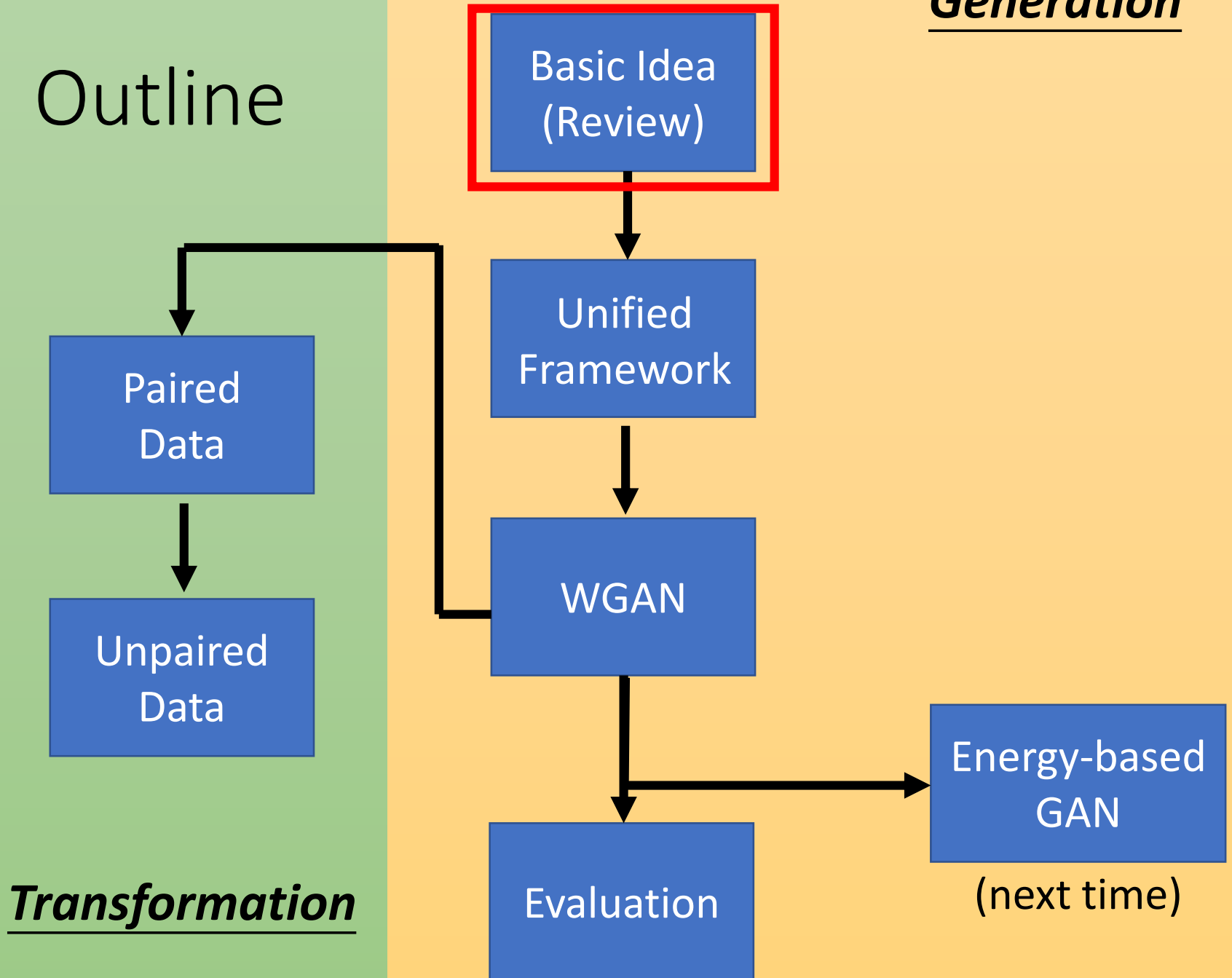


Improving Generative Adversarial Network (GAN)

Hung-yi Lee

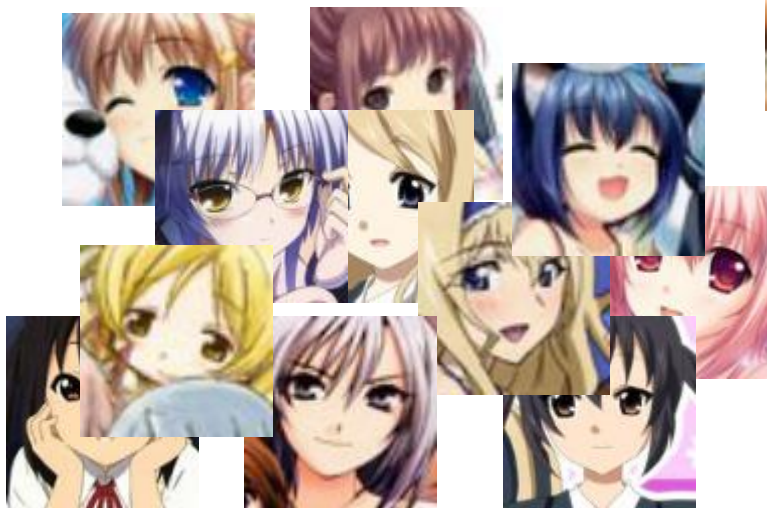
Generation

Outline



Generation

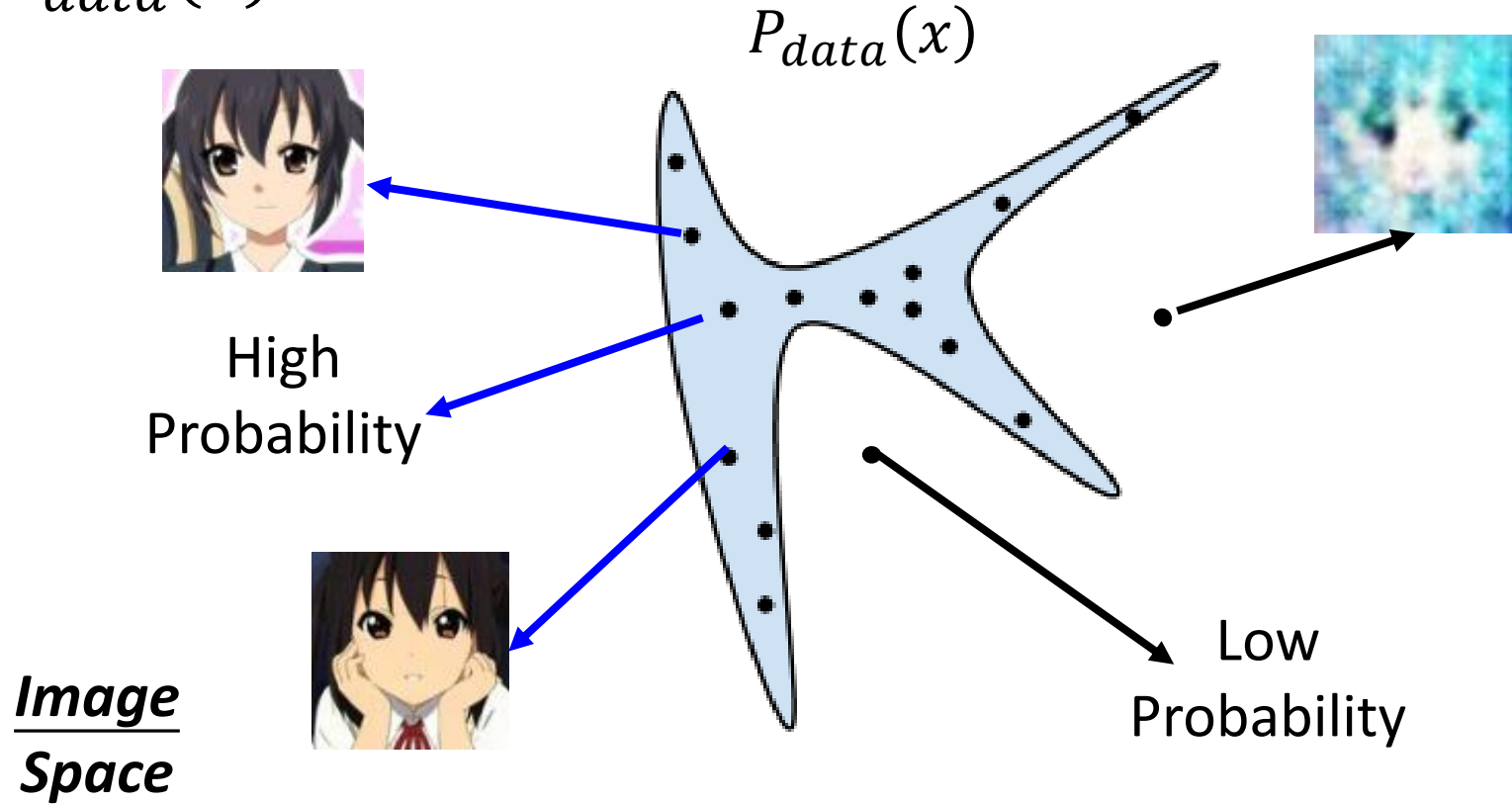
Using Generative
Adversarial
Network (GAN)



Drawing?

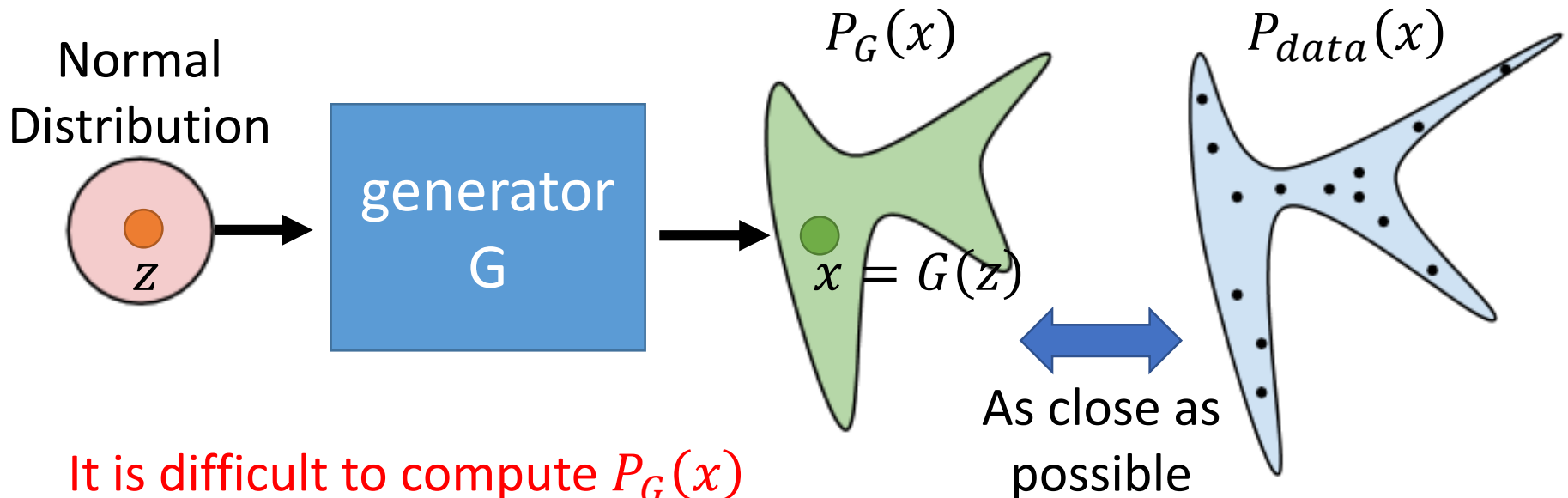
Basic Idea of GAN

- The data we want to generate has a distribution $P_{data}(x)$



Basic Idea of GAN

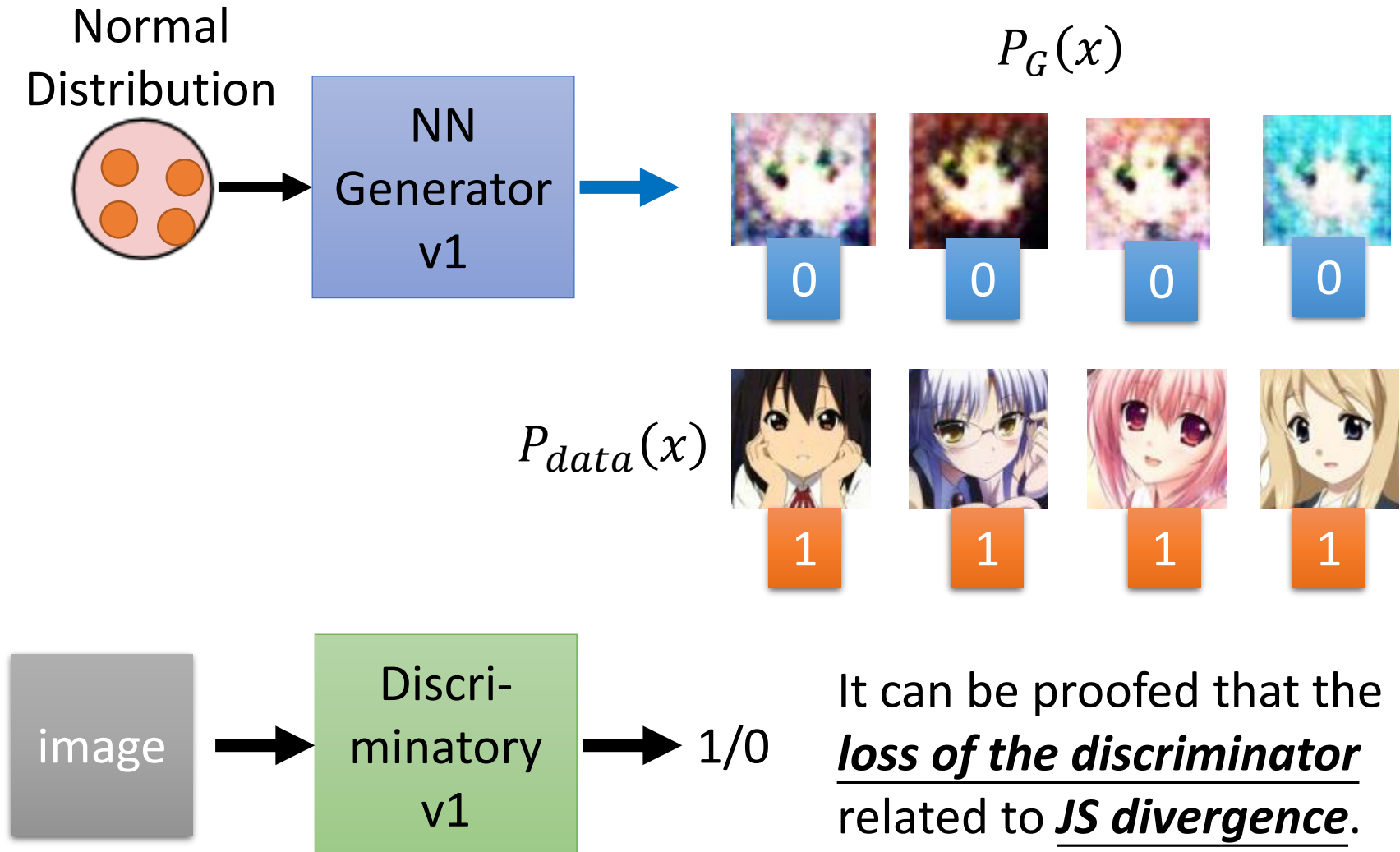
- A generator G is a network. The network defines a probability distribution.



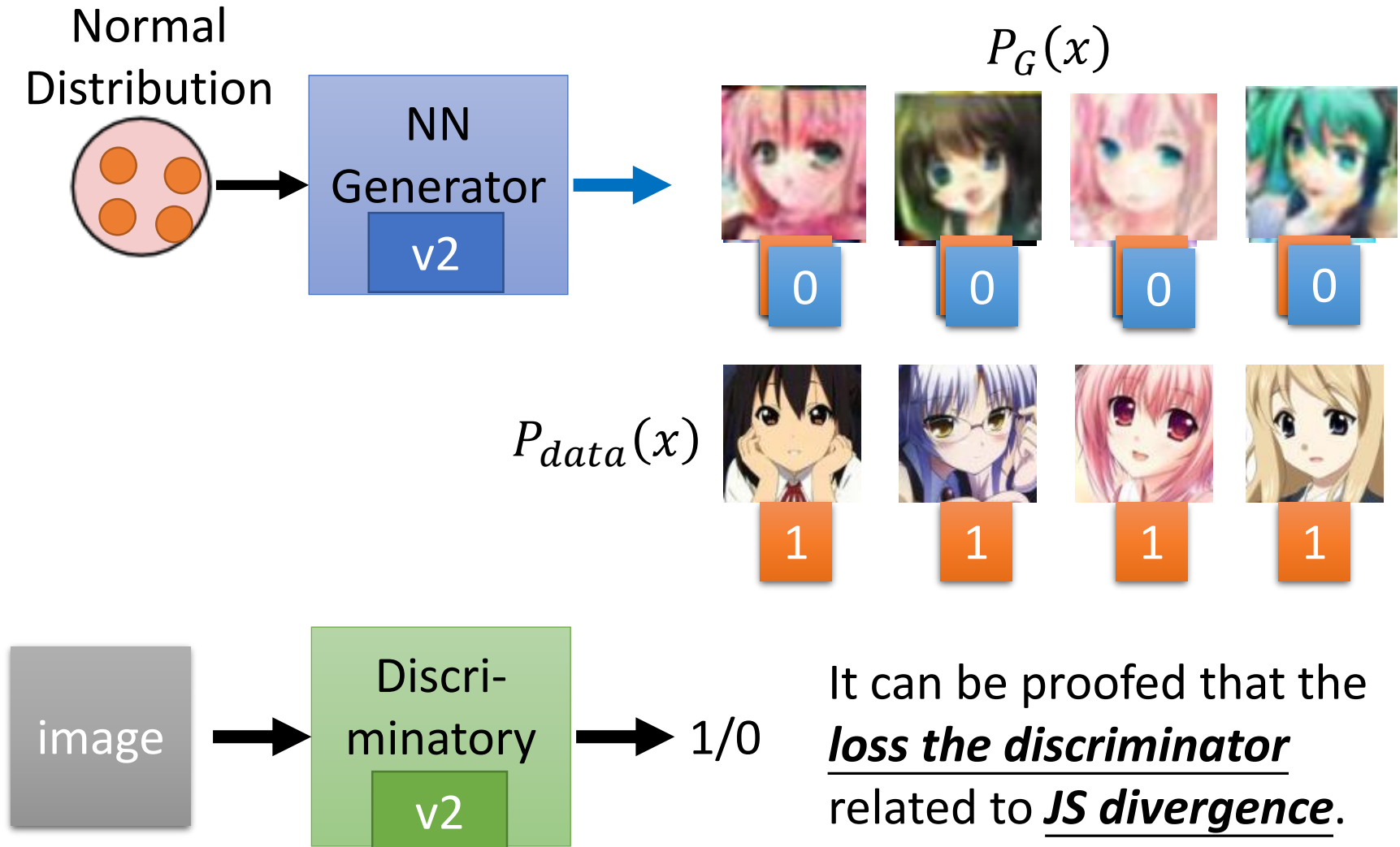
It is difficult to compute $P_G(x)$

We can only sample from the distribution.

Basic Idea of GAN



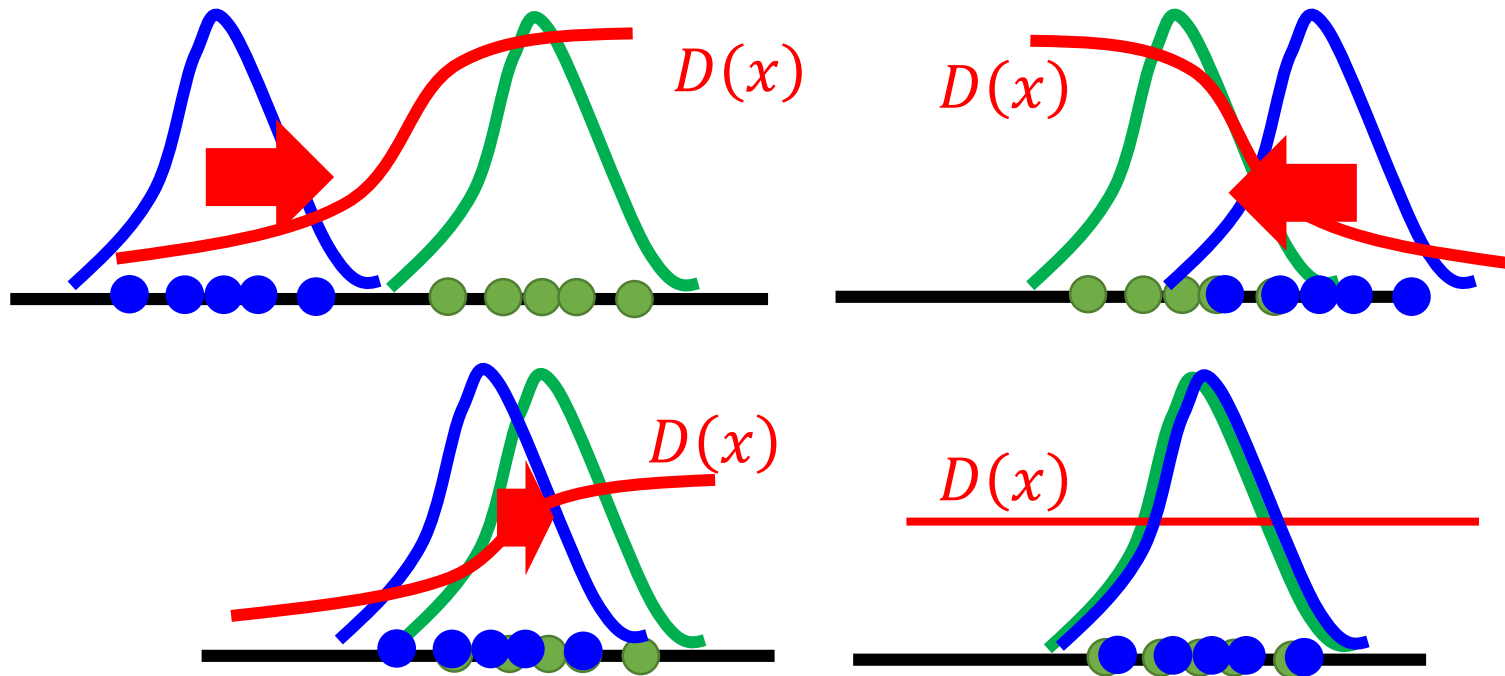
Basic Idea of GAN



Intuition

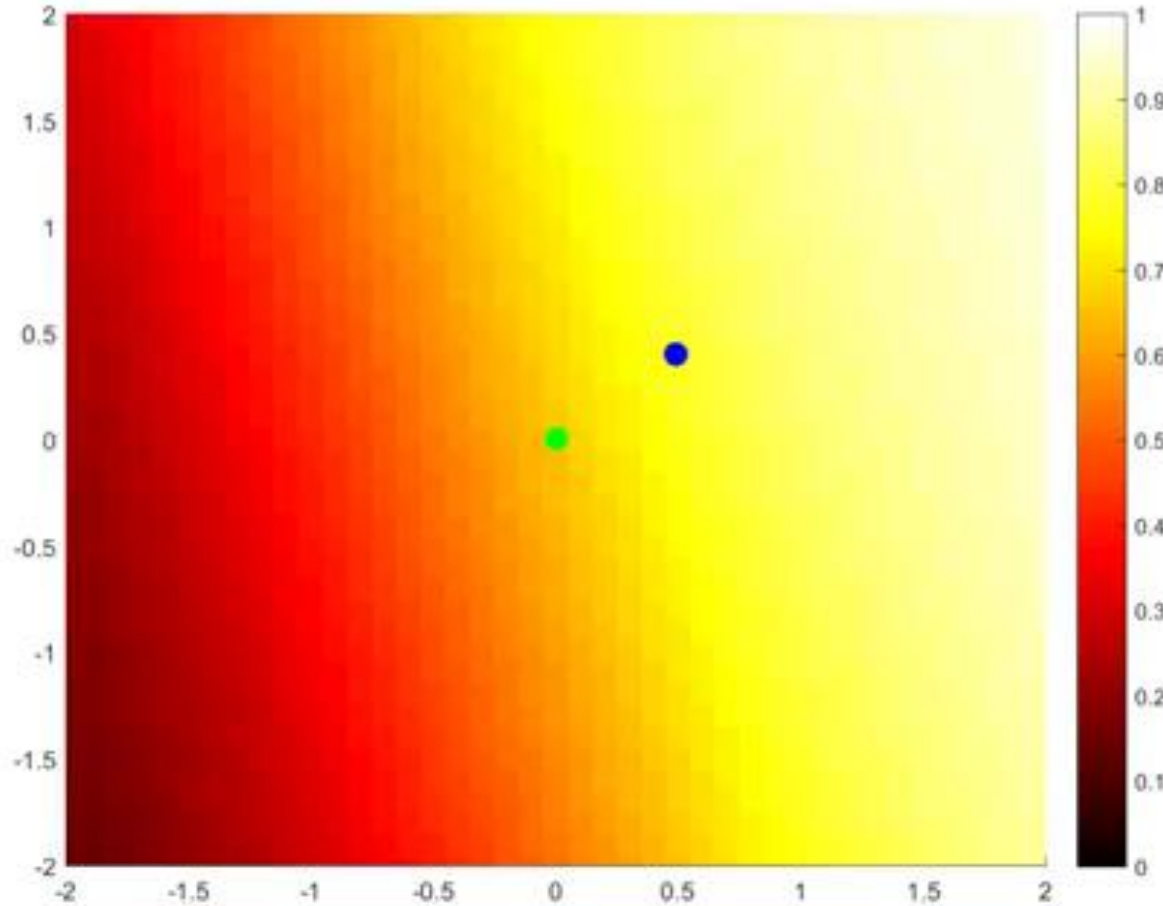
- Discriminator
- Data (target) distribution
- Generated distribution

- Discriminator leads the generator



Original GAN

The discriminator is flat in the end.



Source: <https://www.youtube.com/watch?v=ebMei6bYeWw> (credit: Benjamin Striner)

Algorithm

Initialize θ_d for D and θ_g for G

- In each training iteration:

Learning
D

Repeat
k times

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

Learning
G

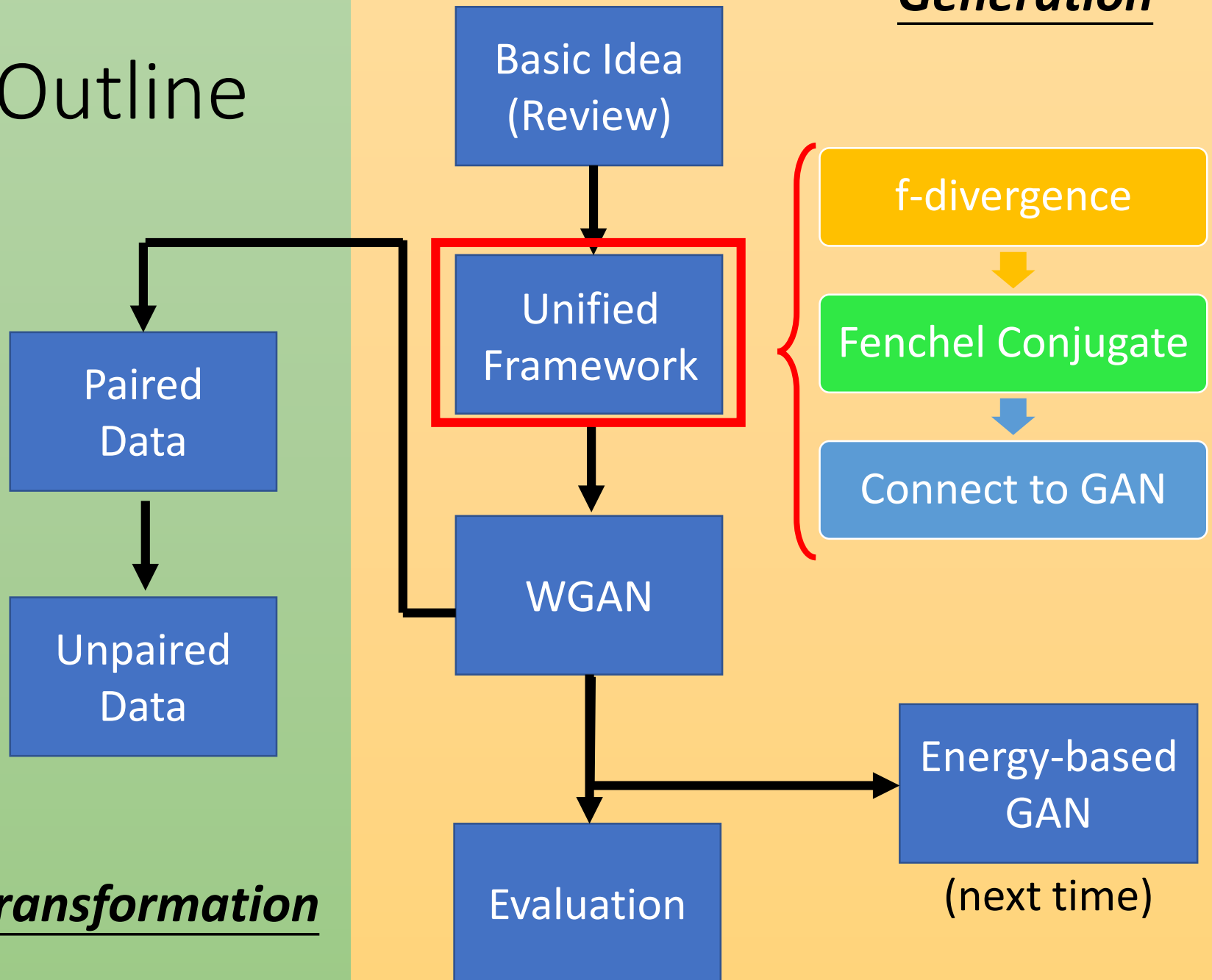
Only
Once

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

Outline

Generation

Transformation



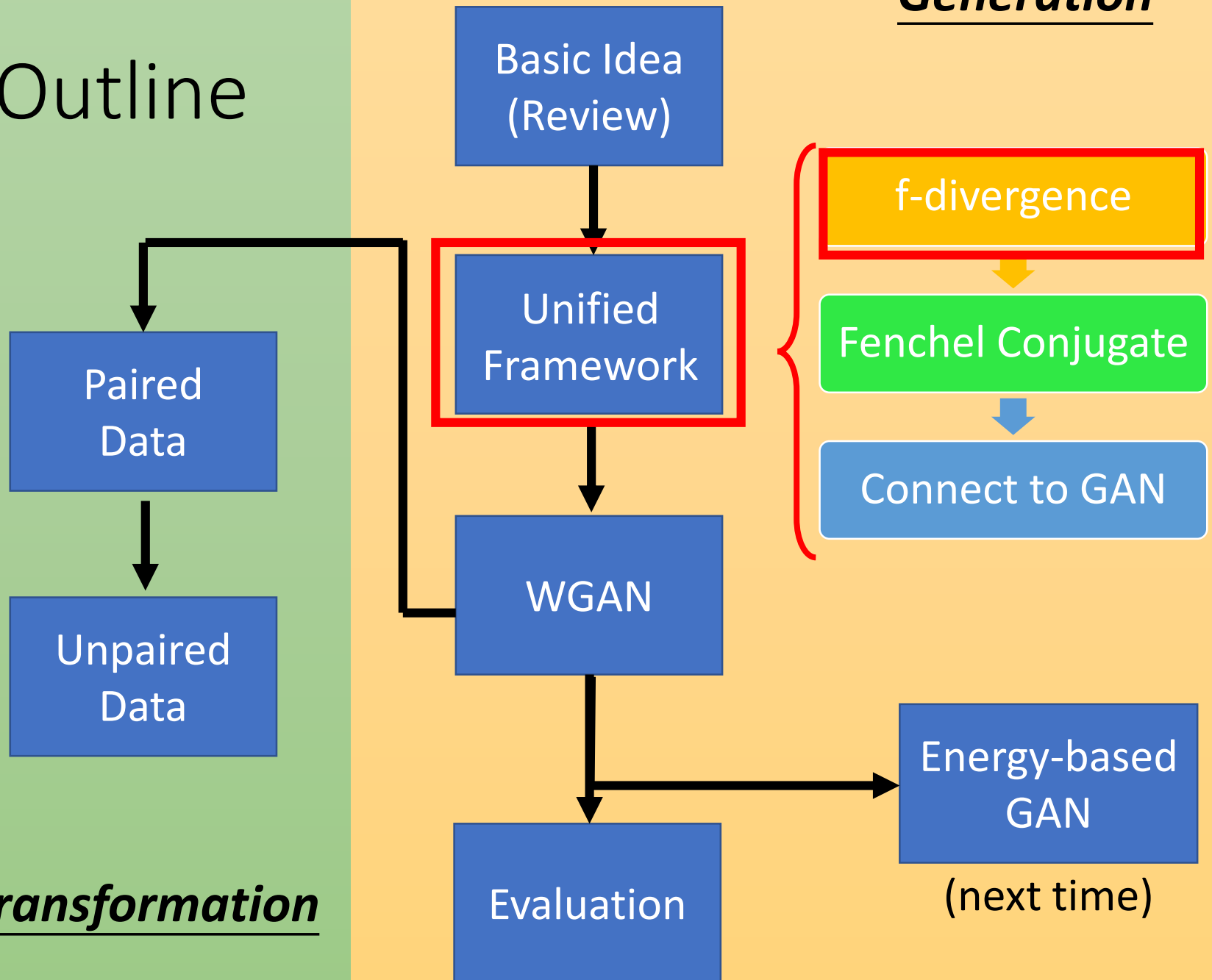
Reference

- Sebastian Nowozin, Botond Cseke, Ryota Tomioka, "**f-GAN**: Training Generative Neural Samplers using Variational Divergence Minimization", NIPS, 2016
- One sentence: you can use any f-divergence

Outline

Generation

Transformation



f-divergence

P and Q are two distributions. $p(x)$ and $q(x)$ are the probability of sampling x .

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx \quad \begin{array}{l} f \text{ is convex} \\ f(1) = 0 \end{array}$$

$D_f(P||Q)$ evaluates the difference of P and Q

If $p(x) = q(x)$ for all x

smallest

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx = 0$$

$\frac{p(x)}{q(x)} = 1$
 $= 0$

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

Because f
is convex

$$\geq f\left(\int_x \cancel{q(x)} \frac{p(x)}{\cancel{q(x)}} dx\right)$$
$$= f(1) = 0$$

If P and Q are the same distributions,
 $D_f(P||Q)$ has the smallest value, which is 0

***f*-divergence**

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

f is convex

$$f(1) = 0$$

$$f(x) = x \log x$$

KL

$$D_f(P||Q) = \int_x q(x) \frac{p(x)}{q(x)} \log \left(\frac{p(x)}{q(x)} \right) dx = \int_x p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

$$f(x) = -\log x$$

Reverse KL

$$D_f(P||Q) = \int_x q(x) \left(-\log \left(\frac{p(x)}{q(x)} \right) \right) dx = \int_x q(x) \log \left(\frac{q(x)}{p(x)} \right) dx$$

$$f(x) = (x - 1)^2$$

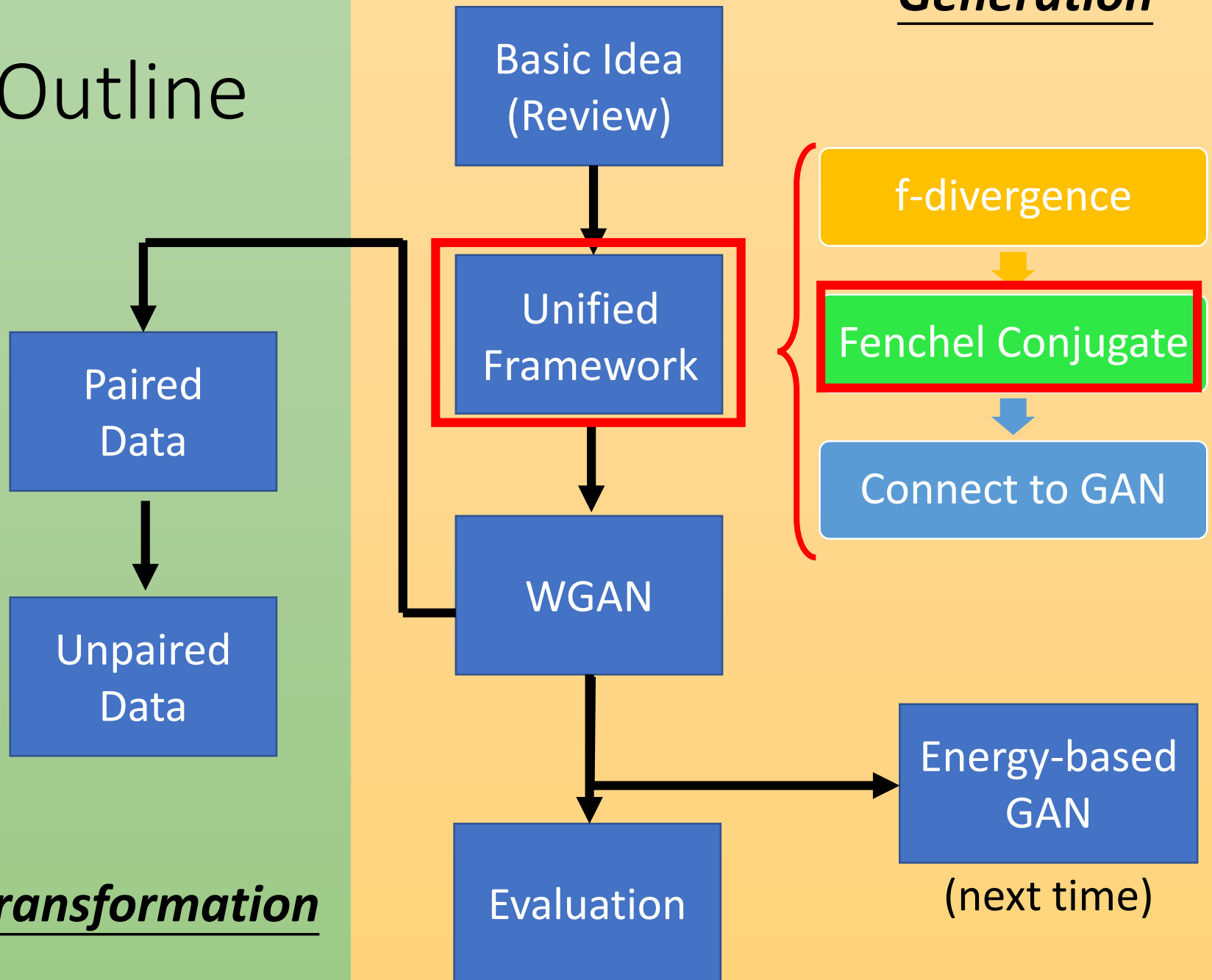
Chi Square

$$D_f(P||Q) = \int_x q(x) \left(\frac{p(x)}{q(x)} - 1 \right)^2 dx = \int_x \frac{(p(x) - q(x))^2}{q(x)} dx$$

Outline

Generation

Transformation



Fenchel Conjugate

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

f is convex, $f(1) = 0$

- Every convex function f has a conjugate function f^*

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$

$$f^*(t_1) = \max_{x \in \text{dom}(f)} \{xt_1 - f(x)\}$$

$$x_1 t_1 - f(x_1)$$

$$x_2 t_1 - f(x_2)$$

$$x_3 t_1 - f(x_3)$$

t_1

$$f^*(t_2) = \max_{x \in \text{dom}(f)} \{xt_2 - f(x)\}$$

$$x_3 t_2 - f(x_3)$$

$$x_2 t_2 - f(x_2)$$

$$x_1 t_2 - f(x_1)$$

t_2

t



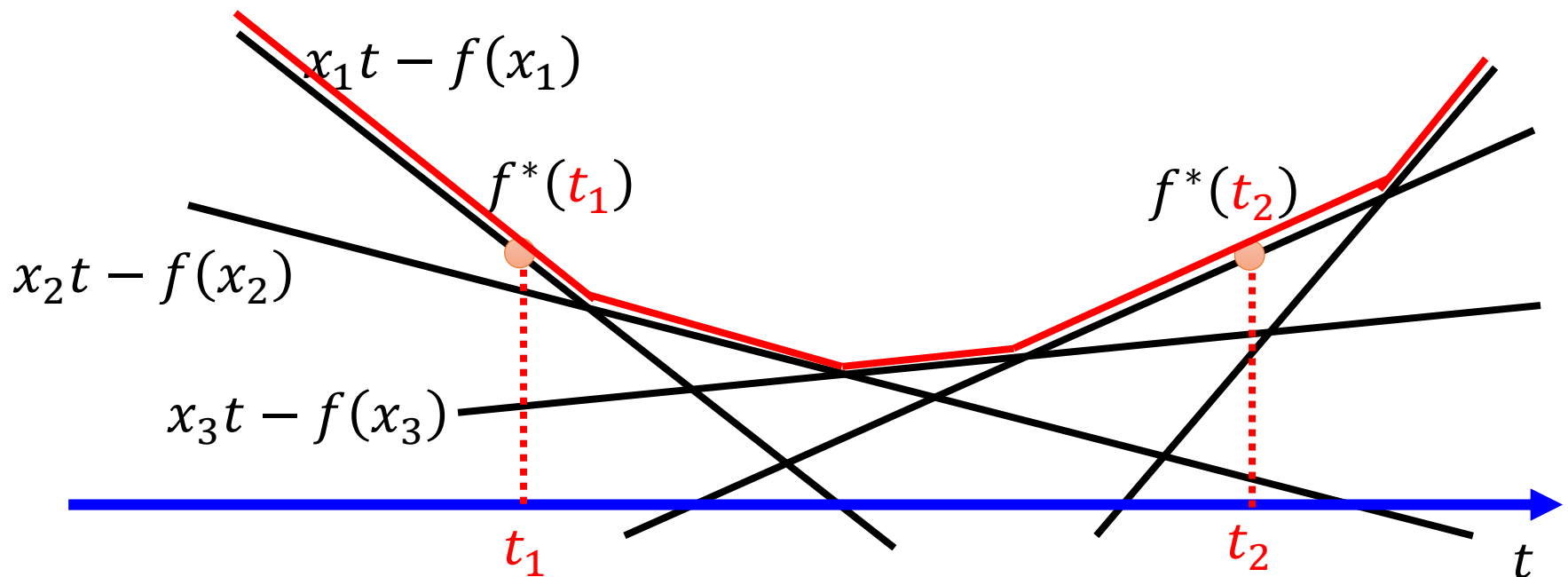
Fenchel Conjugate

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

f is convex, $f(1) = 0$

- Every convex function f has a conjugate function f^*

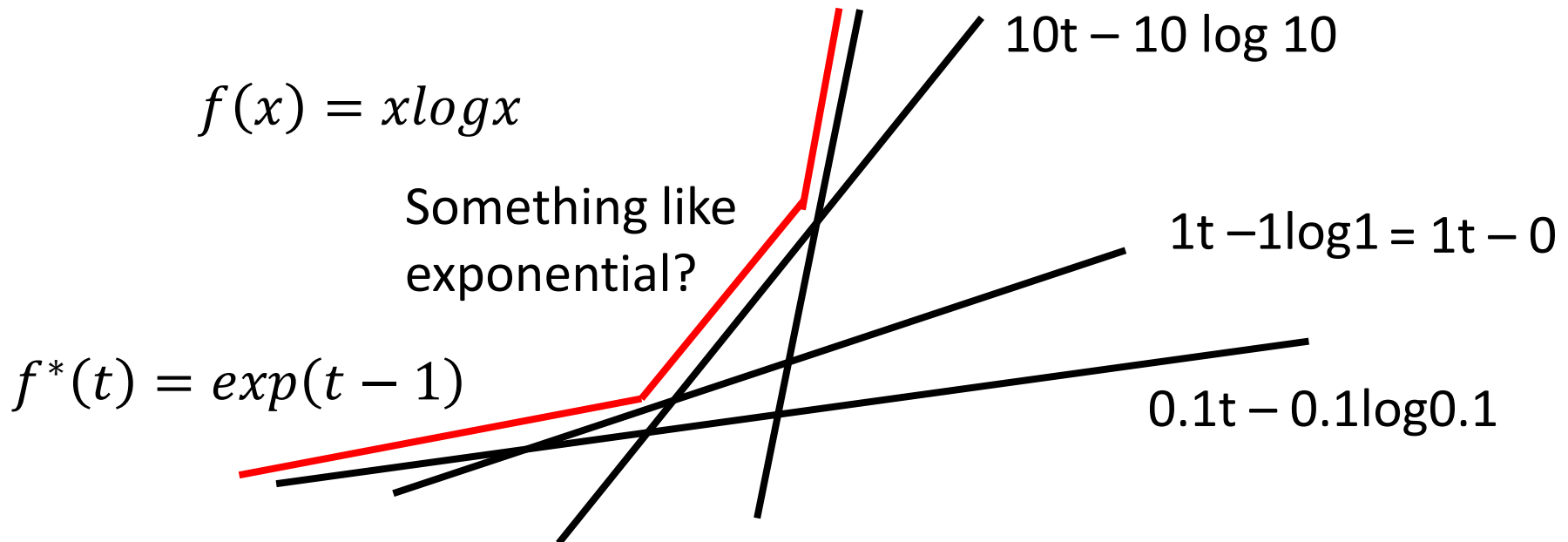
$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$



Fenchel Conjugate

- Every convex function f has a conjugate function f^*

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$



Fenchel Conjugate

- Every convex function f has a conjugate function f^*
- $(f^*)^* = f$

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$

$$f(x) = x \log x \quad \longleftrightarrow \quad f^*(t) = \exp(t - 1)$$

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - x \log x\}$$

$$g(x) = xt - x \log x \quad \text{Given } t, \text{ find } x \text{ maximizing } g(x)$$

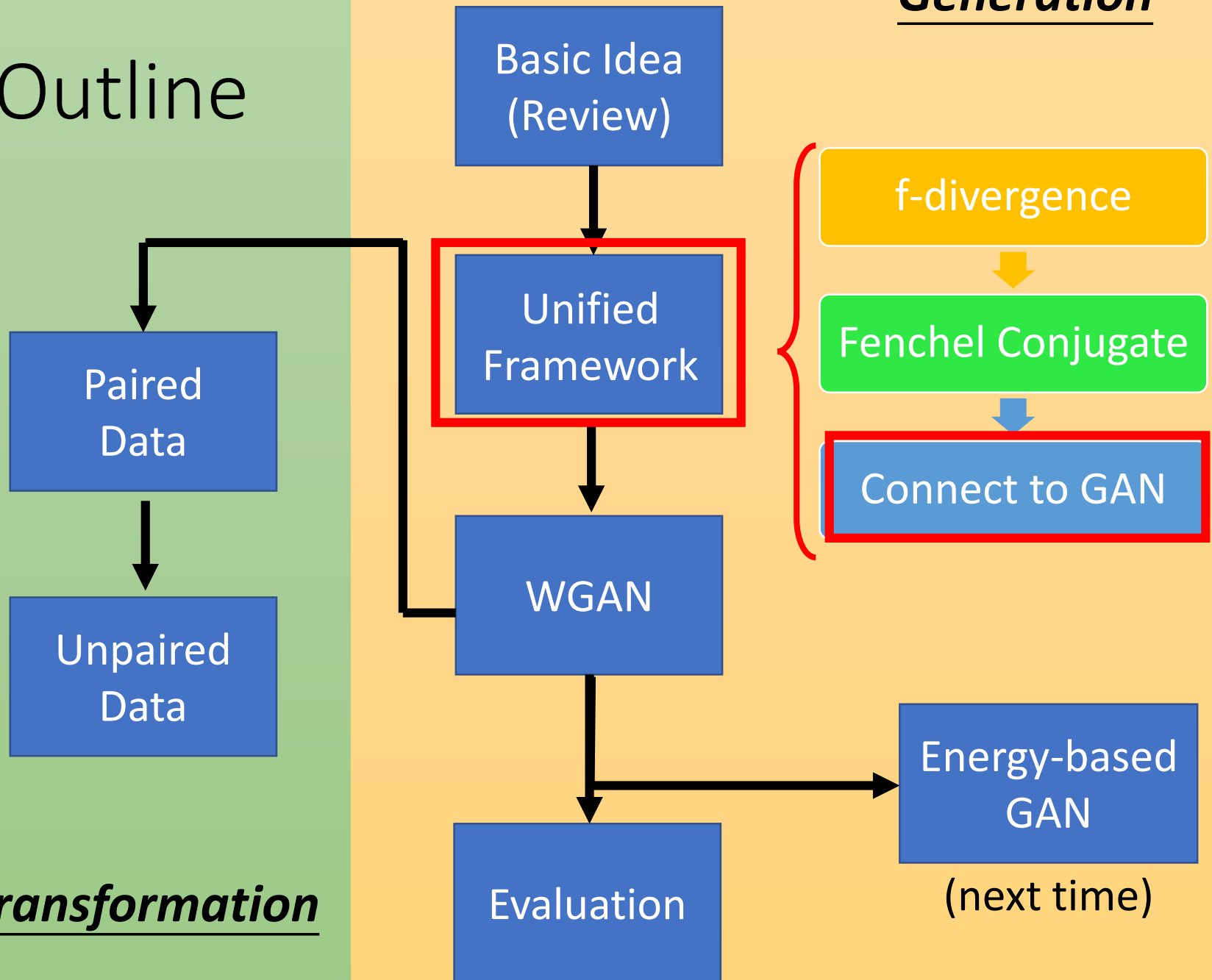
$$t - \log x - 1 = 0 \quad x = \exp(t - 1)$$

$$f^*(t) = \exp(t - 1) \times t - \exp(t - 1) \times (t - 1) = \exp(t - 1)$$

Outline

Generation

Transformation



Connection with GAN

$$f^*(t) = \sup_{x \in \text{dom}(f)} \{xt - f(x)\} \quad \longleftrightarrow \quad f(\underline{x}) = \max_{t \in \text{dom}(f^*)} \{\underline{xt} - f^*(t)\}$$

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx \quad \boxed{\frac{p(x)}{q(x)}} \quad \boxed{\frac{p(x)}{q(x)}}$$

$$= \int_x q(x) \left(\max_{t \in \text{dom}(f^*)} \left\{ \frac{p(x)}{q(x)} \underline{t} - f^*(\underline{t}) \right\} \right) dx$$

$$\approx \max_D \int_x p(x) D(x) dx - \int_x q(x) f^*(D(x)) dx$$

D is a function
whose input is x,
and output is t

$$\begin{aligned} D_f(P||Q) &\geq \int_x q(x) \left(\frac{p(x)}{q(x)} \underline{D(x)} - f^*(\underline{D(x)}) \right) dx \\ &= \int_x p(x) D(x) dx - \int_x q(x) f^*(D(x)) dx \end{aligned}$$

Connection with GAN

$$D_f(P||Q) \approx \max_D \int p(x)D(x)dx - \int q(x)f^*(D(x))dx$$

$$= \max_D \{ E_{x \sim P}[D(x)] - E_{x \sim Q}[f^*(D(x))] \}$$

Samples from P

Samples from Q

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

$$G^* = \arg \min_G D_f(P_{data}||P_G)$$

Original GAN has
different $V(G,D)$

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

$$= \arg \min_G \max_D V(G, D) \text{ familiar? 😊}$$

Double-loop v.s. Single-step

$$G^* = \arg \min_G \max_D V(G, D) \quad \longrightarrow \quad G^* = \arg \min_{\theta_G} \max_{\theta_D} V(\theta_G, \theta_D)$$

- Original paper of GAN: double-loop algorithm

- In each iteration

- Given a generator θ_G^t , $\theta_D^t = \arg \max_{\theta_D} V(\theta_G^t, \theta_D)$

- Update the parameters many times to find θ_D^t

- Update generator once:

- $\theta_G^{t+1} \leftarrow \theta_G^t - \eta \nabla_{\theta_G} V(\theta_G^t, \theta_D^t)$

Inner Loop

- Paper of f-GAN: Single-step algorithm

- In each iteration, given θ_G^t and θ_D^t

- $\theta_D^{t+1} \leftarrow \theta_D^t + \eta \nabla_{\theta_D} \underline{V(\theta_G^t, \theta_D^t)}$

- $\theta_G^{t+1} \leftarrow \theta_G^t - \eta \nabla_{\theta_G} \underline{V(\theta_G^t, \theta_D^t)}$

One Backpropagation

Outer Loop

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

Name	$D_f(P Q)$	Generator $f(u)$
Total variation	$\frac{1}{2} \int p(x) - q(x) \, dx$	$\frac{1}{2} u - 1 $
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \, dx$	$u \log u$
Reverse Kullback-Leibler	$\int q(x) \log \frac{q(x)}{p(x)} \, dx$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \, dx$	$(u - 1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)} \, dx$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 \, dx$	$(\sqrt{u} - 1)^2$
Jeffrey	$\int (p(x) - q(x)) \log \left(\frac{p(x)}{q(x)} \right) \, dx$	$(u - 1) \log u$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, dx$	$-(u + 1) \log \frac{1+u}{2} + u \log u$
Jensen-Shannon-weighted	$\int p(x) \pi \log \frac{p(x)}{\pi p(x)+(1-\pi)q(x)} + (1 - \pi)q(x) \log \frac{q(x)}{\pi p(x)+(1-\pi)q(x)} \, dx$	$\pi u \log u - (1 - \pi + \pi u) \log(1 - \pi + \pi u)$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, dx - \log(4)$	$u \log u - (u + 1) \log(u + 1)$

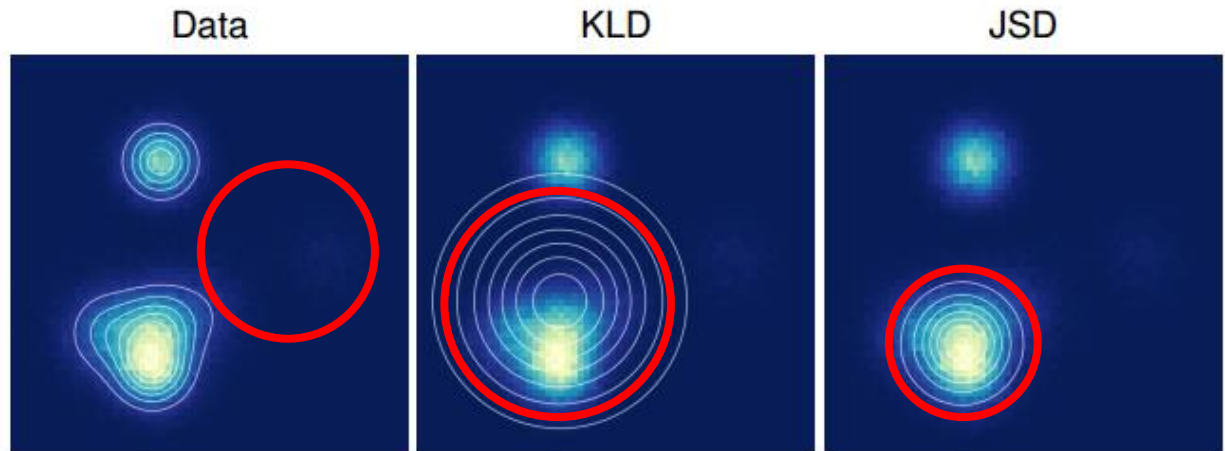
Name	Conjugate $f^*(t)$
Total variation	t
Kullback-Leibler (KL)	$\exp(t - 1)$
Reverse KL	$-1 - \log(-t)$
Pearson χ^2	$\frac{1}{4}t^2 + t$
Neyman χ^2	$2 - 2\sqrt{1 - t}$
Squared Hellinger	$\frac{t}{1-t}$
Jeffrey	$W(e^{1-t}) + \frac{1}{W(e^{1-t})} + t - 2$
Jensen-Shannon	$-\log(2 - \exp(t))$
Jensen-Shannon-weighted	$(1 - \pi) \log \frac{1-\pi}{1-\pi e^{t/\pi}}$
GAN	$-\log(1 - \exp(t))$

Using the f-divergence
you like ☺

<https://arxiv.org/pdf/1606.00709.pdf>

Experimental Results

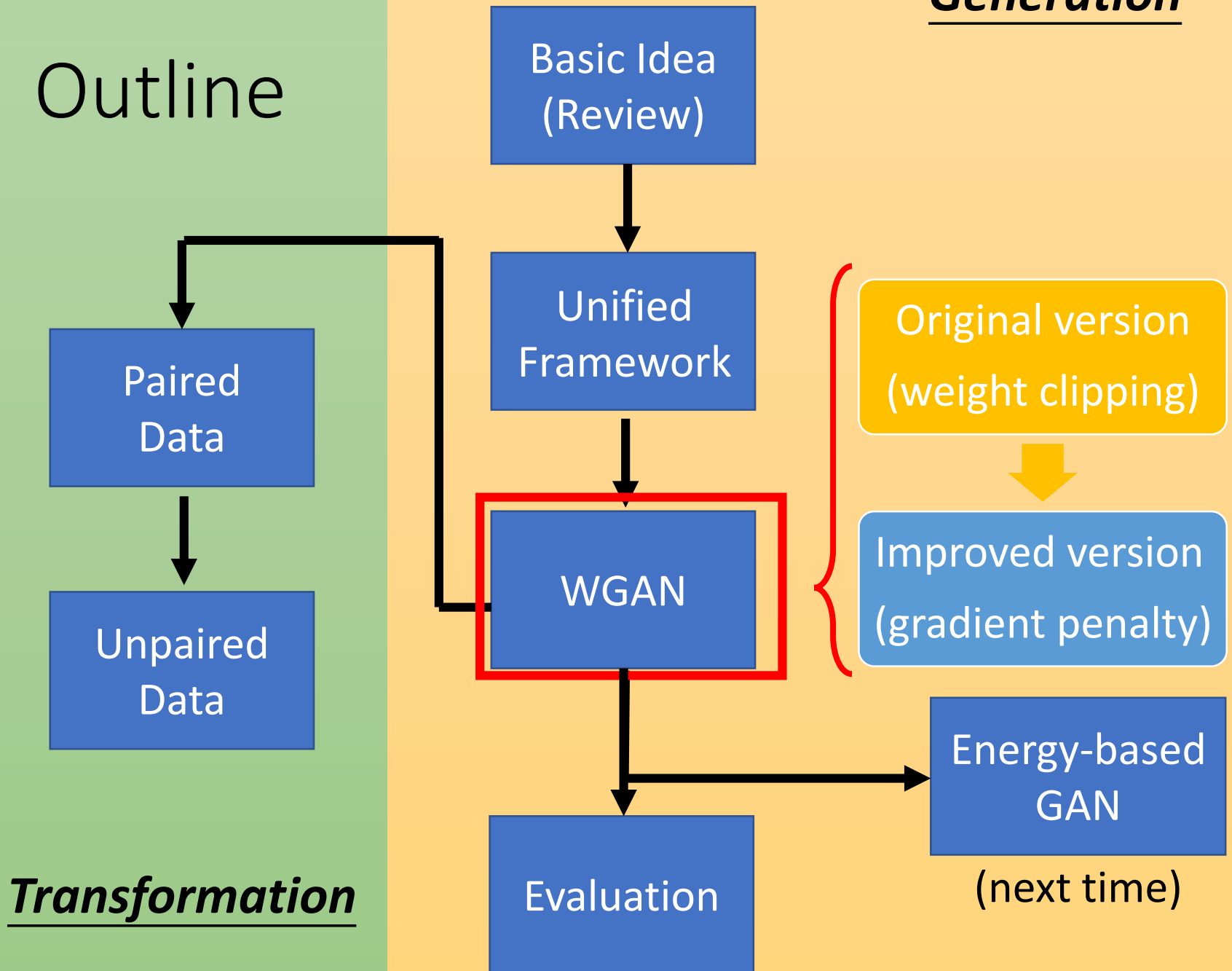
- Approximate a mixture of Gaussians by single mixture



train \ test	KL	KL-rev	JS	Jeffrey	Pearson
KL	0.2808	0.3423	0.1314	0.5447	0.7345
KL-rev	0.3518	0.2414	0.1228	0.5794	1.3974
JS	0.2871	0.2760	0.1210	0.5260	0.92160
Jeffrey	0.2869	0.2975	0.1247	0.5236	0.8849
Pearson	0.2970	0.5466	0.1665	0.7085	0.648

Generation

Outline

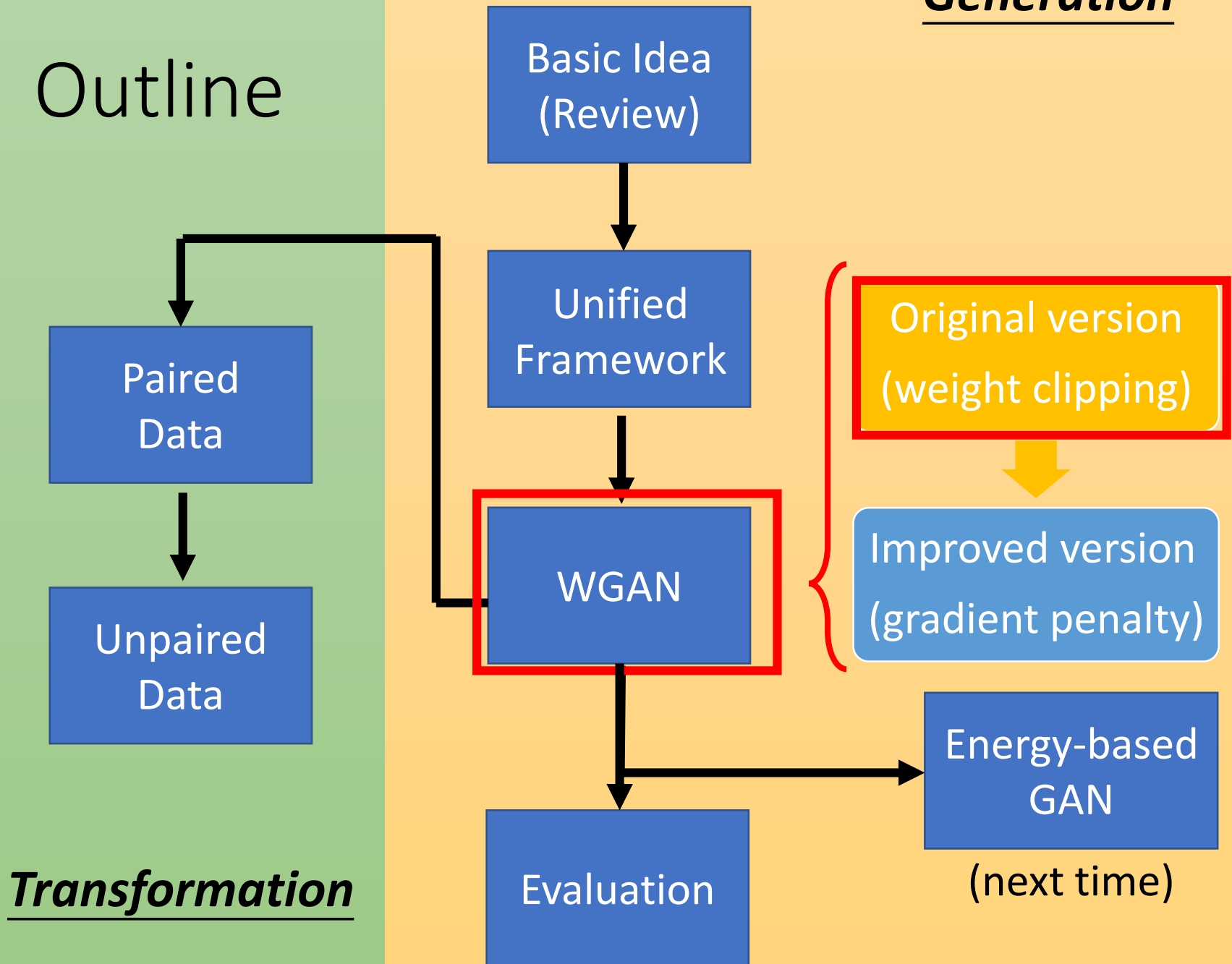


Reference

- 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
- One sentence for WGAN: Using Earth Mover’s Distance to evaluate two distributions
 - Earth Mover’s Distance = Wasserstein Distance

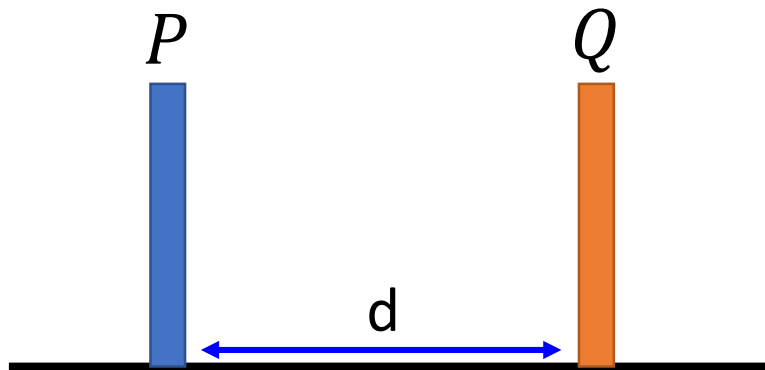
Generation

Outline



Earth Mover's Distance

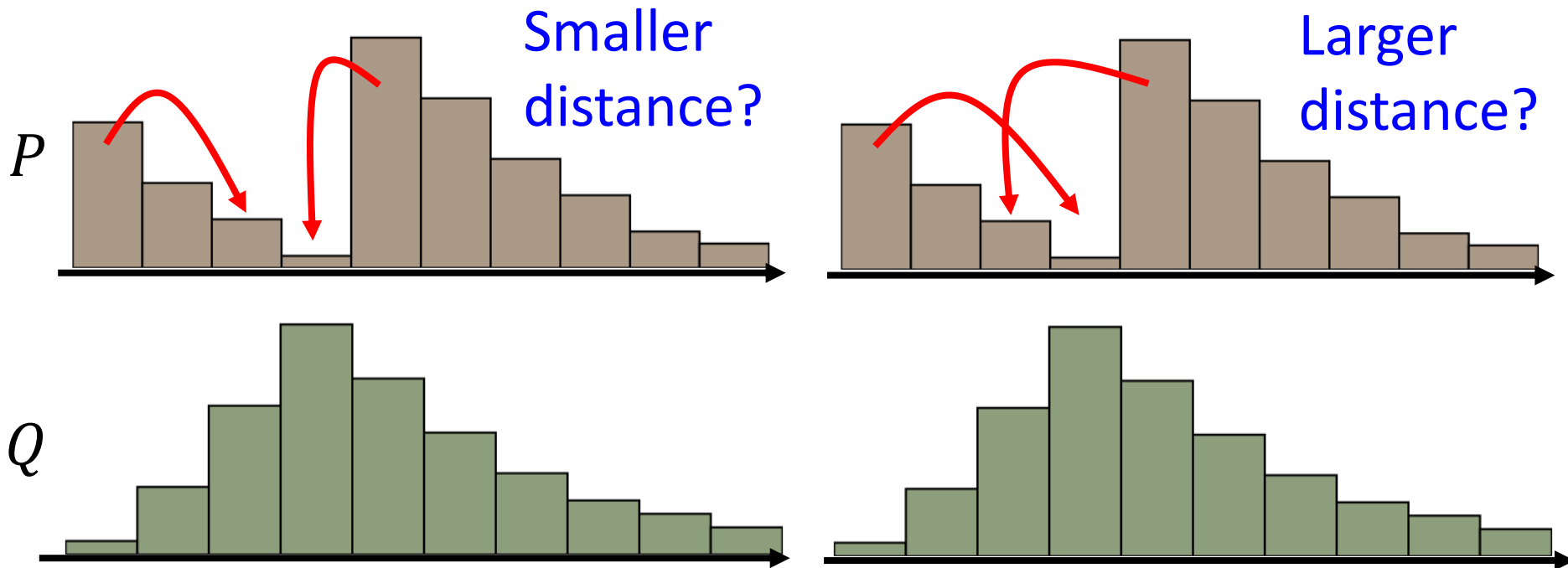
- Considering one distribution P as a pile of earth, and another distribution Q as the target
- The average distance the earth mover has to move the earth.



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



Earth Mover's Distance

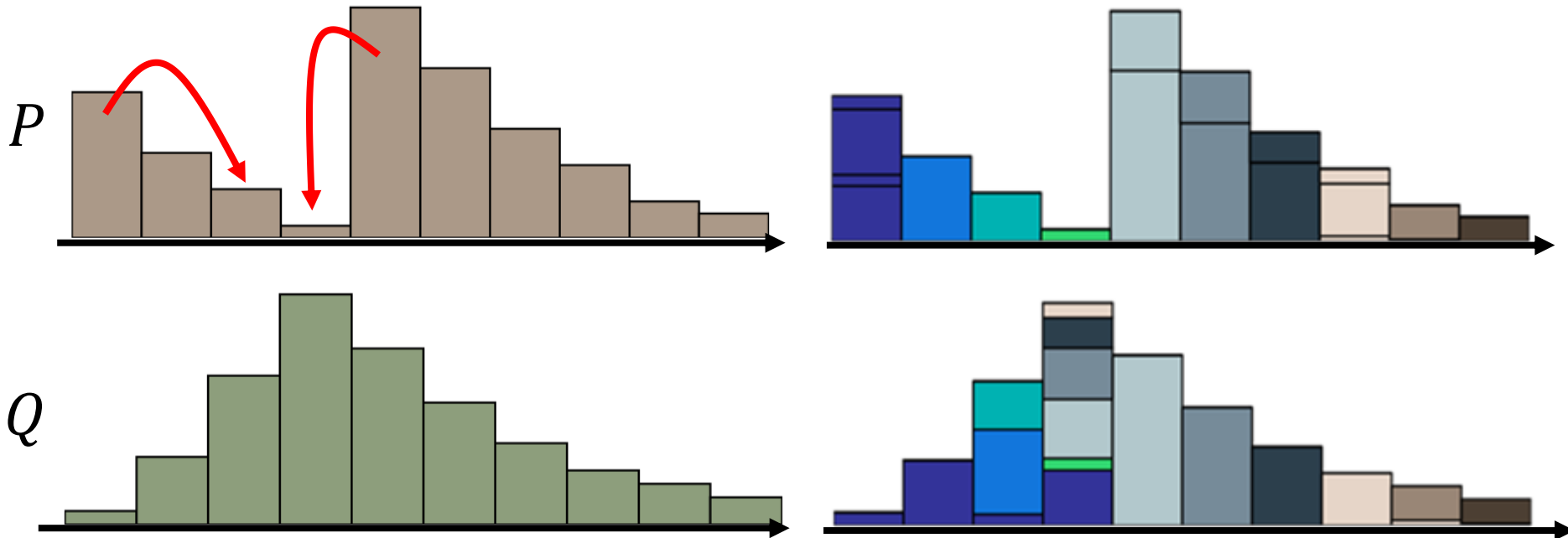


There many possible “moving plans”.

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

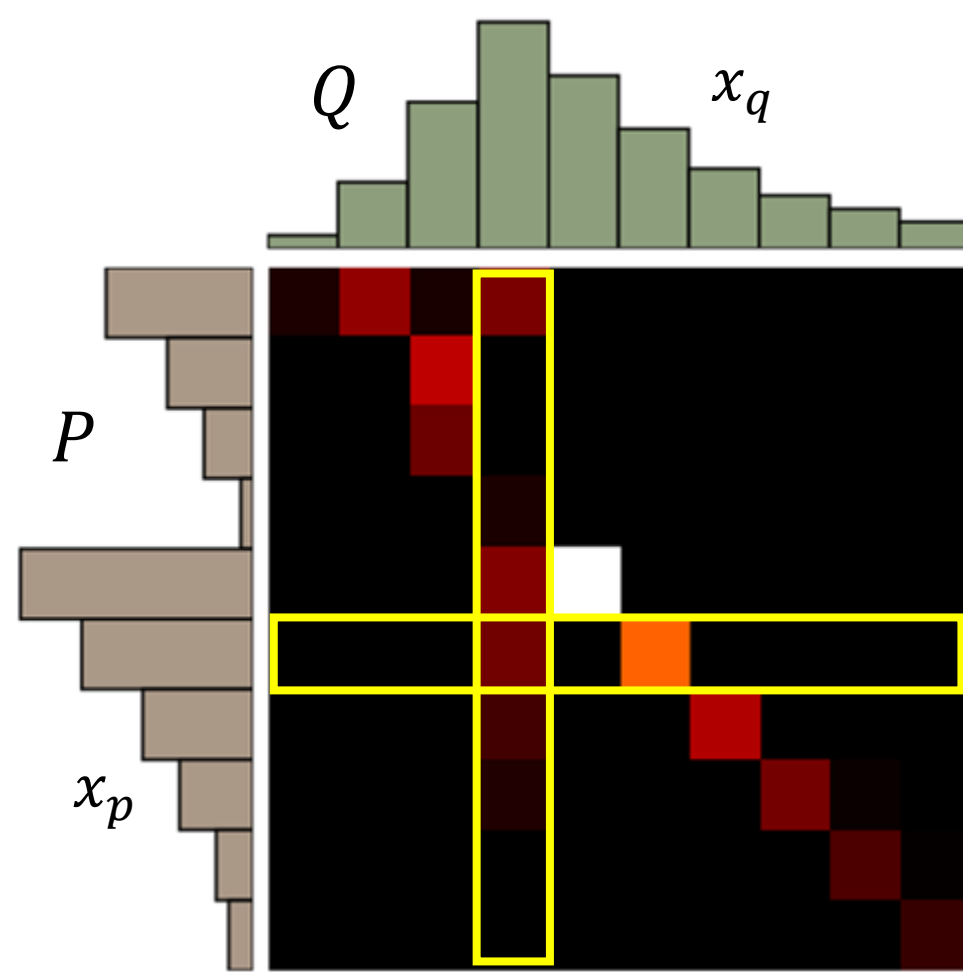
Earth Mover's Distance

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.



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.

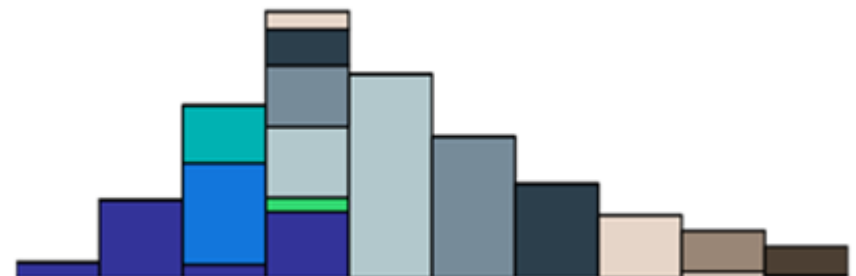
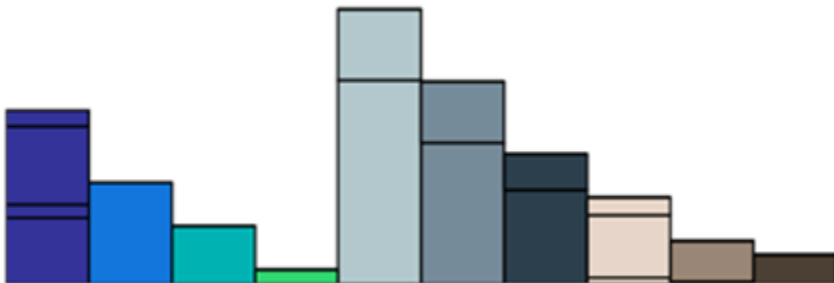
Average distance of a plan γ :

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

Earth Mover’s Distance:

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

The best plan

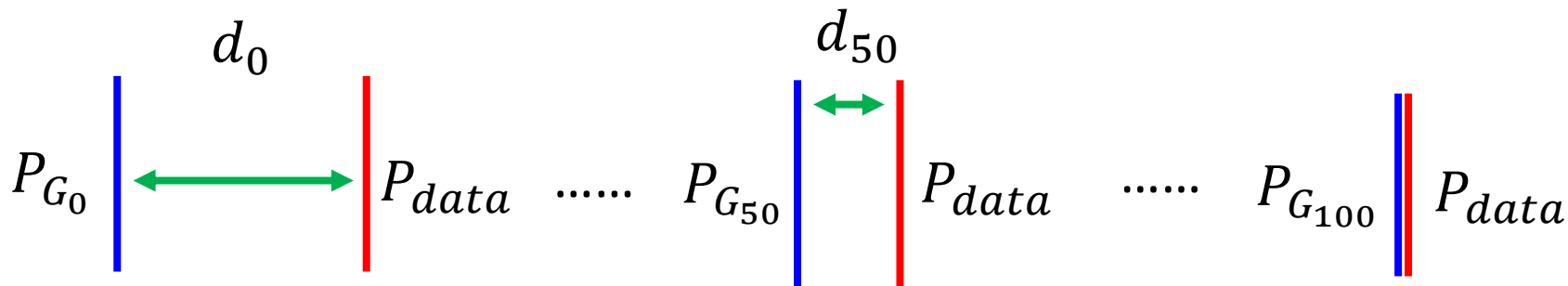
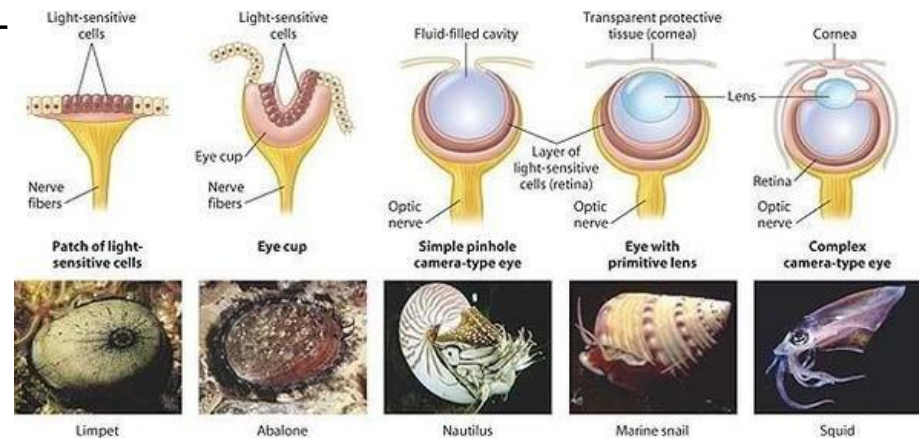


Why Earth Mover's Distance?

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



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



$$JS(P_{G_0}, P_{data}) = \log 2$$

$$JS(P_{G_{50}}, P_{data}) = \log 2$$

$$JS(P_{G_{100}}, P_{data}) = 0$$

$$W(P_{G_0}, P_{data}) = d_0$$

$$W(P_{G_{50}}, P_{data}) = d_{50}$$

$$W(P_{G_{100}}, P_{data}) = 0$$

Back to the GAN framework

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

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

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

Lipschitz Function

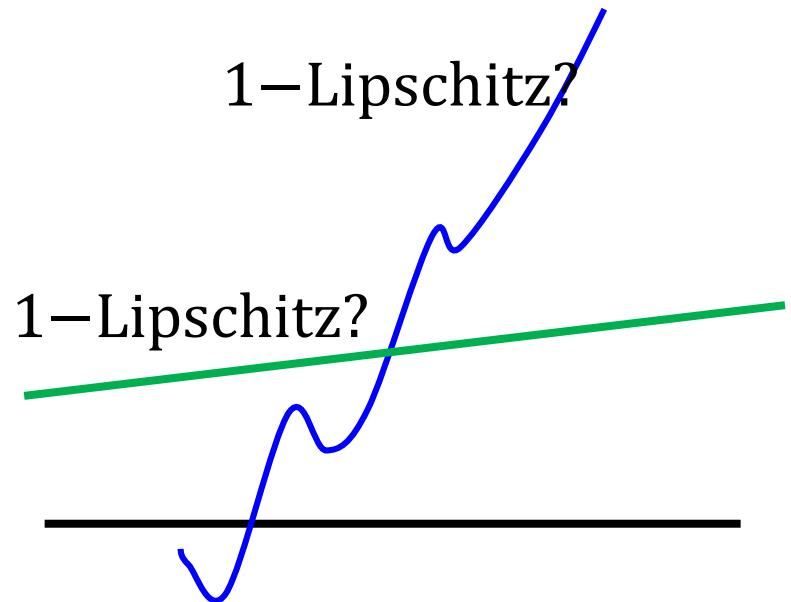
$$\|f(x_1) - f(x_2)\| \leq K \|x_1 - x_2\|$$

Output
change

Input
change

$K=1$ for "1 - Lipschitz"

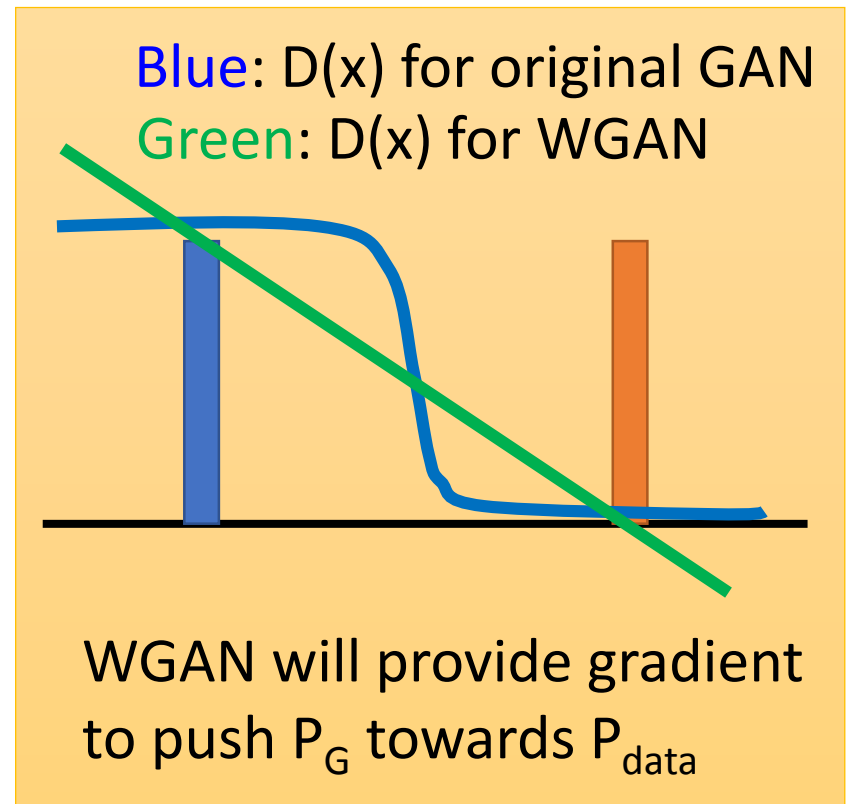
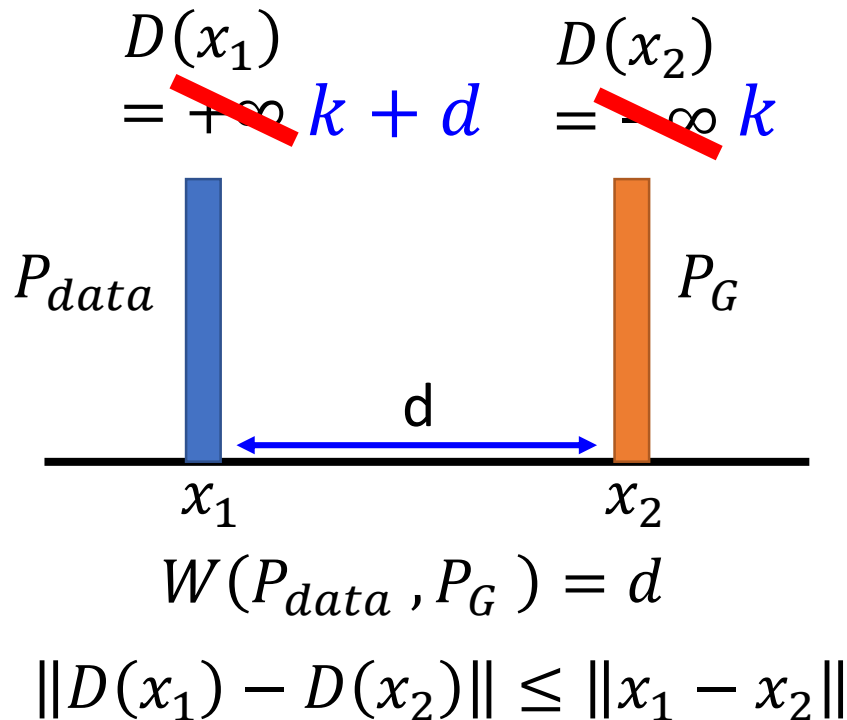
Do not change fast



Back to the GAN framework

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

$k + d$ k



Back to the GAN framework

$$\textcolor{red}{K} \ W(P_{data}, P_G)$$

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

How to use gradient descent to optimize?

Weight clipping:

Force the weights w between c and $-c$

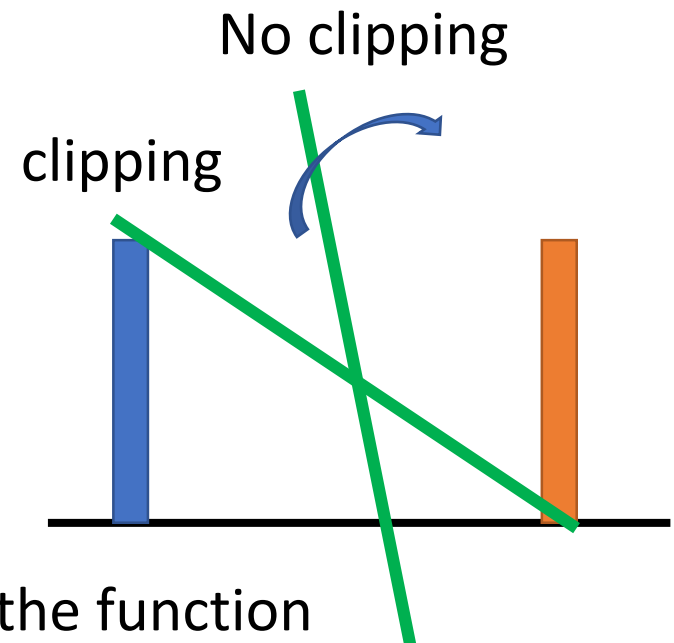
After parameter update,
if $w > c$, then $w=c$; if $w < -c$, then $w=-c$

We only ensure that

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

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 $\{x^1, x^2, \dots, x^m\}$ from data distribution $P_{data}(x)$
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from the prior $P_{prior}(z)$
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- 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 $\{z^1, z^2, \dots, z^m\}$ from the prior $P_{prior}(z)$
- Update generator parameters θ_g to minimize
 - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log D(x^i) - \frac{1}{m} \sum_{i=1}^m D(G(z^i))$
 - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

CNN generator:



W-GAN

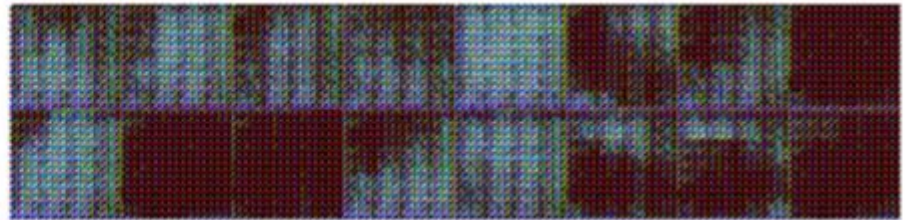


GAN

CNN generator (no batch normalization, bad structure):

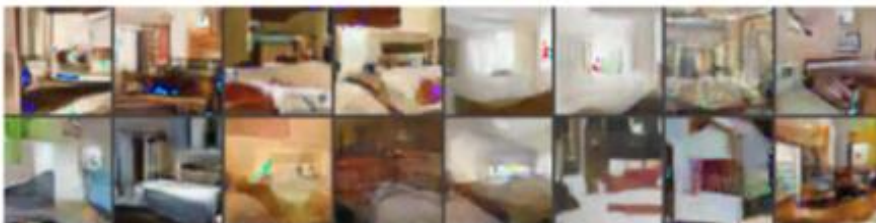


W-GAN



GAN

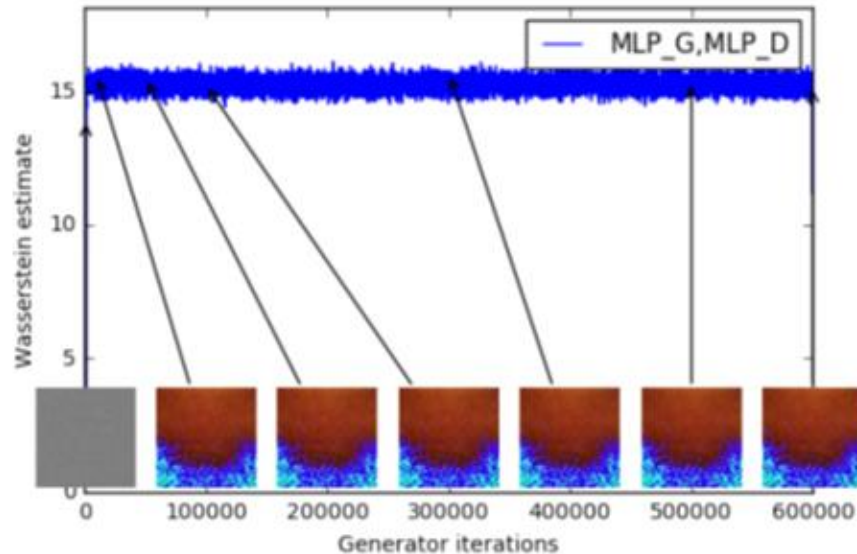
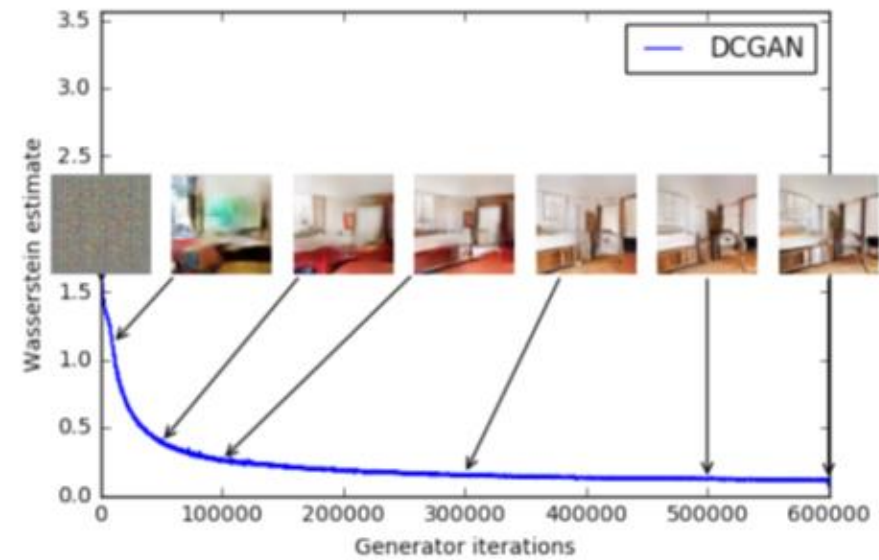
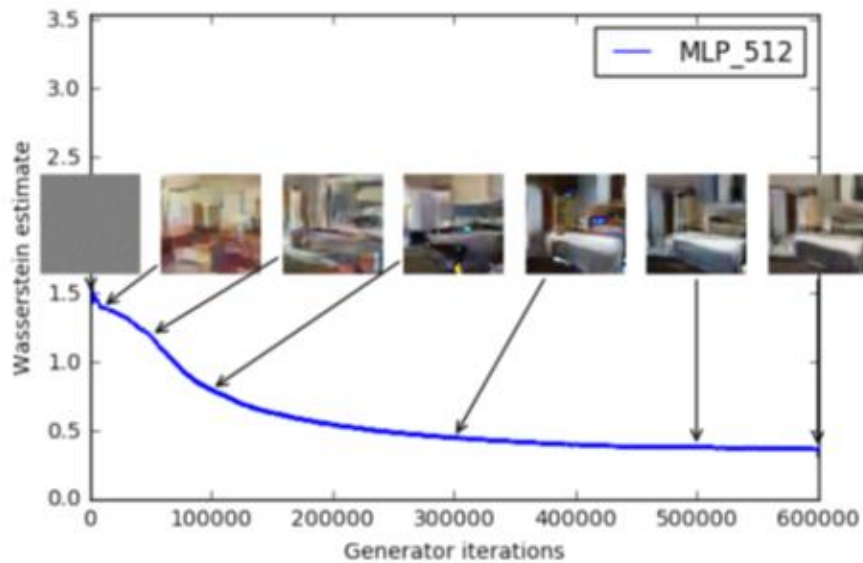
MLP generator:



W-GAN



GAN

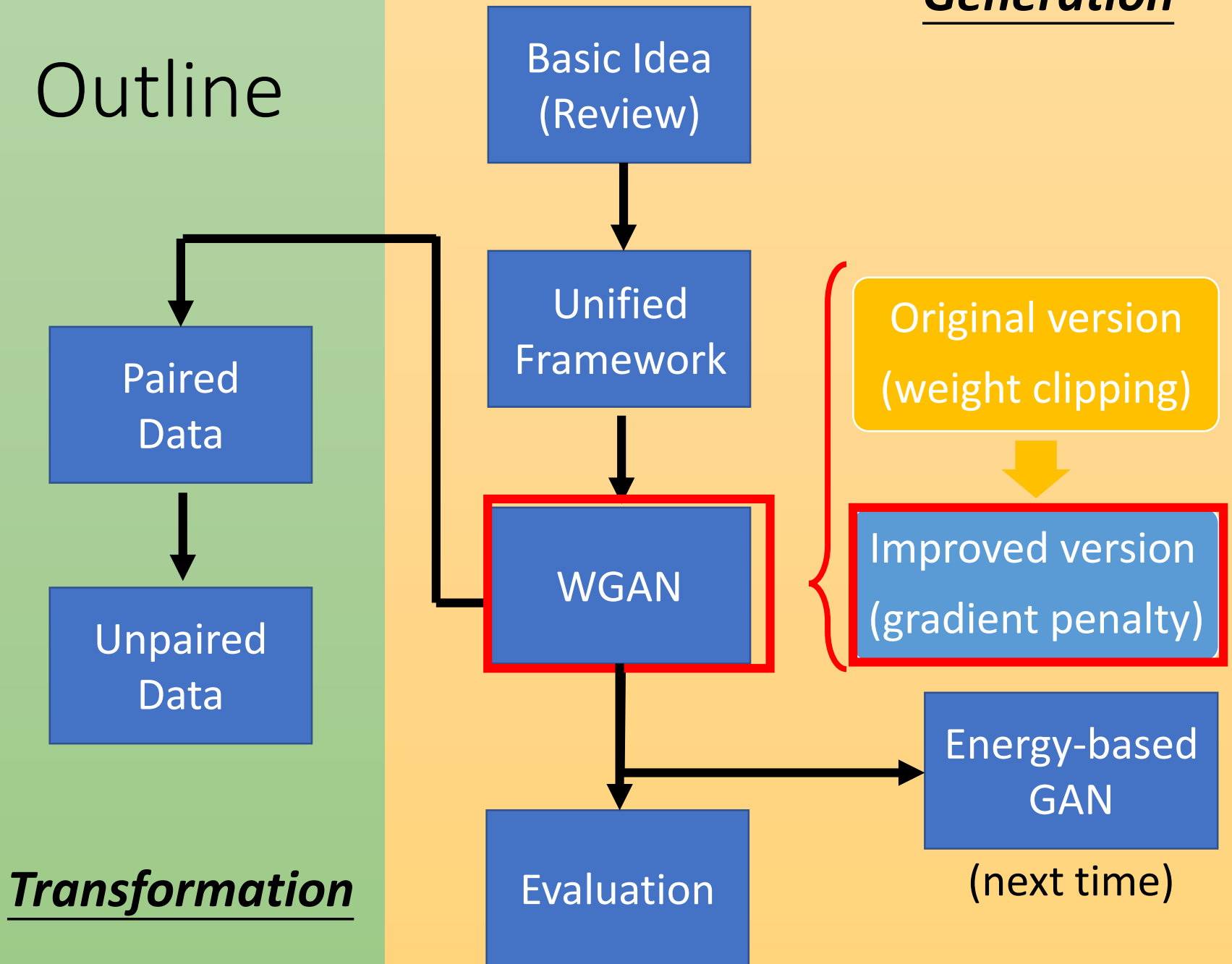


Vertical

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

Generation

Outline



Improved WGAN

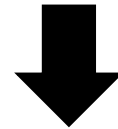
$$W(P_{data}, P_G) = \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

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

$$D \in 1-Lipschitz \iff \|\nabla_x D(x)\| \leq 1 \text{ for all } x$$

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

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

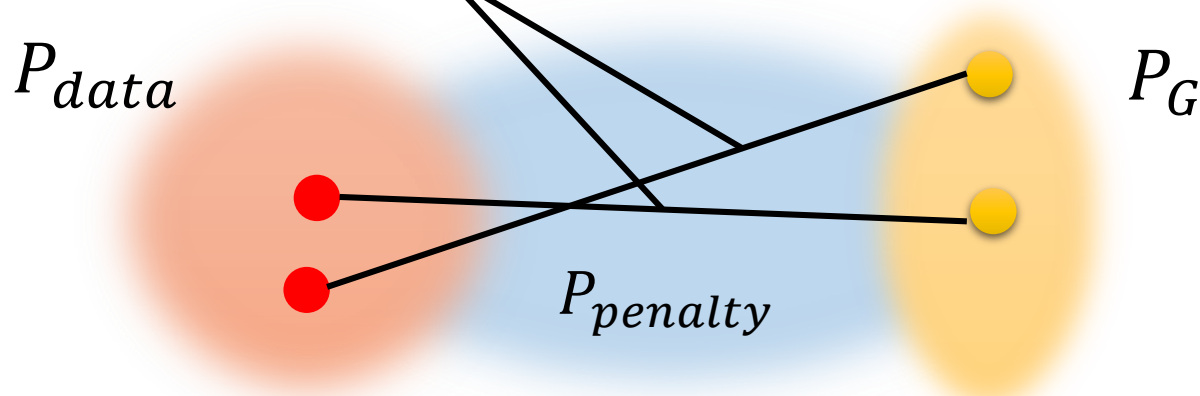


$$- \lambda E_{x \sim P_{penalty}} [\max(0, \|\nabla_x D(x)\| - 1)]$$

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

Improved WGAN

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



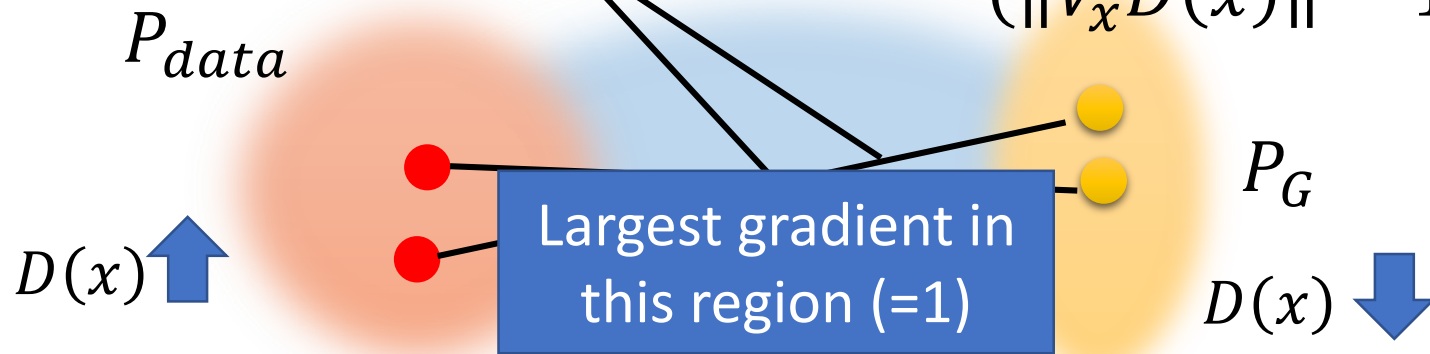
“Given that enforcing the Lipschitz constraint everywhere is intractable, enforcing it **only along these straight lines** seems sufficient and experimentally results in good performance.”

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

Improved WGAN

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

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

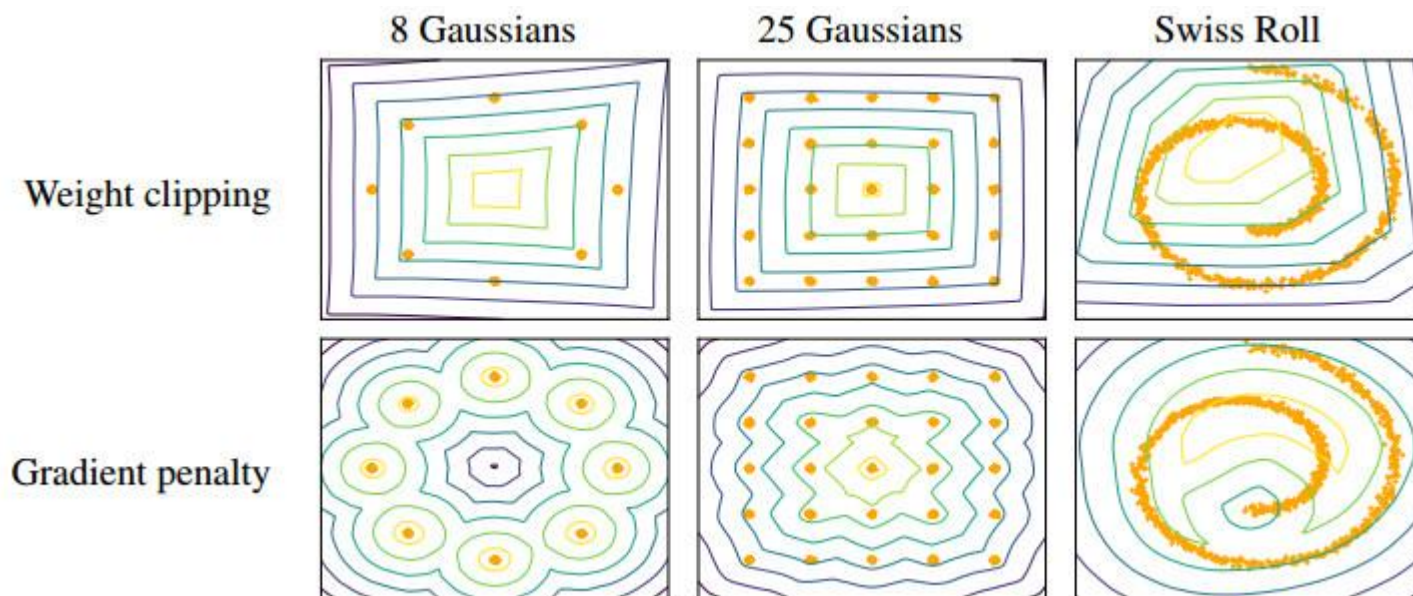
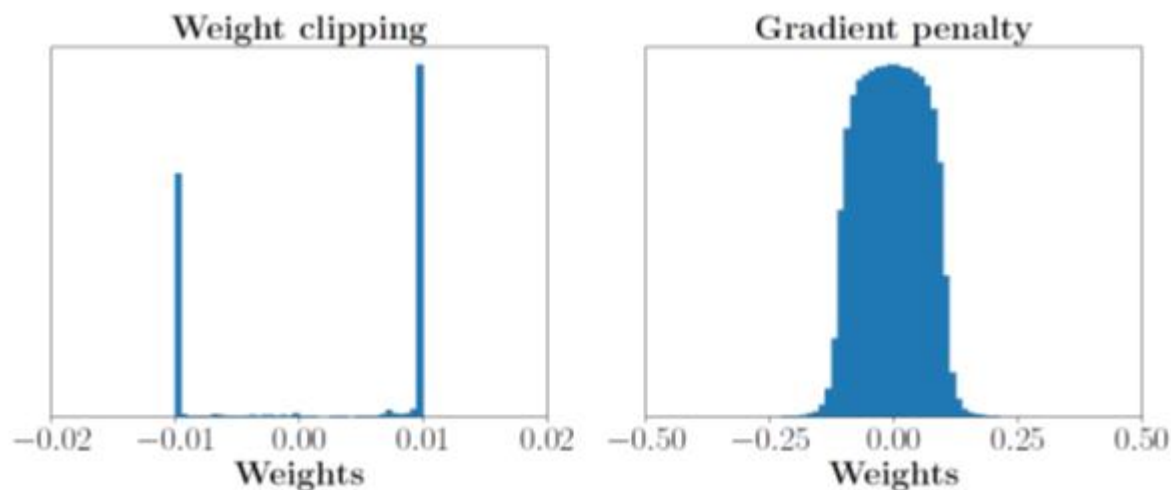


“One may wonder why we penalize the norm of the gradient for differing from 1, instead of just penalizing large gradients. The reason is that the optimal critic ... actually has gradients with norm 1 almost everywhere under P_r and P_g ”

(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.”

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)



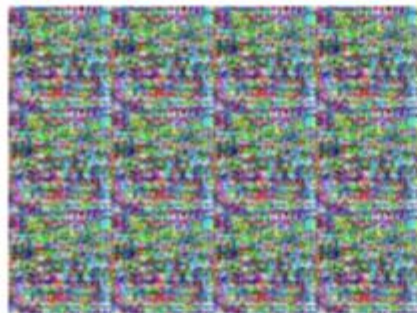
DCGAN

LSGAN

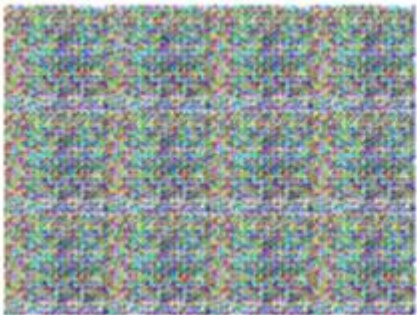
Original
WGAN

Improved
WGAN

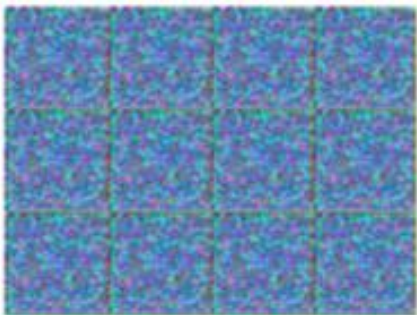
G: MLP, D: CNN



G: CNN (bad structure), D: CNN

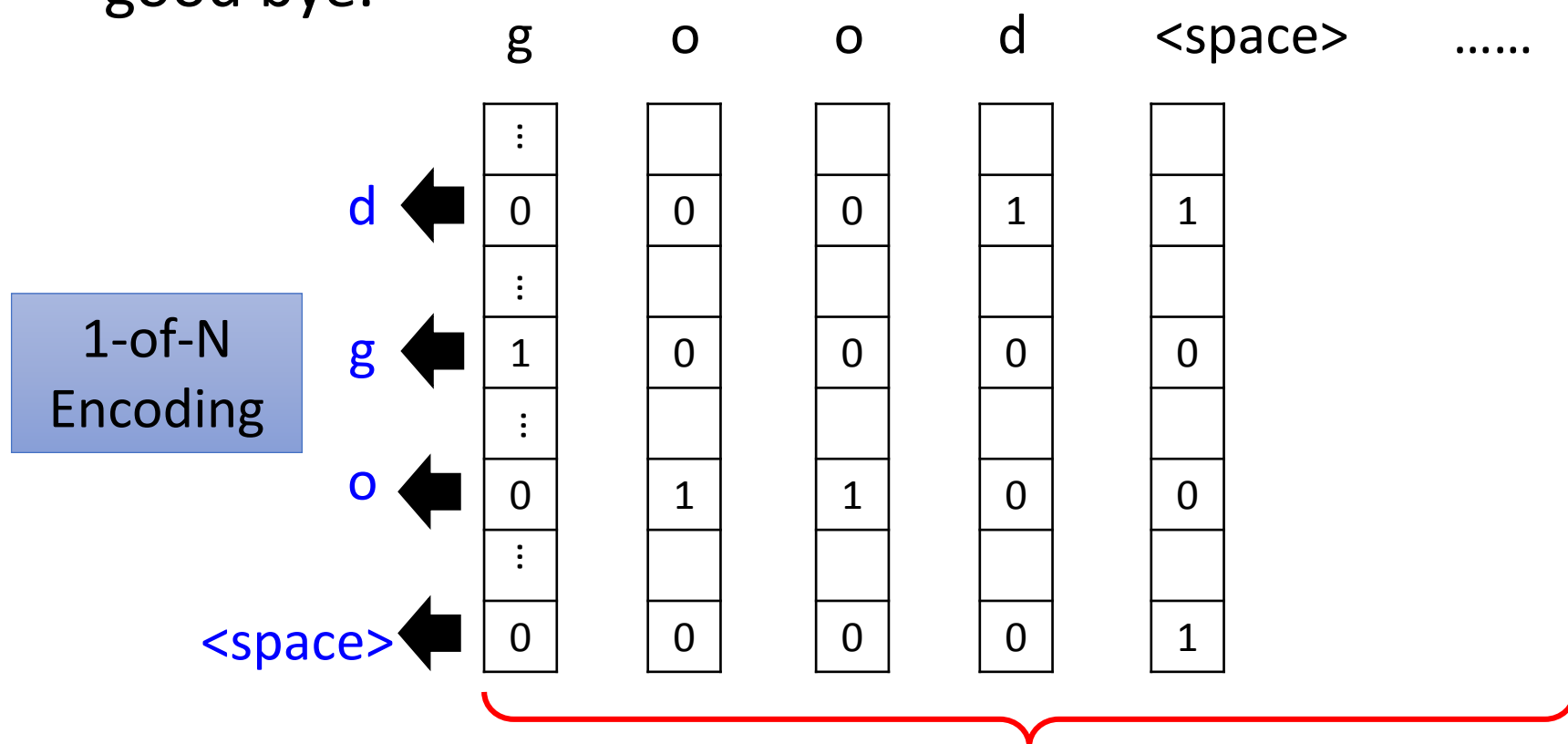


G: 101 layer, D: 101 layer



Sentence Generation

- good bye.



Consider this matrix as an "image"

Sentence Generation

- Real sentence

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

A binary classifier
can immediately
find the difference.

- Generated

0.9	0.1	0.1	0	0
0.1	0.9	0.1	0	0
0	0	0.7	0.1	0
0	0	0.1	0.8	0.1
0	0	0	0.1	0.9

No overlap

Can never
be 1-of-N

WGAN is helpful

Improved WGAN successfully generating sentences

<https://arxiv.org/abs/1704.00028>

WGAN with gradient penalty

Busino game camperate spent odea
In the bankaway of smarling the
SingersMay , who kill that invic
Keray Pents of the same Reagun D
Manging include a tudancs shat "
His Zuith Dudget , the Denmbern
In during the Uitational questio
Divos from The ' noth ronkies of
She like Monday , of macunsuer S
The investor used ty the present
A papees are cointy congress oo
A few year inom the group that s
He said this syenn said they wan
As a world 1 88 ,for Autouries
Foand , th Word people car , Il
High of the upseader homing pull
The guipe is worly move dogsfor
The 1874 incidested he could be
The allo tooks to security and c

Solice Norkedin pring in since
ThiS record (31.) UBS) and Ch
It was not the annuas were plogr
This will be us , the ect of DAN
These leaded as most-worsd p2 a0
The time I paid0a South Cubry i
Dour Fraps higs it was these del
This year out howneed allowed lo
Kaulna Seto consficutes to repor
A can teal , he was schoon news
In th 200. Pesish picriers rega
Konney Panice rimimber the teami
The new centuct cut Denester of
The near , had been one injostie
The inceston to week to shorted
The company the high product of
20 - The time of accomplete , wh
John WVuderenson seqiivic spends
A ceetens in indestredly the Wat

感謝 李仲翊 同學提供實
驗結果

W-GAN – 唐詩鍊成

輸出 32 個字 (包含標點)

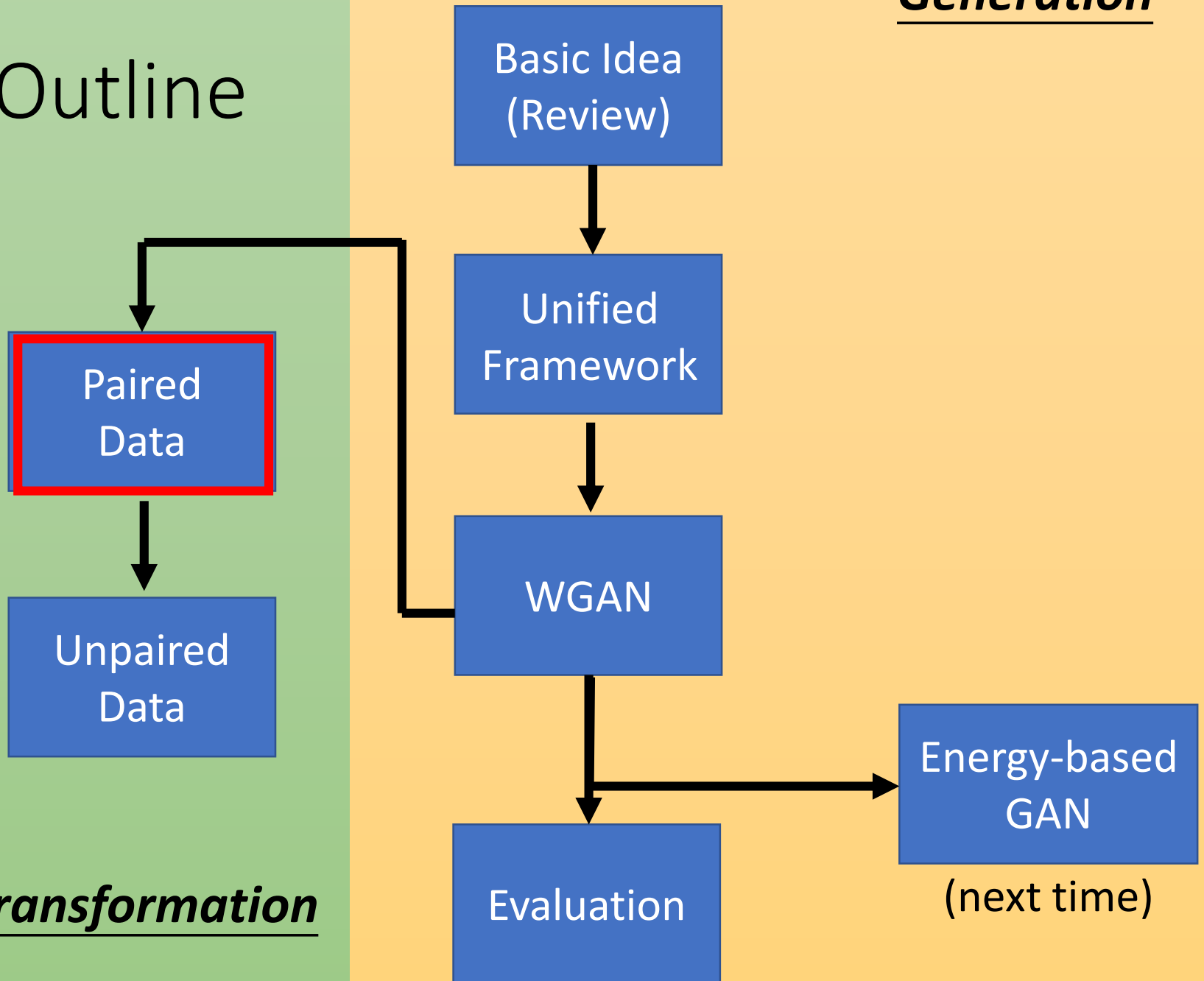
- 升雲白遲丹齋取，此酒新巷市入頭。黃道故海歸中後，不驚入得韻子門。
- 據口容章蕃翎翎，邦貸無遊隔將毬。外蕭曾臺遠出畧，此計推上呂天夢。
- 新來寶伎泉，手雪泓臺蓑。曾子花路魏，不謀散薦船。
- 功持牧度機邈爭，不躔官嬉牧涼散。不迎白旅今掩冬，盡蘸金祇可停。
- 玉十洪沄爭春風，溪子風佛挺橫鞋。盤盤稅焰先花齋，誰過飄鶴一丞幢。
- 海人依野庇，為阻例沉迴。座花不佐樹，弟闌十名儂。
- 入維當興日世瀕，不評皺。頭醉空其杯，駸園凋送頭。
- 鉢笙動春枝，寶叅潔長知。官為密爛去，絆粒薛一靜。
- 吾涼腕不楚，縱先待旅知。楚人縱酒待，一蔓飄聖猜。
- 折幕故癘應韻子，徑頭霜瓊老徑徑。尚錯春鏘熊悽梅，去吹依能九將香。
- 通可矯目鸚須淨，丹迤挈花一抵嫖。外子當目中前醒，迎日幽筆鈎弧前。
- 庭愛四樹人庭好，無衣服仍繡秋州。更怯風流欲鳩雲，帛陽舊據畝婷儻。

More about Discrete Output

- SeqGAN
 - Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, AAAI, 2017
 - Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, Dan Jurafsky, Adversarial Learning for Neural Dialogue Generation, arXiv preprint, 2017
- Boundary seeking GAN
 - R Devon Hjelm, Athul Paul Jacob, Tong Che, Kyunghyun Cho, Yoshua Bengio, “Boundary-Seeking Generative Adversarial Networks”, arXiv preprint, 2017
- Gumbel-Softmax
 - Matt J. Kusner, José Miguel Hernández-Lobato, GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution, arXiv preprint, 2016
 - Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, Yoshua Bengio, Maximum-Likelihood Augmented Discrete Generative Adversarial Networks, arXiv preprint, 2017

Generation

Outline



Transformation

Conditional GAN

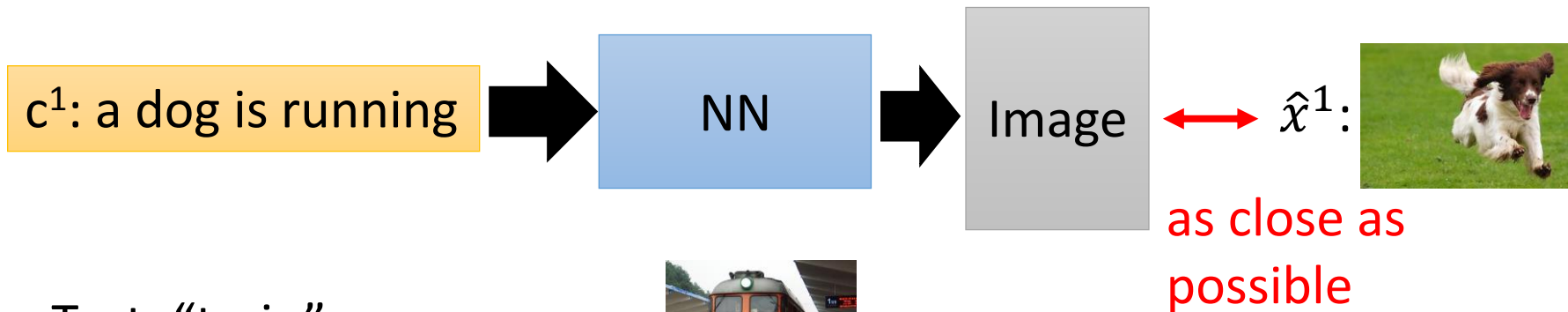
c^1 : a dog is running \hat{x}^1 :



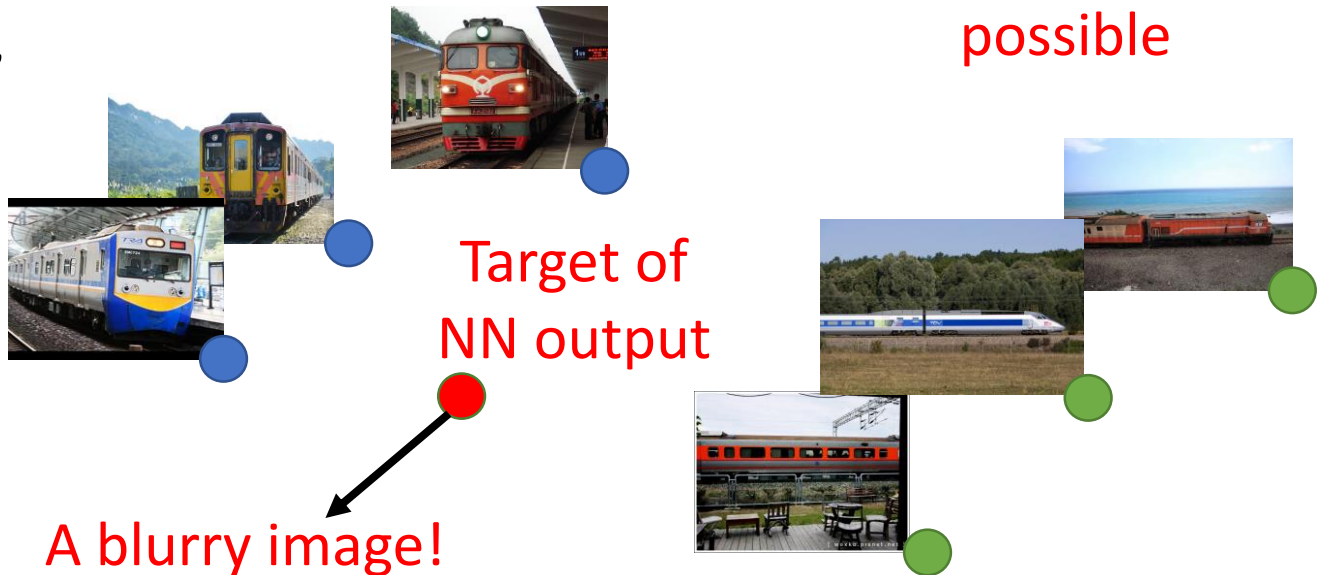
c^2 : a bird is flying \hat{x}^2 :



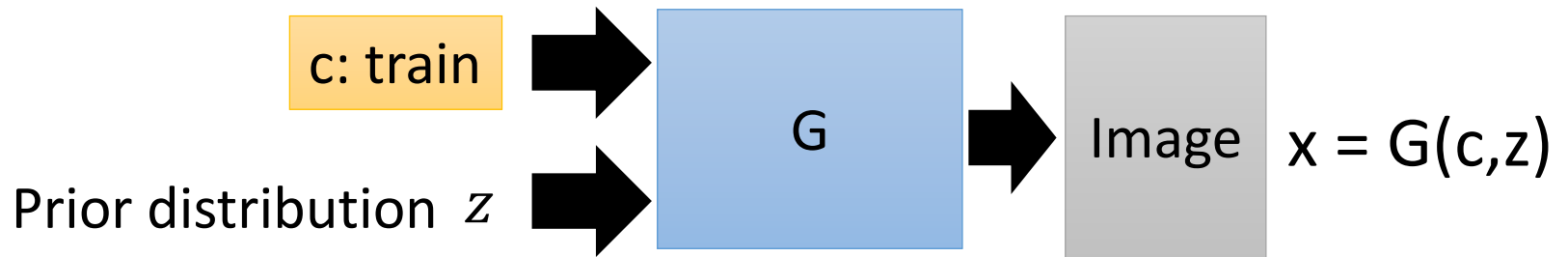
- **Text to image** by traditional supervised learning



Text: "train"



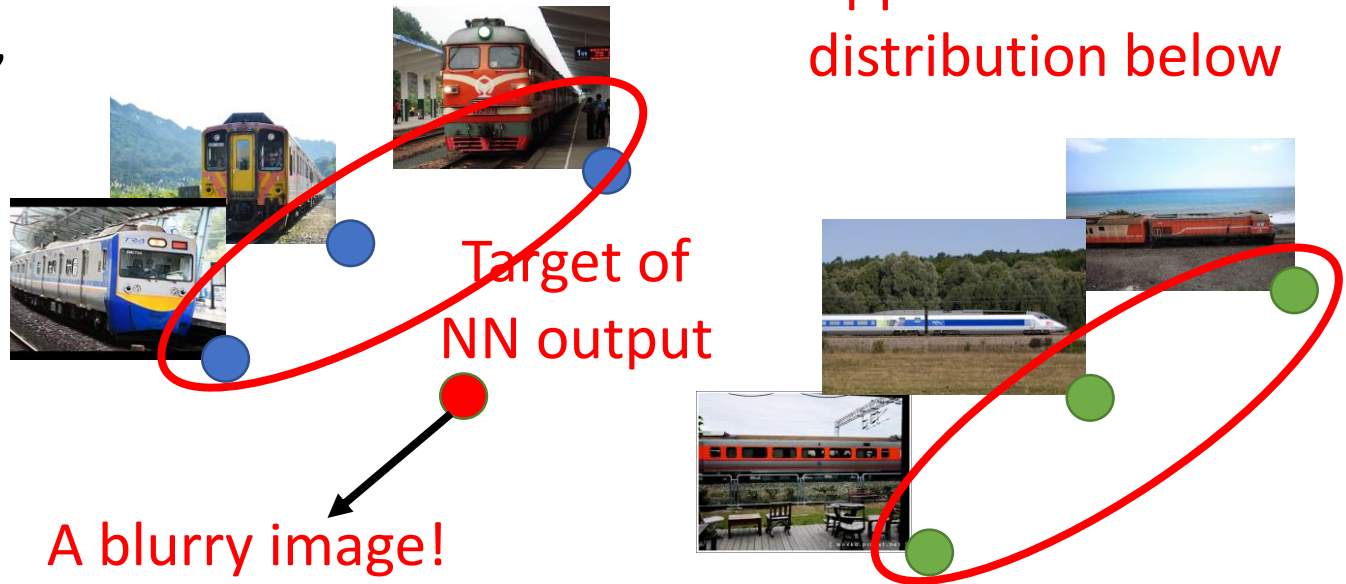
Conditional GAN



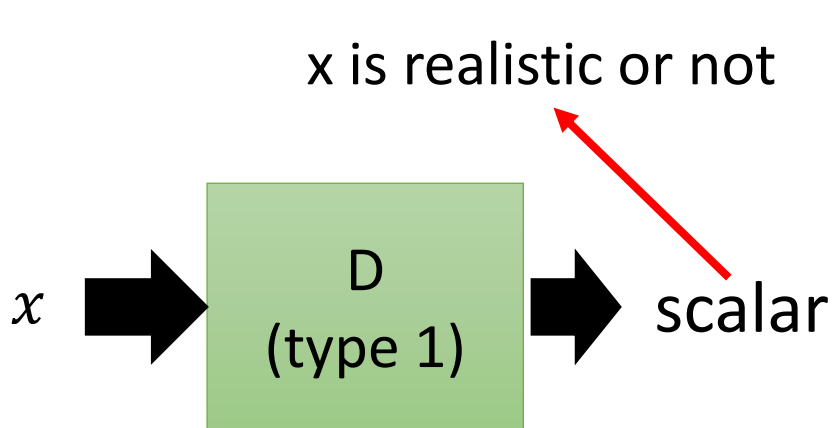
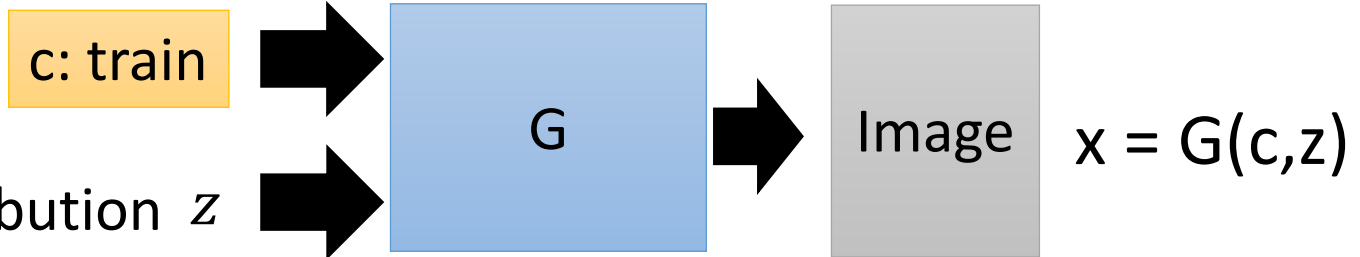
It is a distribution.


Approximate the distribution below


Text: "train"

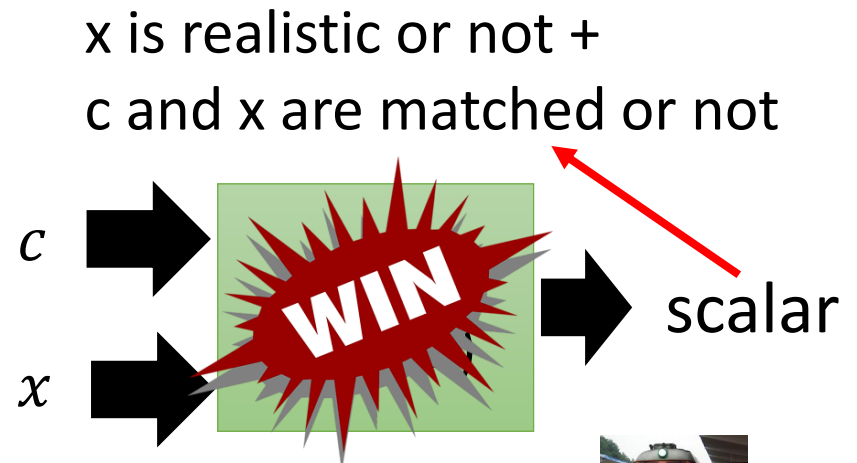


Conditional GAN



Positive example: 

Negative example: 



Positive example: (train, )

Negative example:



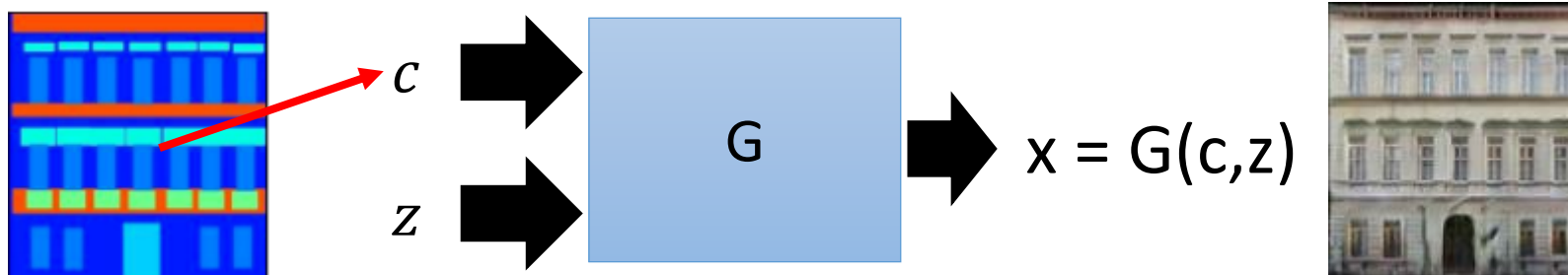
(train, ) (cat, )

Image-to-image



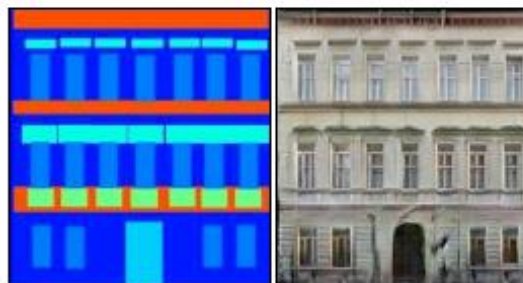
Labels to Street Scene



input

output

Labels to Facade



input

output

BW to Color



input

output

Aerial to Map



input

output

Day to Night



input

output

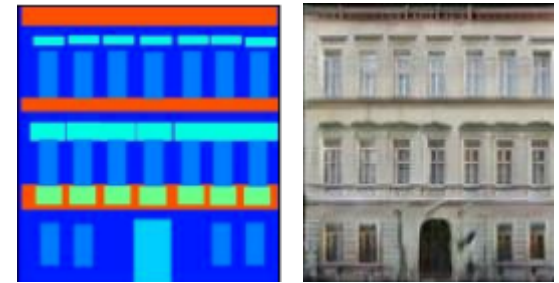
Edges to Photo



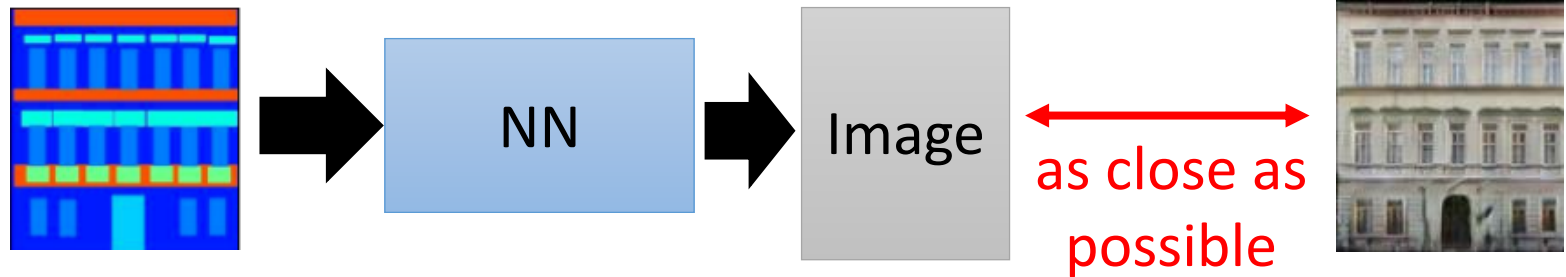
input

output

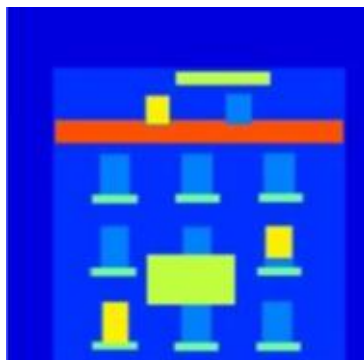
Image-to-image



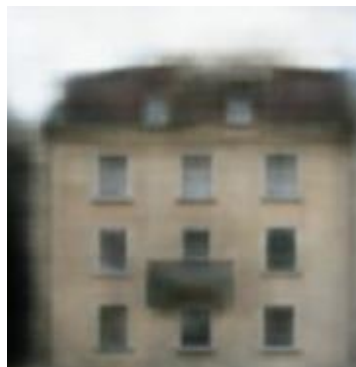
- Traditional supervised approach



Testing:



input

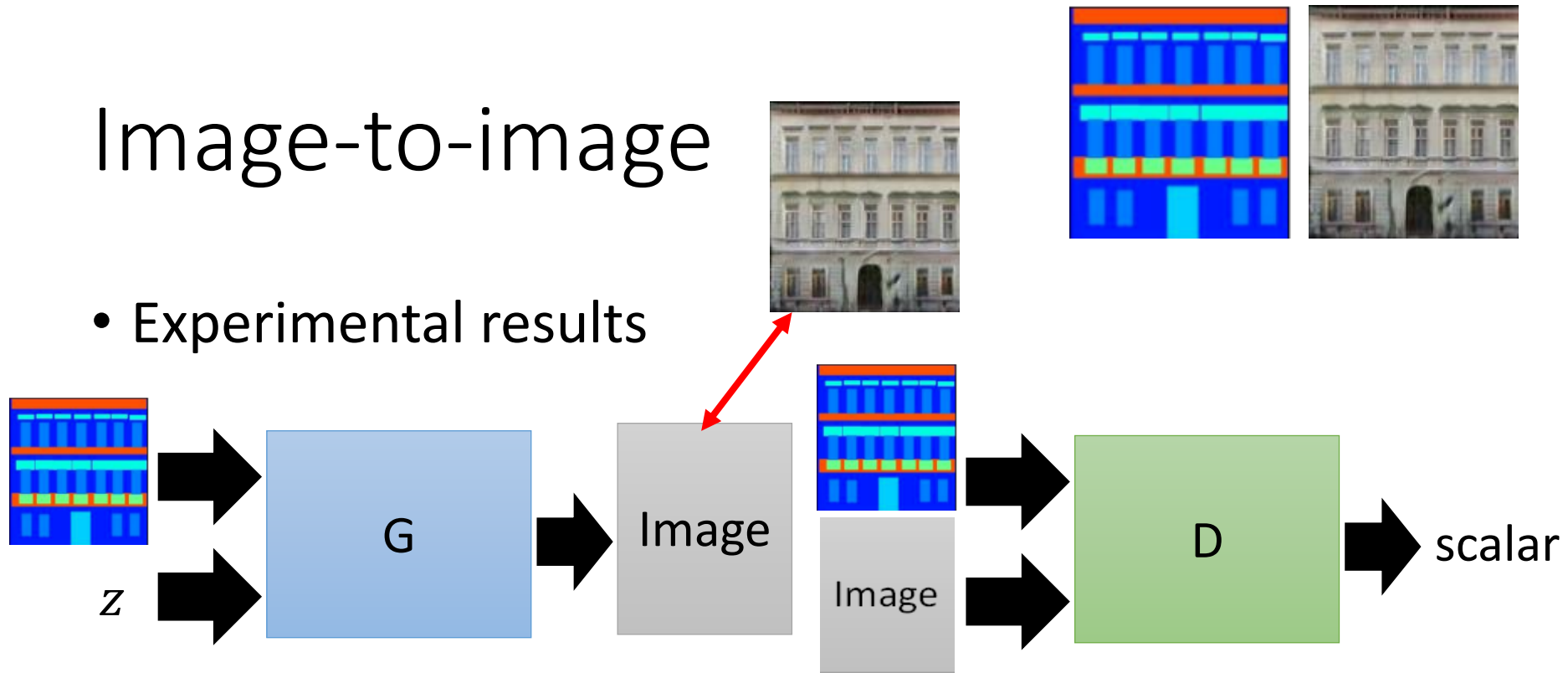


close

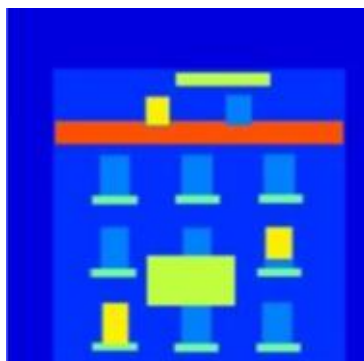
It is blurry because it is the average of several images.

Image-to-image

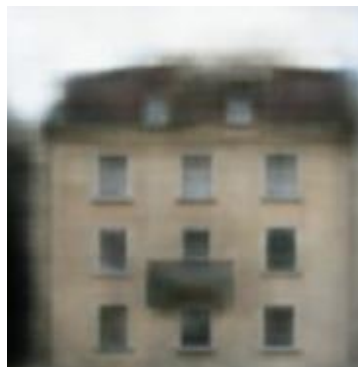
- Experimental results



Testing:



input



close



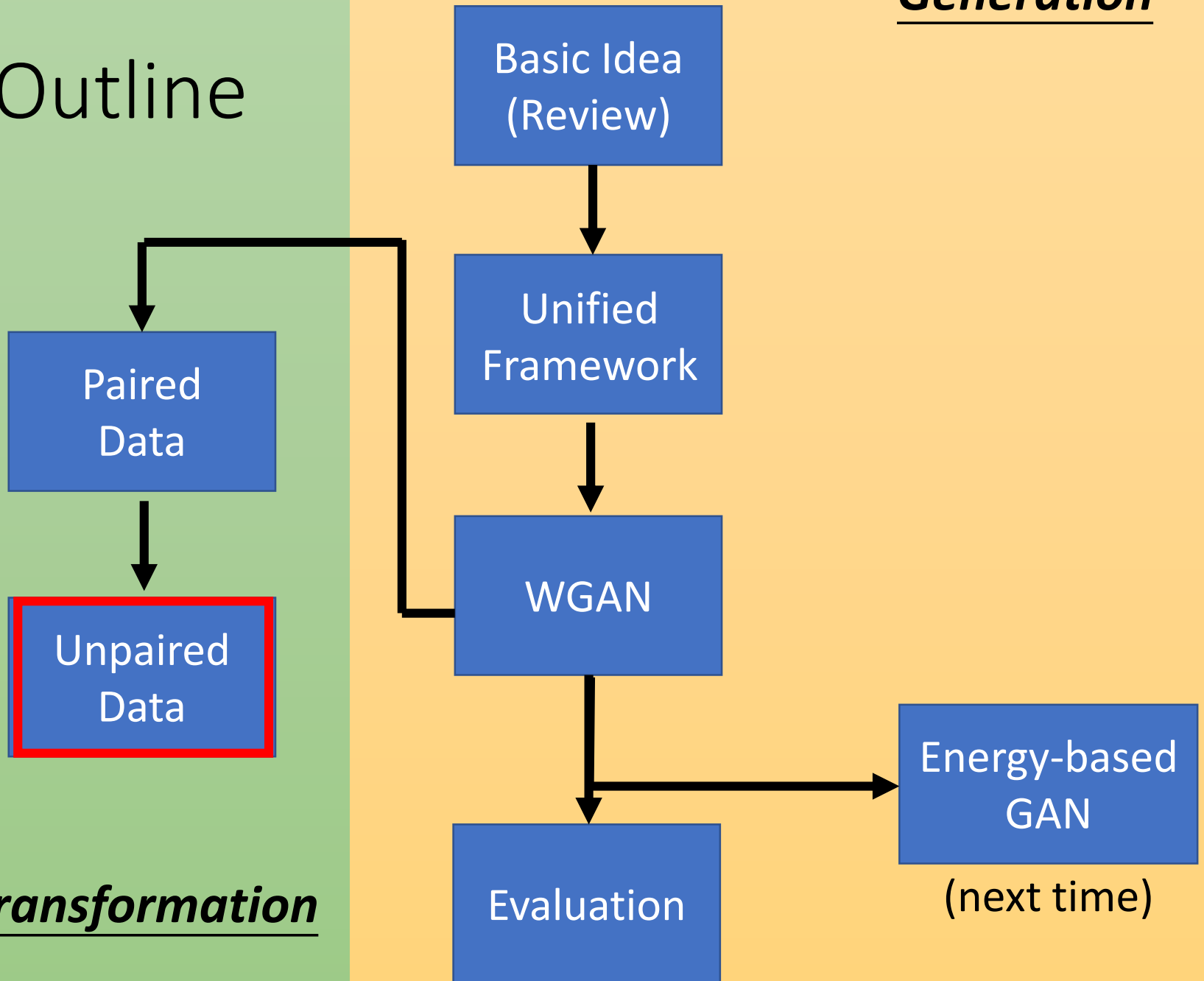
GAN



GAN + close

Generation

Outline

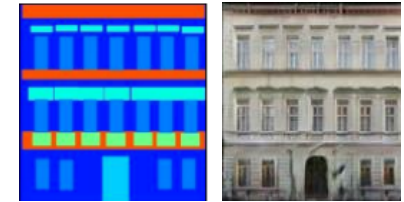


Transformation

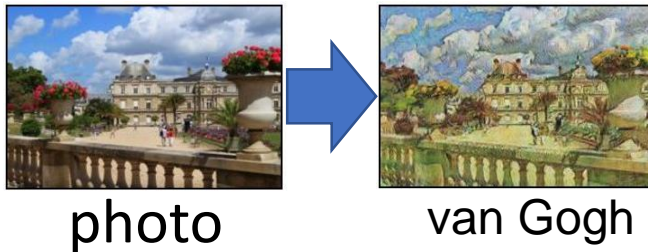
Unpaired Transformation

- Cycle GAN, Disco GAN

paired data



Transform an object from one domain to another *without paired data*



Monet ↔ Photos



Monet → photo



photo → Monet

Zebras ↔ Horses



horse → zebra

Summer ↔ Winter



summer → winter



winter → summer

Cycle GAN

<https://arxiv.org/abs/1703.10593>

<https://junyanz.github.io/CycleGAN/>

Domain X



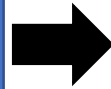
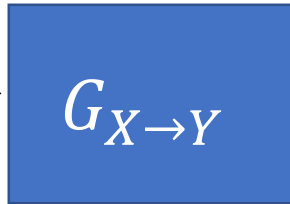
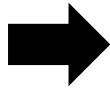
Domain Y



Domain X



ignore input



Become similar
to domain Y

Not what we want



→ scalar



Input image
belongs to
domain Y or not



Domain Y

Cycle GAN

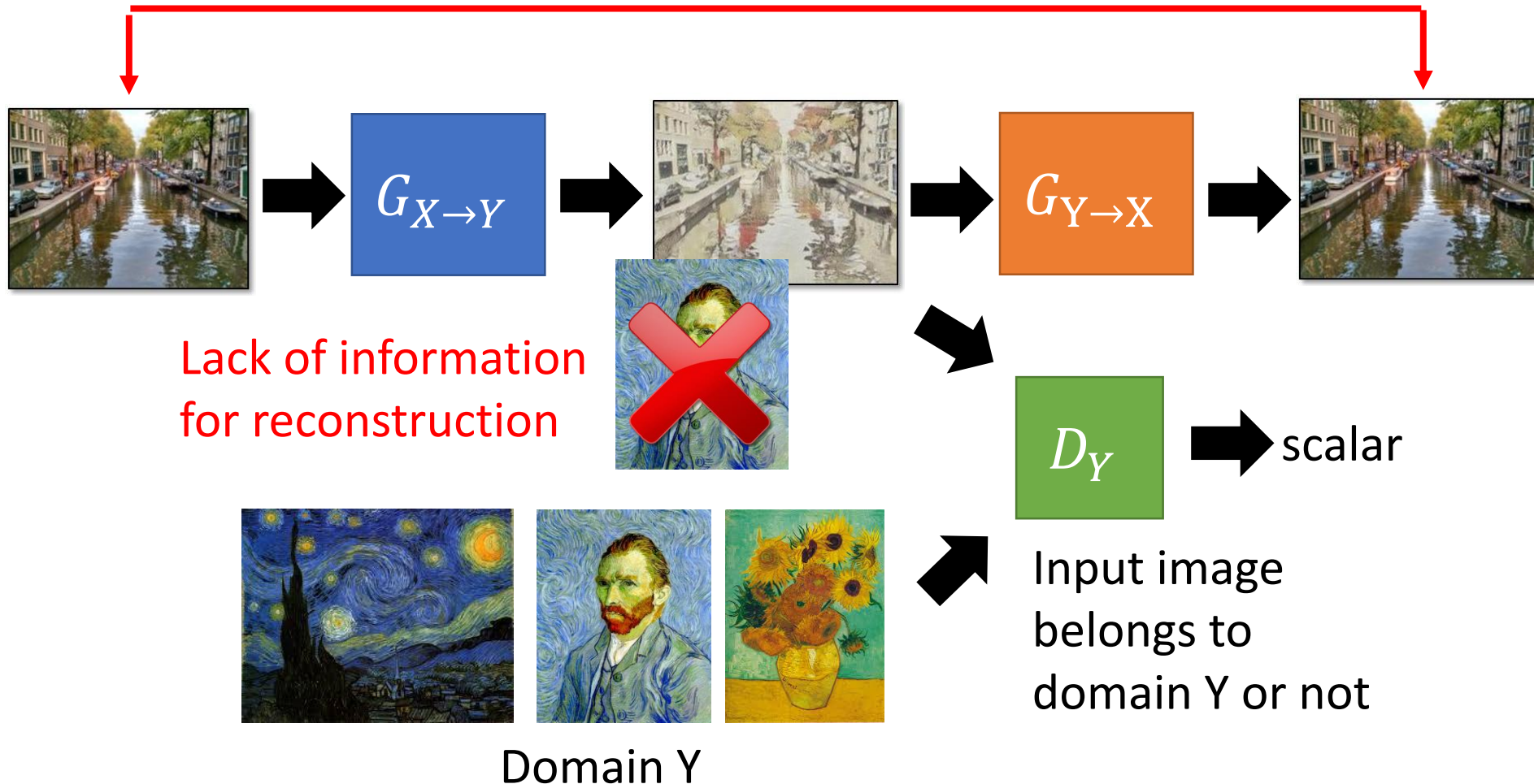
Domain X



Domain Y



as close as possible



Cycle GAN

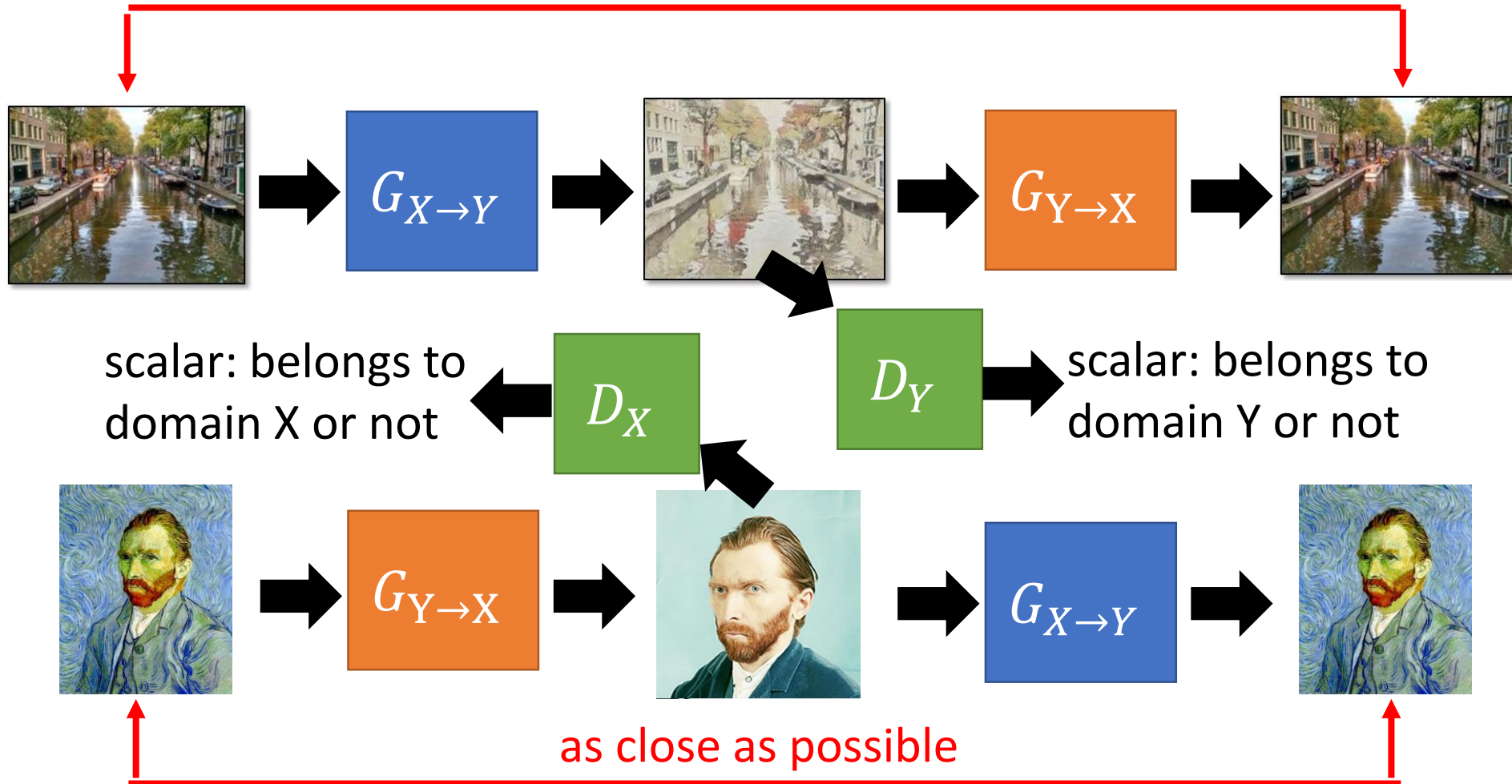
Domain X



Domain Y



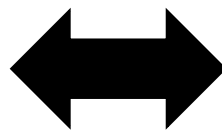
as close as possible



Unpaired Transformation

– 真人動畫化

- <http://qiita.com/Hiking/items/8d36d9029ad1203aac55>



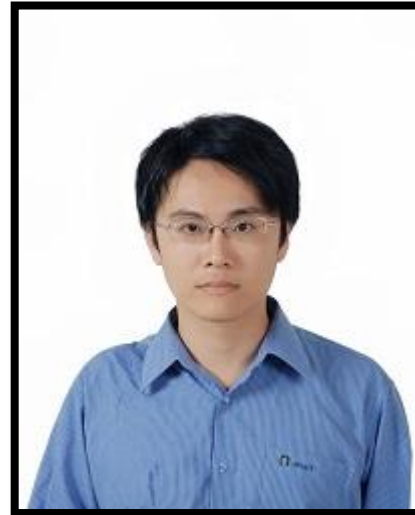
<http://www.iis.ee.ic.ac.uk/cxiong/database.html>

把真人頭像變成動漫人物

Unpaired Transformation

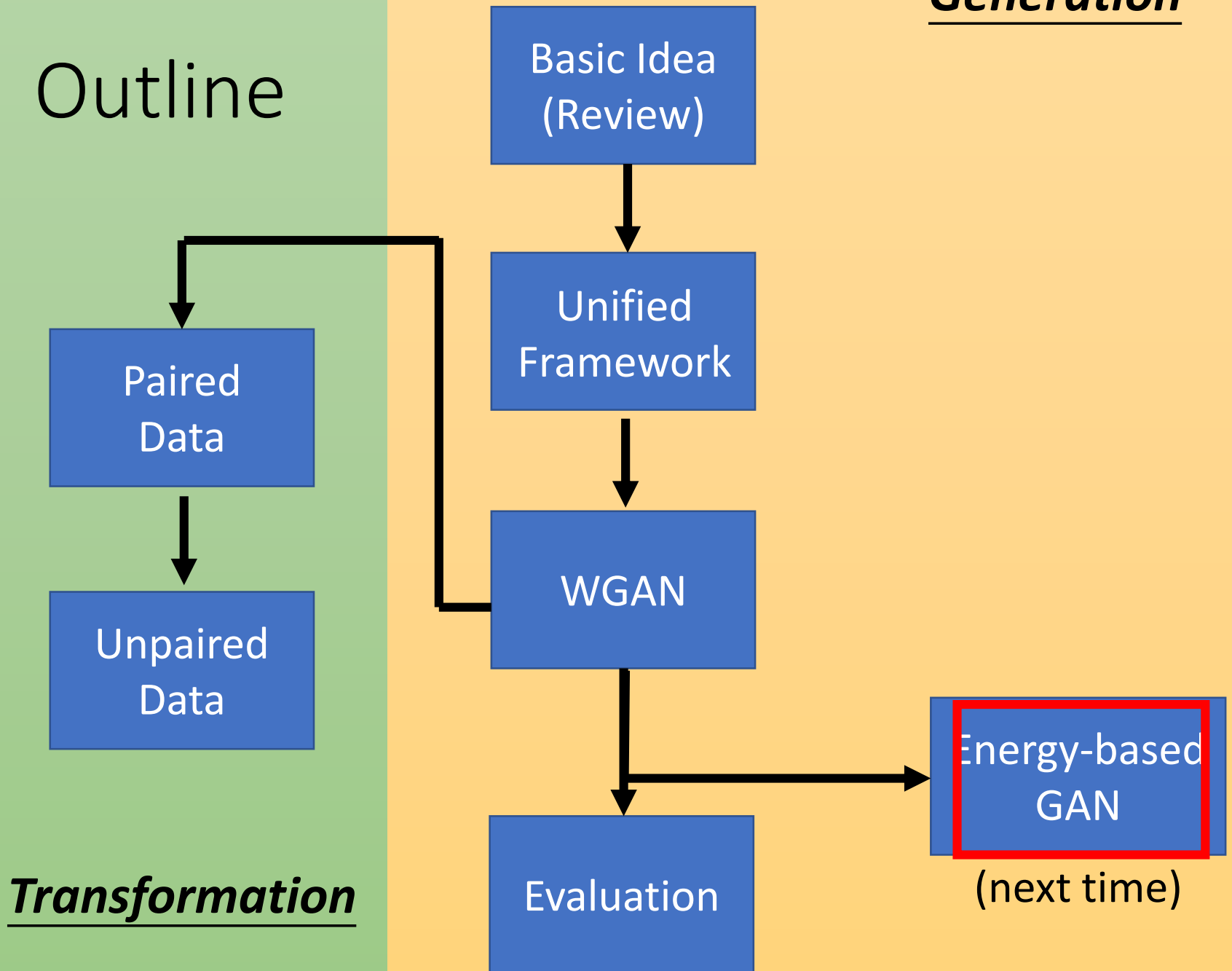
– 真人動畫化

- Using the code:
https://github.com/Hiking/kawaii_creator
- It is not cycle GAN,
Disco GAN



Generation

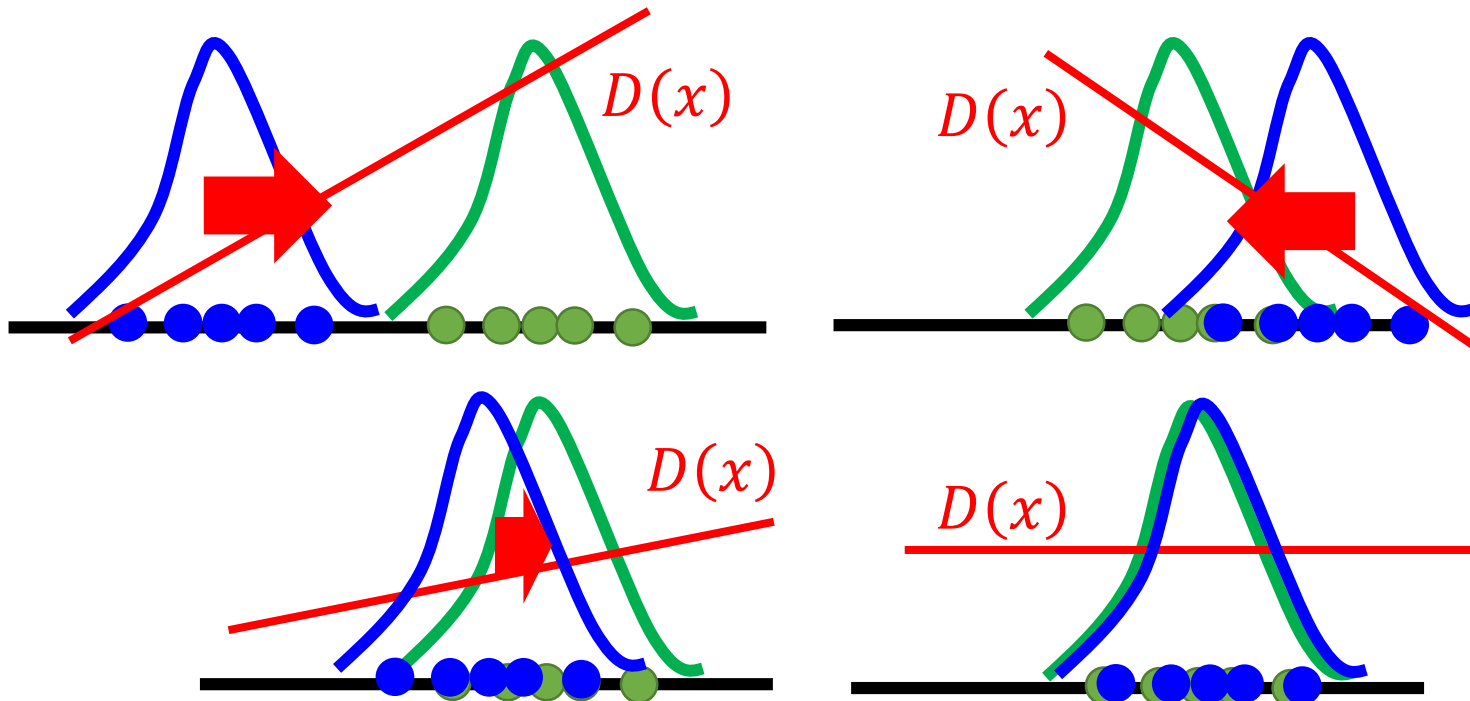
Outline



Review

- Discriminator
- Data (target) distribution
- Generated distribution

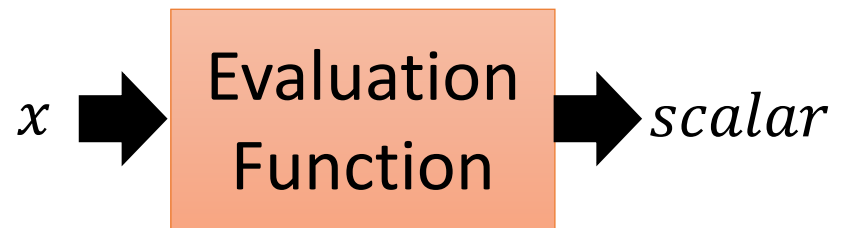
- Role of discriminator: lead the generator



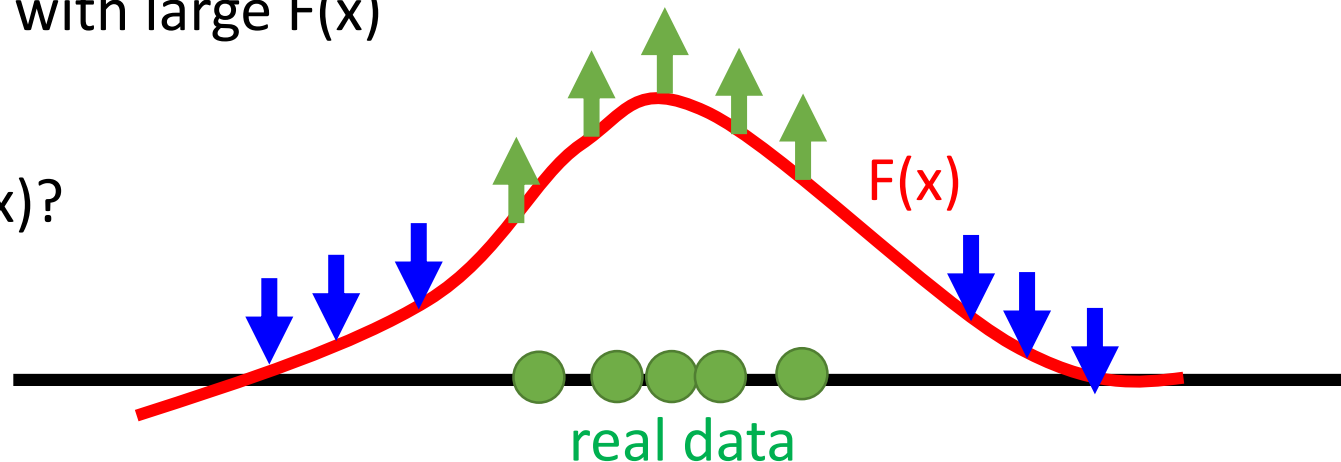
When the data distribution and generated distribution is the same
The discriminator will be useless (flat).

Energy-based Model

- We want to find an evaluation function $F(x)$
 - Input: object x (e.g. images), output: scalar (how good x is)
 - Real x has high $F(x)$
 - $F(x)$ can be a network
- We can find good x by $F(x)$:
 - Generate x with large $F(x)$



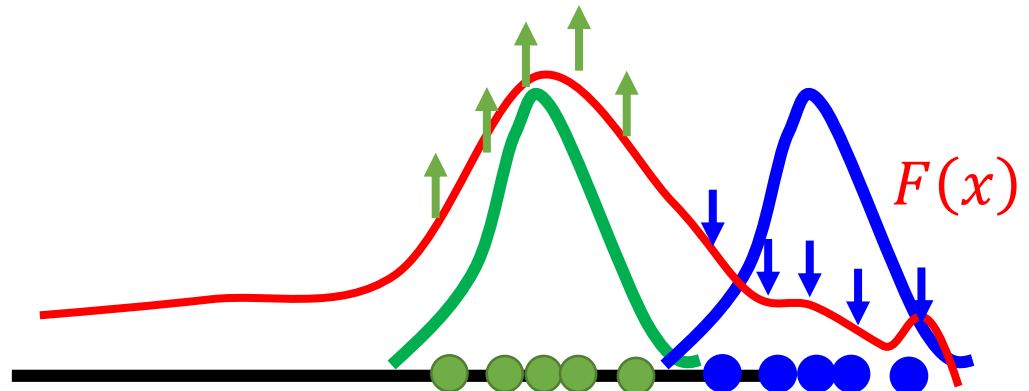
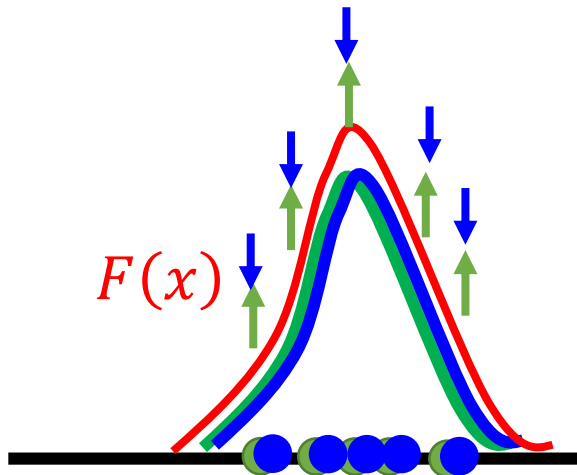
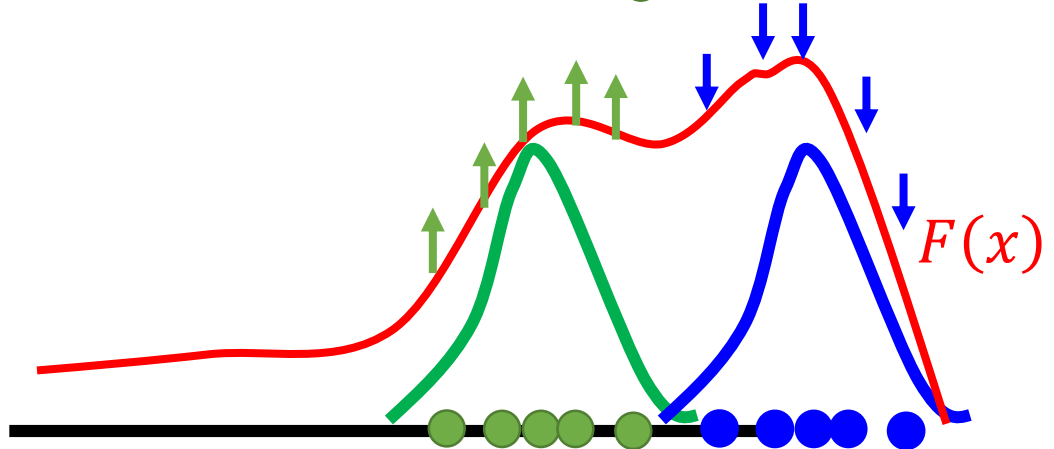
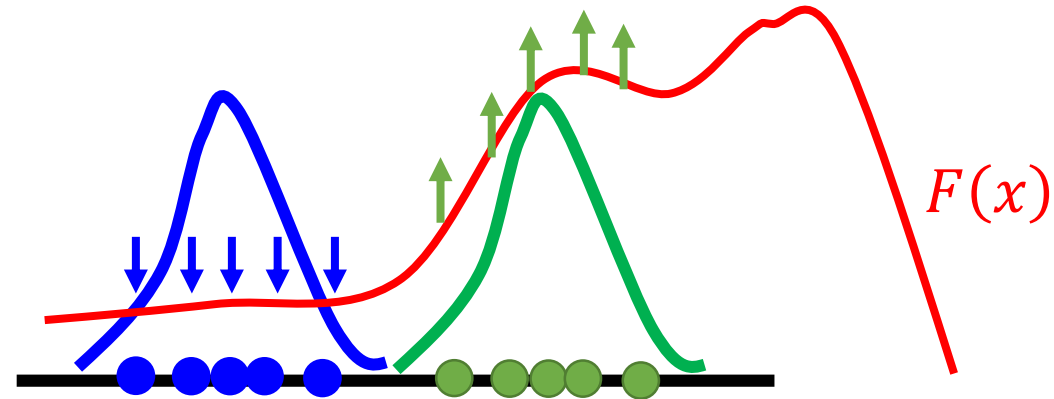
- How to find $F(x)$?



Energy-based GAN

- We want to find an evaluation function $F(x)$
- How to find $F(x)$?

In the end



Energy-based Model

- Preview: Framework of structured learning (Energy-based Model)
 - ML Lecture 21: Structured Learning - Introduction
 - <https://www.youtube.com/watch?v=5OYu0vxXEv8>
 - ML Lecture 22: Structured Learning - Linear Model
 - <https://www.youtube.com/watch?v=HfPw40JPays>
 - ML Lecture 23: Structured Learning - Structured SVM
 - <https://www.youtube.com/watch?v=YjvGVVrCrhQ>
 - ML Lecture 24: Structured Learning - Sequence Labeling
 - <https://www.youtube.com/watch?v=o9FPSqobMys>