results in good performance."

Only give gradient constraint to the region between P_{data} and P_G because they influence how P_G moves to P_{data}

其實是滿合理的，因為要移動PG就是要參考中間這些區域做gradient descend

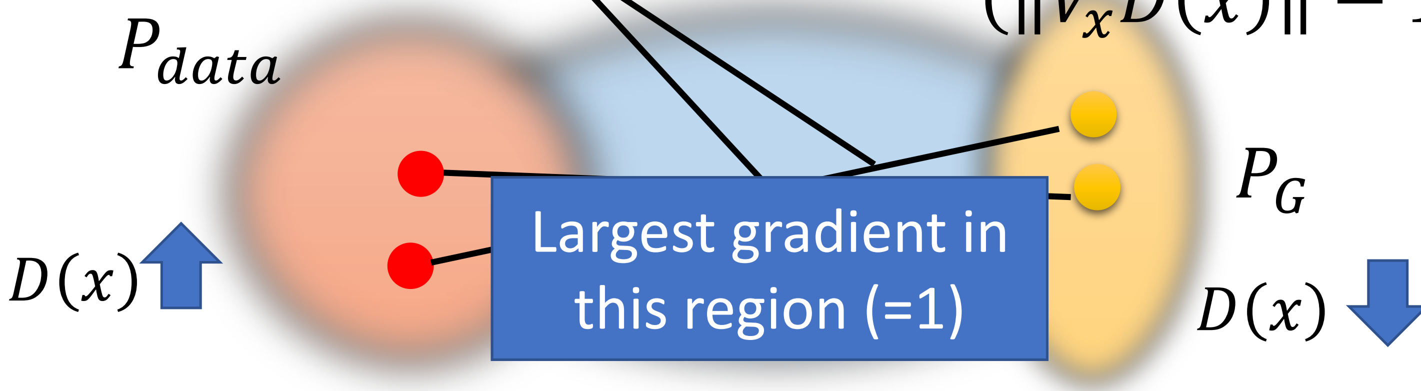
Improved WGAN

實際在paper實作的時候說，根據實驗結果發現所有的gradient都越接近1越好，因此直接對gradient=1做regularization

$$W(P_{data}, P_G) \approx \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)]$$

$$-\lambda E_{x\sim P_{penalty}}[\cancel{max(0, \|\nabla_x D(x)\| - 1)}]\}$$

$$(\|\nabla_x D(x)\| - 1)^2$$



"One may wonder why we penalize the norm of the gradient for differing from 1, instead of just penalizing large gradients. The reason is that the optimal critic ... actually has gradients with norm 1 almost everywhere under Pr and Pg"

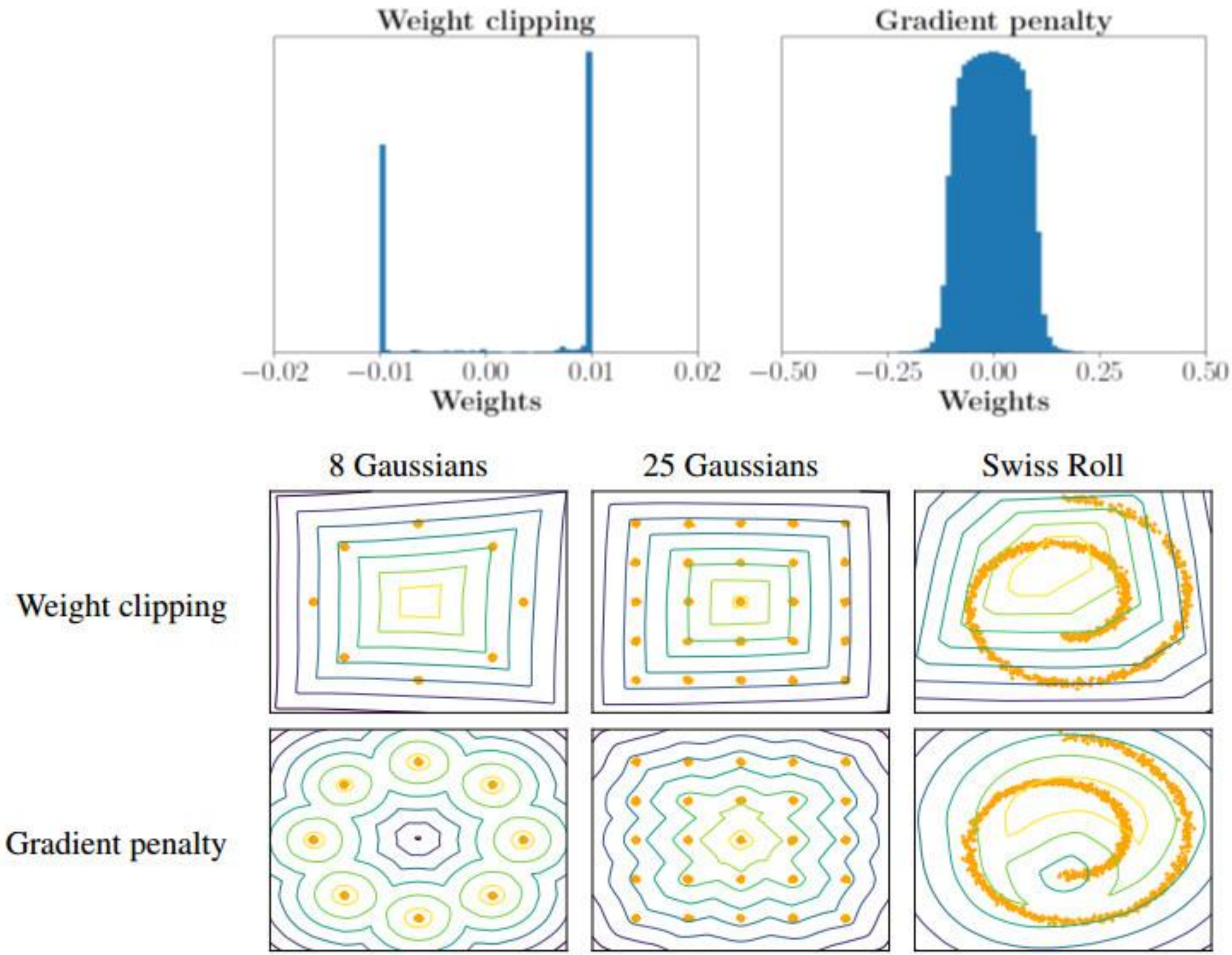
(check the proof in the appendix)

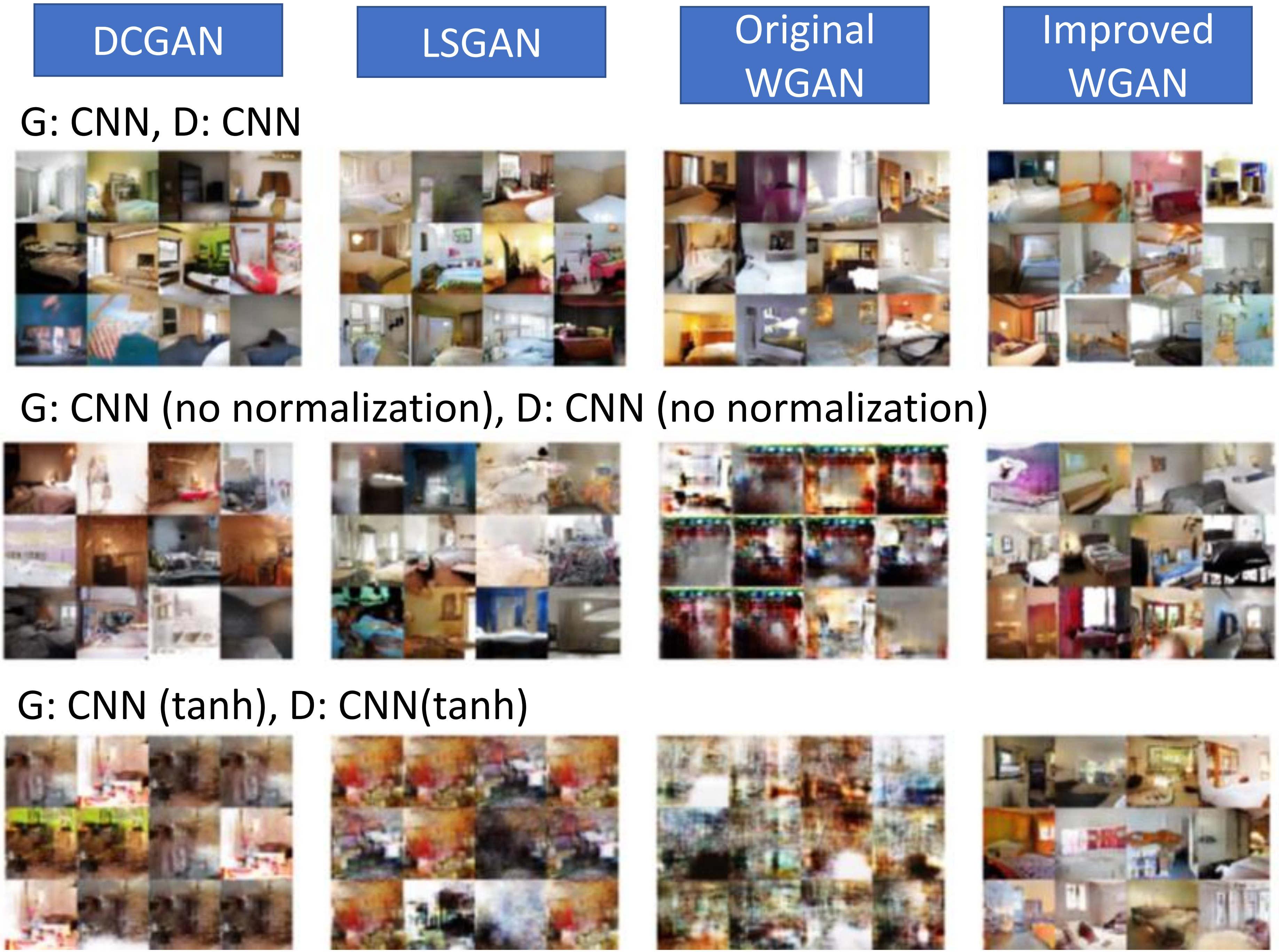
"Simply penalizing overly large gradients also works in theory, but experimentally we found that this approach converged faster and to better optima."

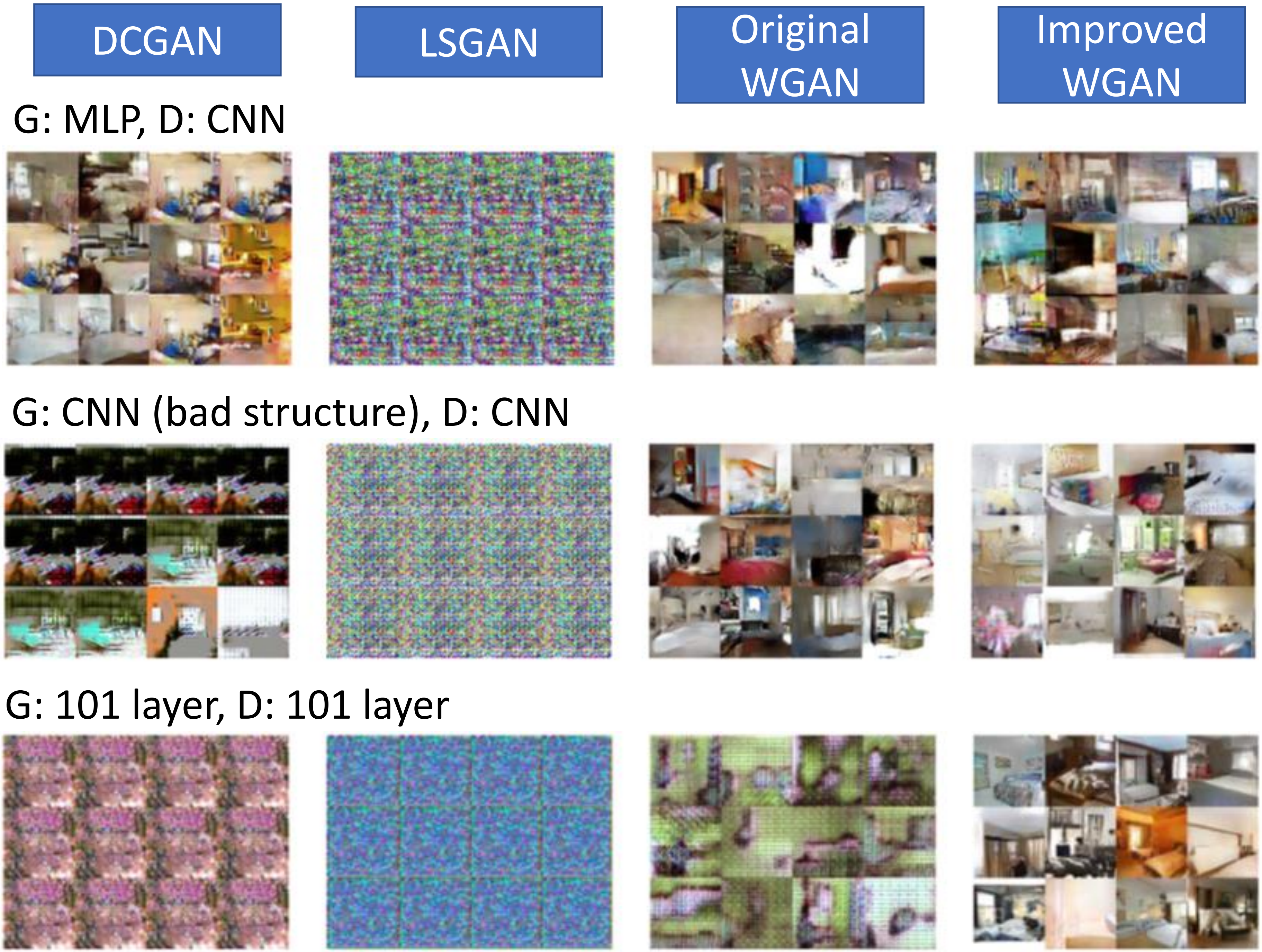
後續還有其他work
how to train your dragon
improved improved WGAN

https://arxiv.org/abs/1704.00028

Improved WGAN




DCGAN
LSGAN
Original WGAN
Improved WGAN
G: CNN, D: CNN
G: CNN (no normalization), D: CNN (no normalization)
G: CNN (tanh), D: CNN(tanh)


DCGAN
LSGAN
Original WGAN
Improved WGAN
G: MLP, D: CNN
G: CNN (bad structure), D: CNN
G: 101 layer, D: 101 layer

ICLR 2018

Spectrum Norm

很強！！有空看一下

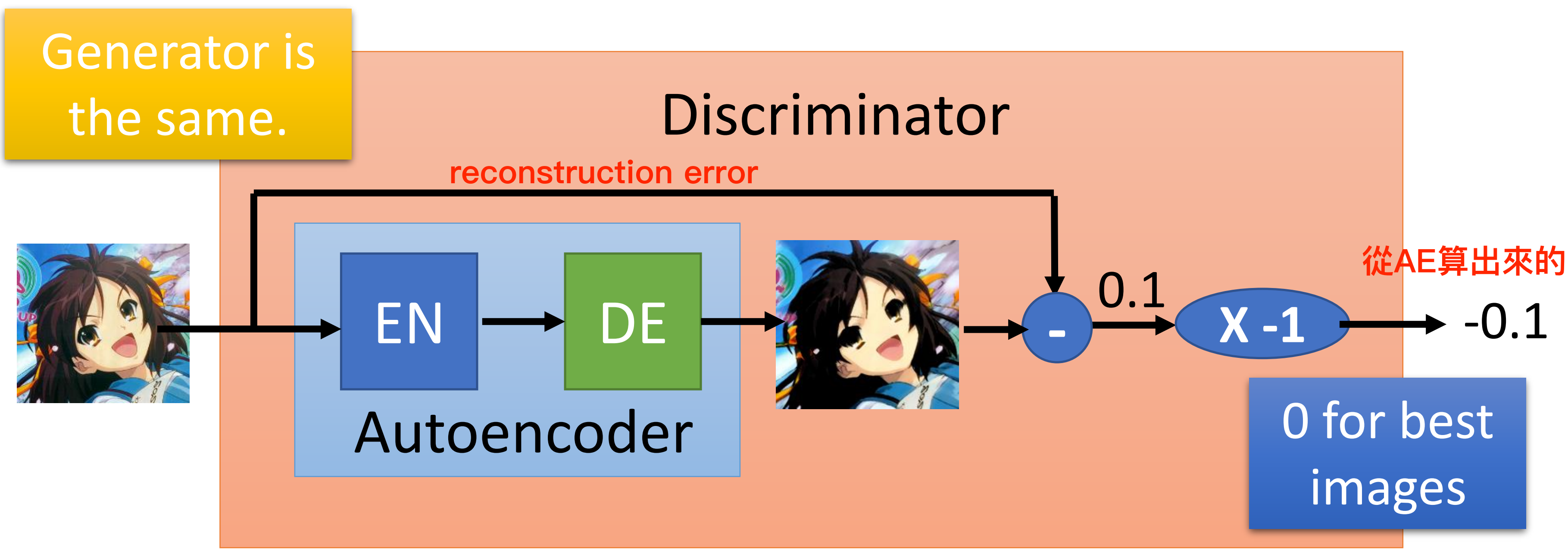
Energy-based GAN

Ref: Junbo Zhao, Michael Mathieu, Yann LeCun, Energy-based Generative Adversarial Network, ICRL 2017

Energy-based GAN (EBGAN)

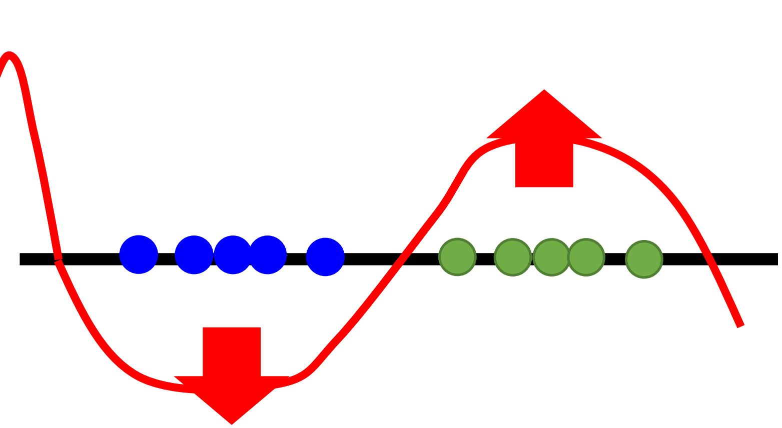
- Using an autoencoder as discriminator D

An image is good. = It can be reconstructed by autoencoder.



優點是Discriminator可以pre-trained，只要讓他看許多positive example就可以根據reconstruct error pretrain

EBGAN

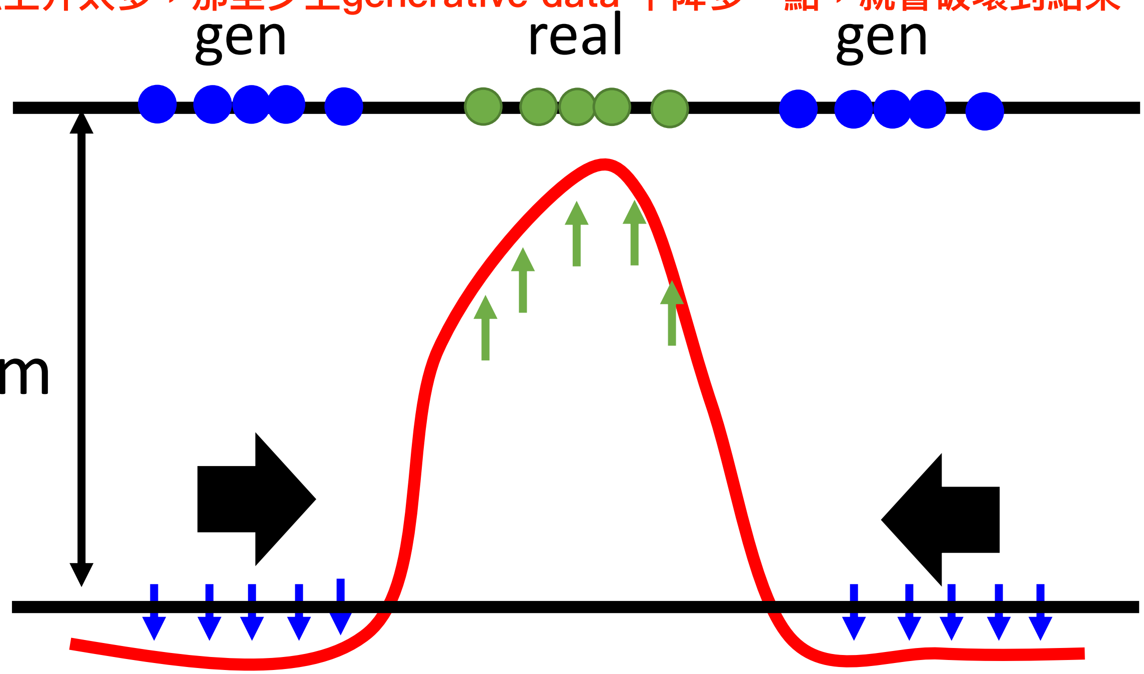


Auto-encoder based discriminator only give limited region large value.

定義一個threshold，使得Generator產生的reconstruction error不會train壞掉

因為擔心G發現real data部分無法上升太多，那至少上generative data 下降多一點，就會破壞到結果

0 is for the best.



Do not have to be very negative

Hard to reconstruct, easy to destroy