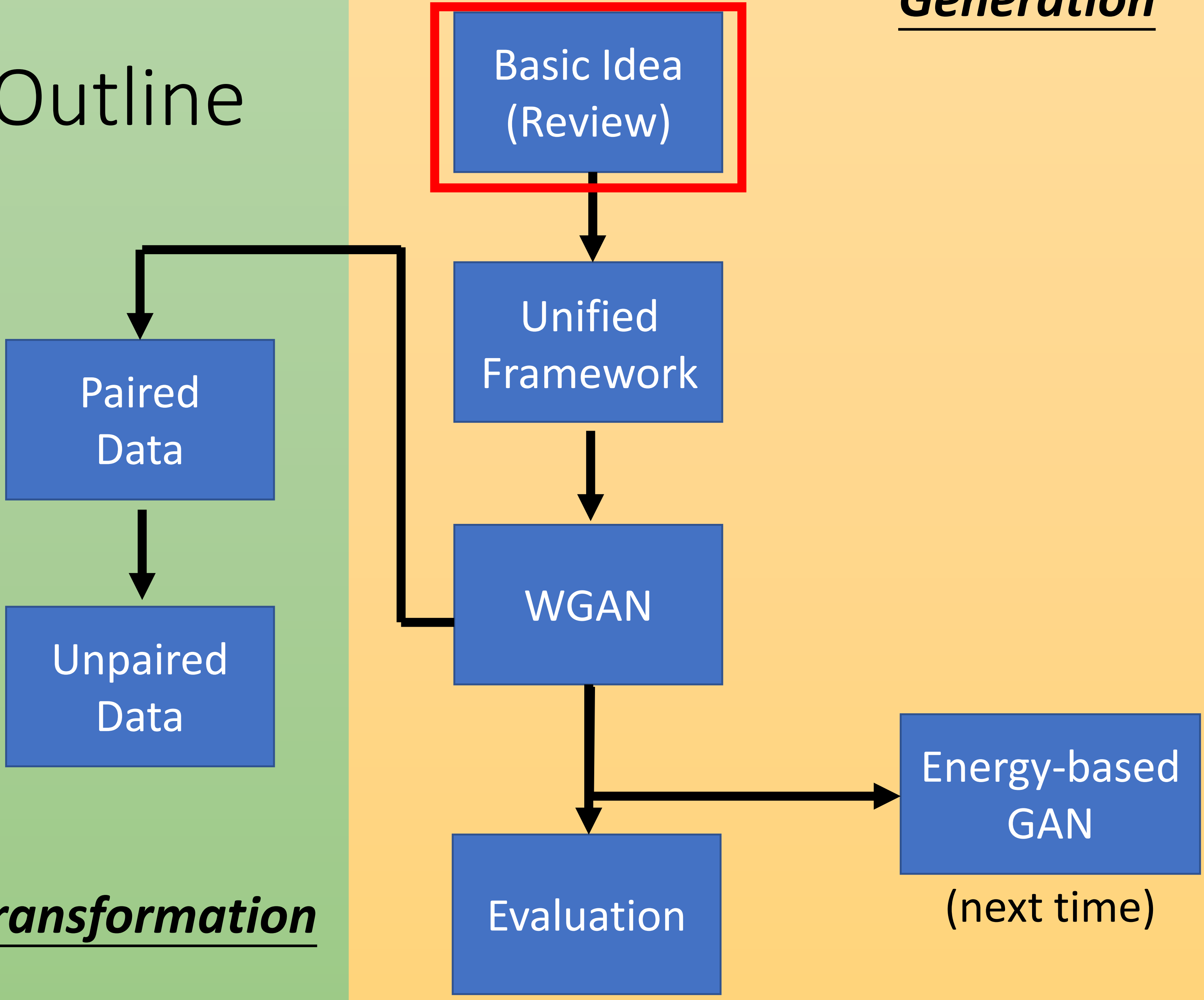
Improving Generative Adversarial Network (GAN)

Hung-yi Lee

Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

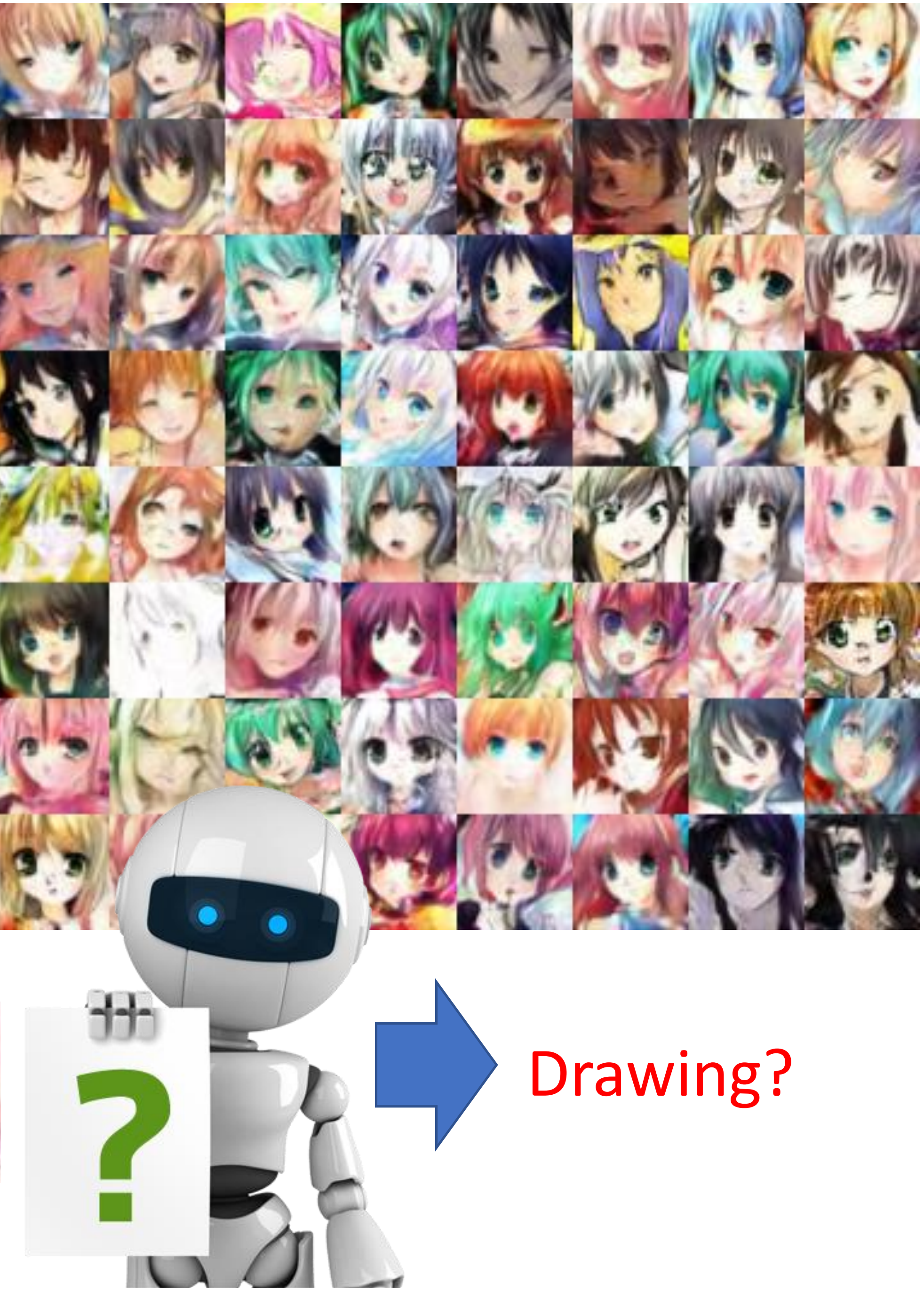
Paired Data

Unpaired Data

Transformation

Generation

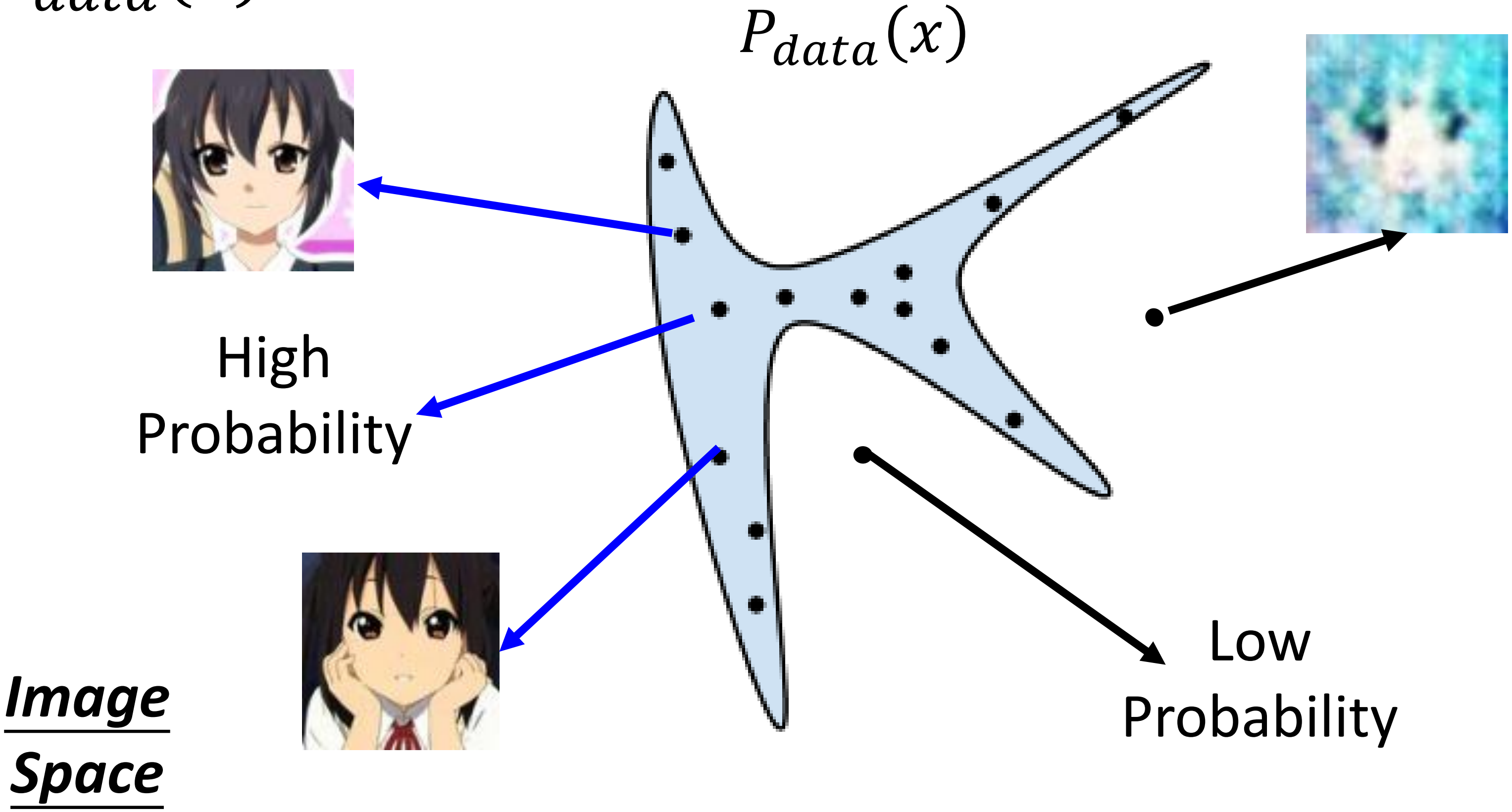
Using Generative Adversarial Network (GAN)



Drawing?

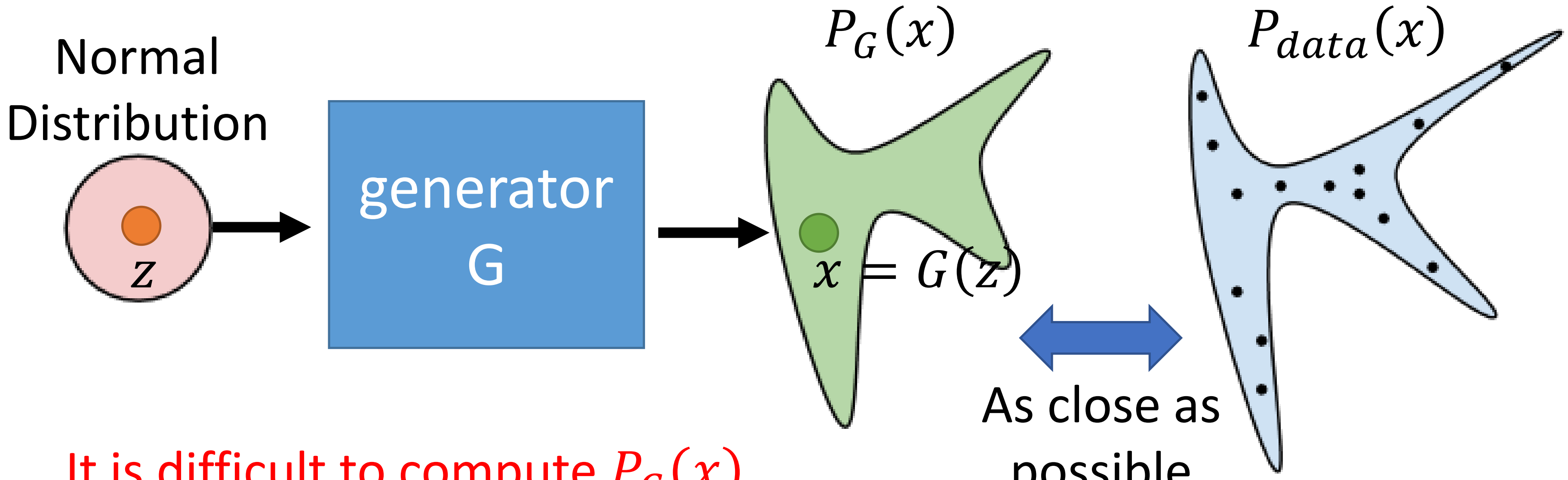
Basic Idea of GAN

- 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.



Normal Distribution

z

generator G

$P_G(x)$

$x = G(z)$

$P_{data}(x)$

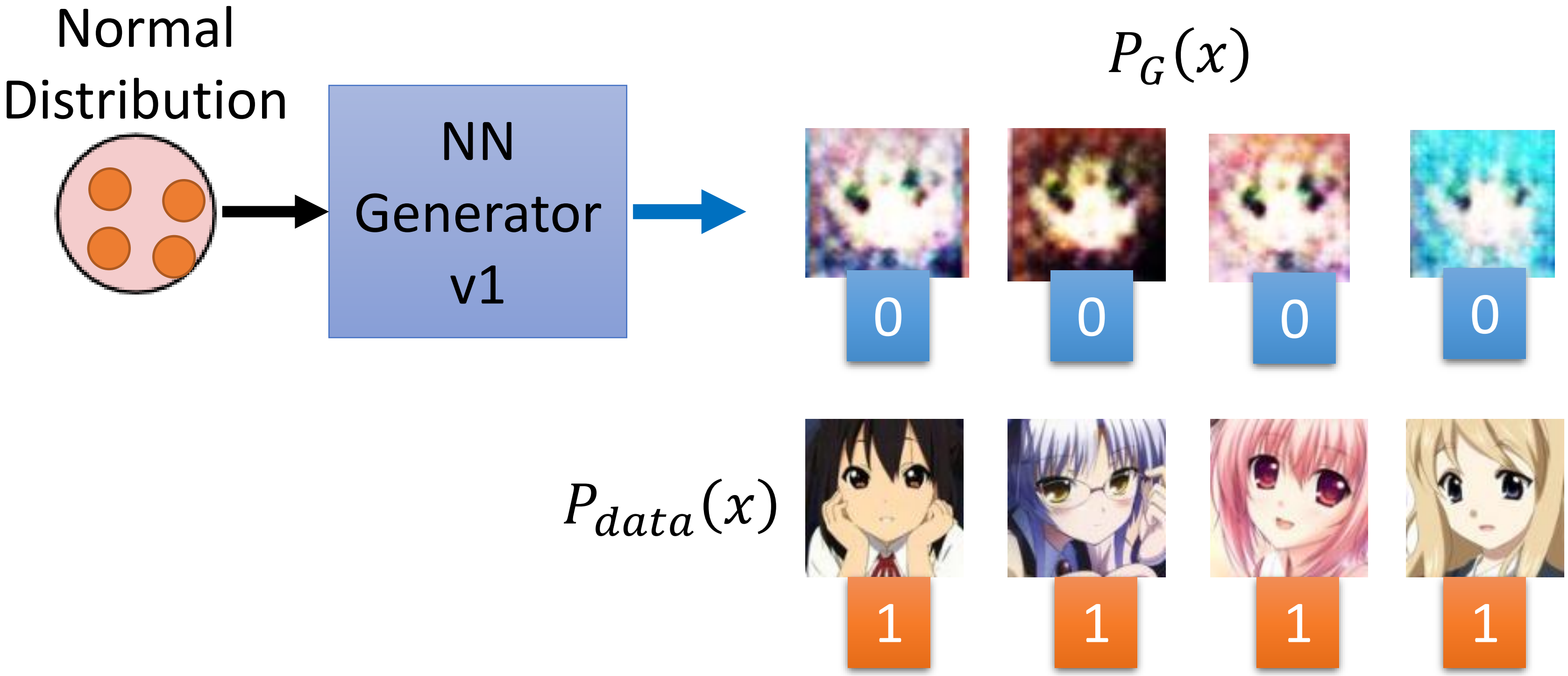
As close as possible

It is difficult to compute $P_G(x)$

We can only sample from the distribution.

https://blog.openai.com/generative-models/

Basic Idea of GAN



Normal Distribution

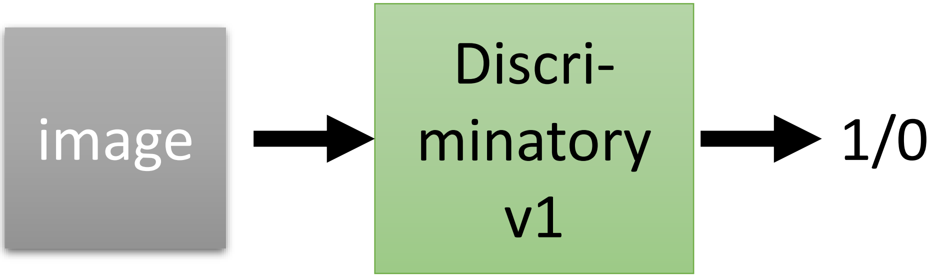
NN Generator v1

$P_G(x)$

0 0 0 0

$P_{data}(x)$

1 1 1 1



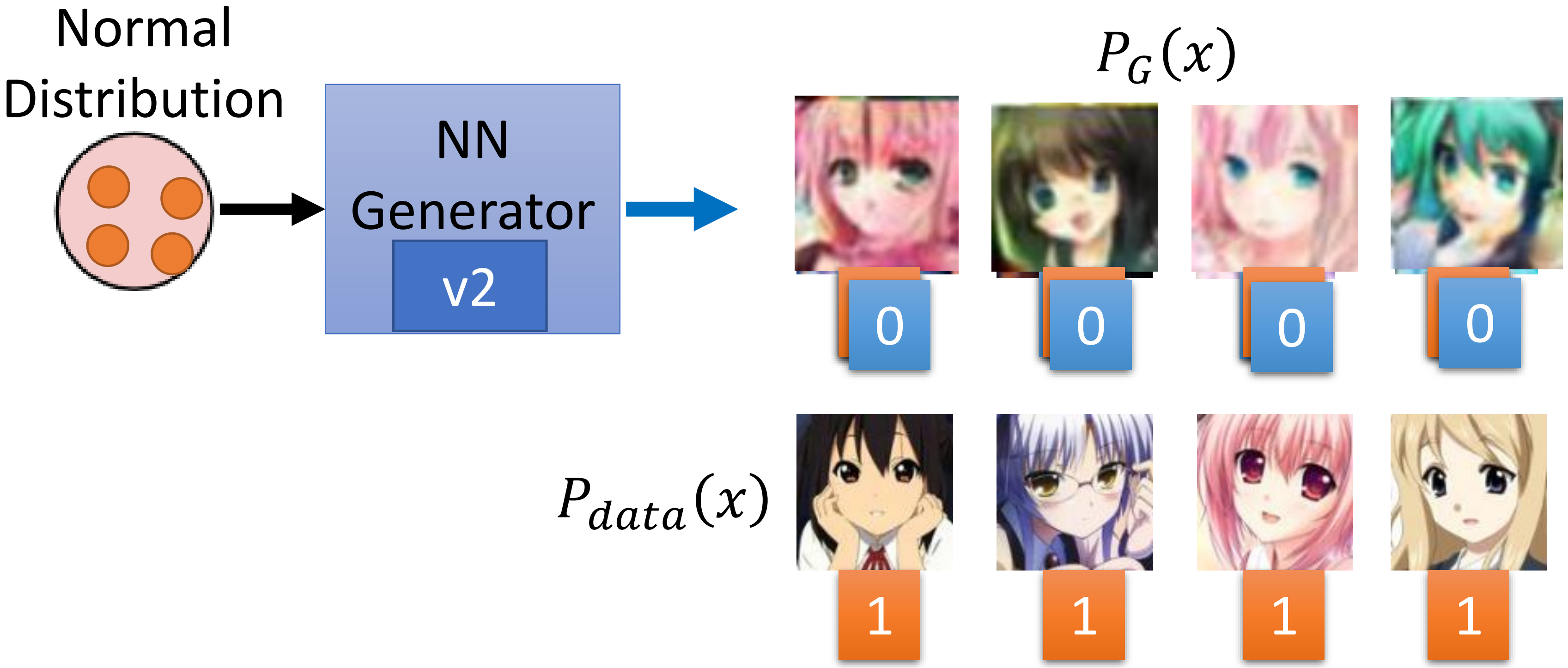
image

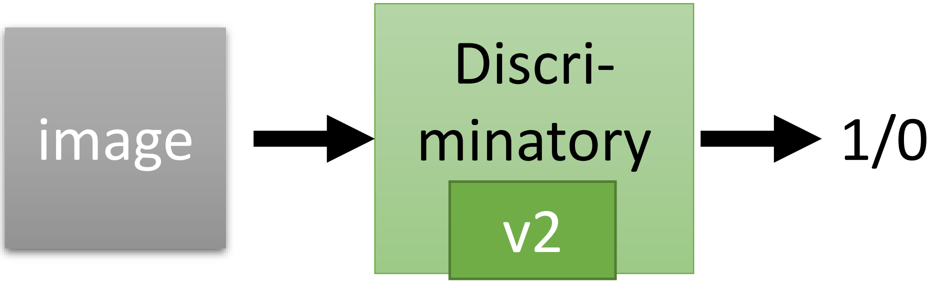
Discri-minatory v1

1/0

It can be proofed that the ***loss of the discriminator*** related to ***JS divergence***.

Basic Idea of GAN



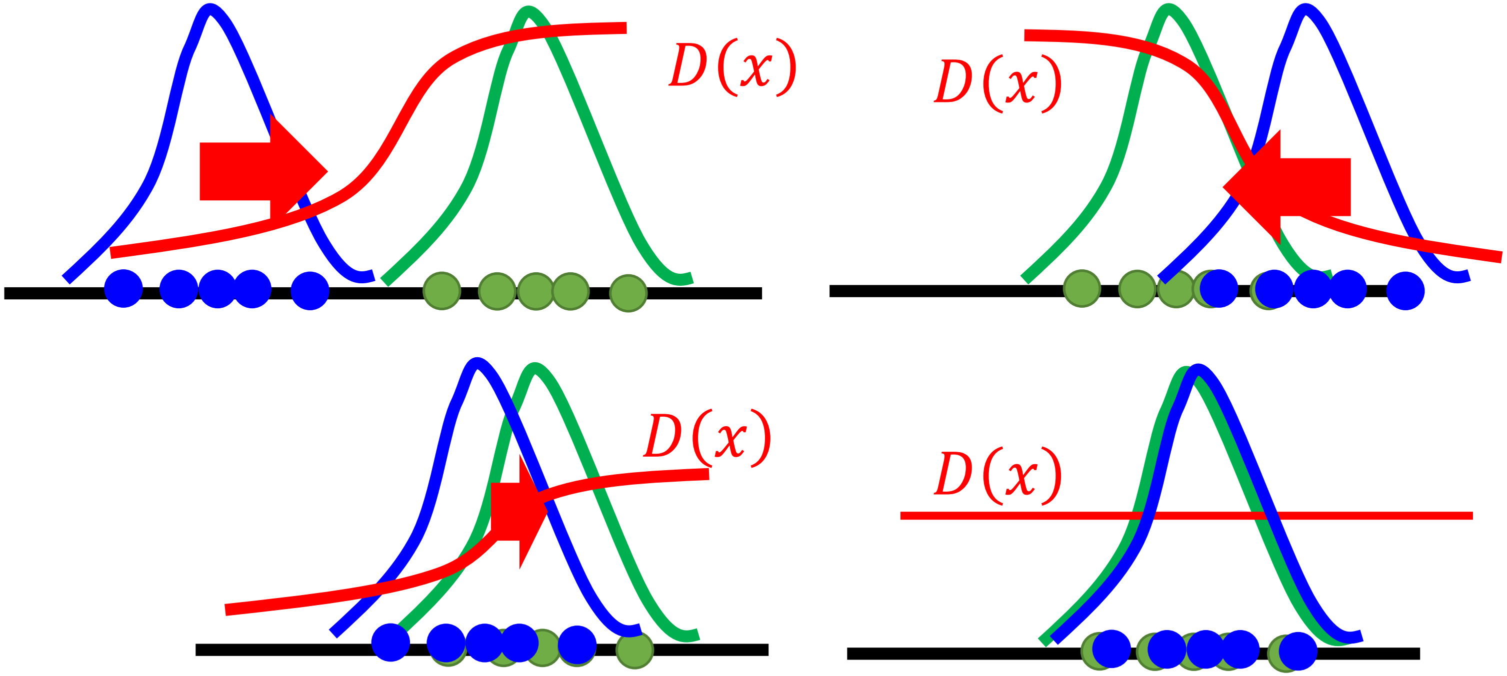


It can be proofed that the ***loss the discriminator*** related to ***JS divergence***.

Intuition

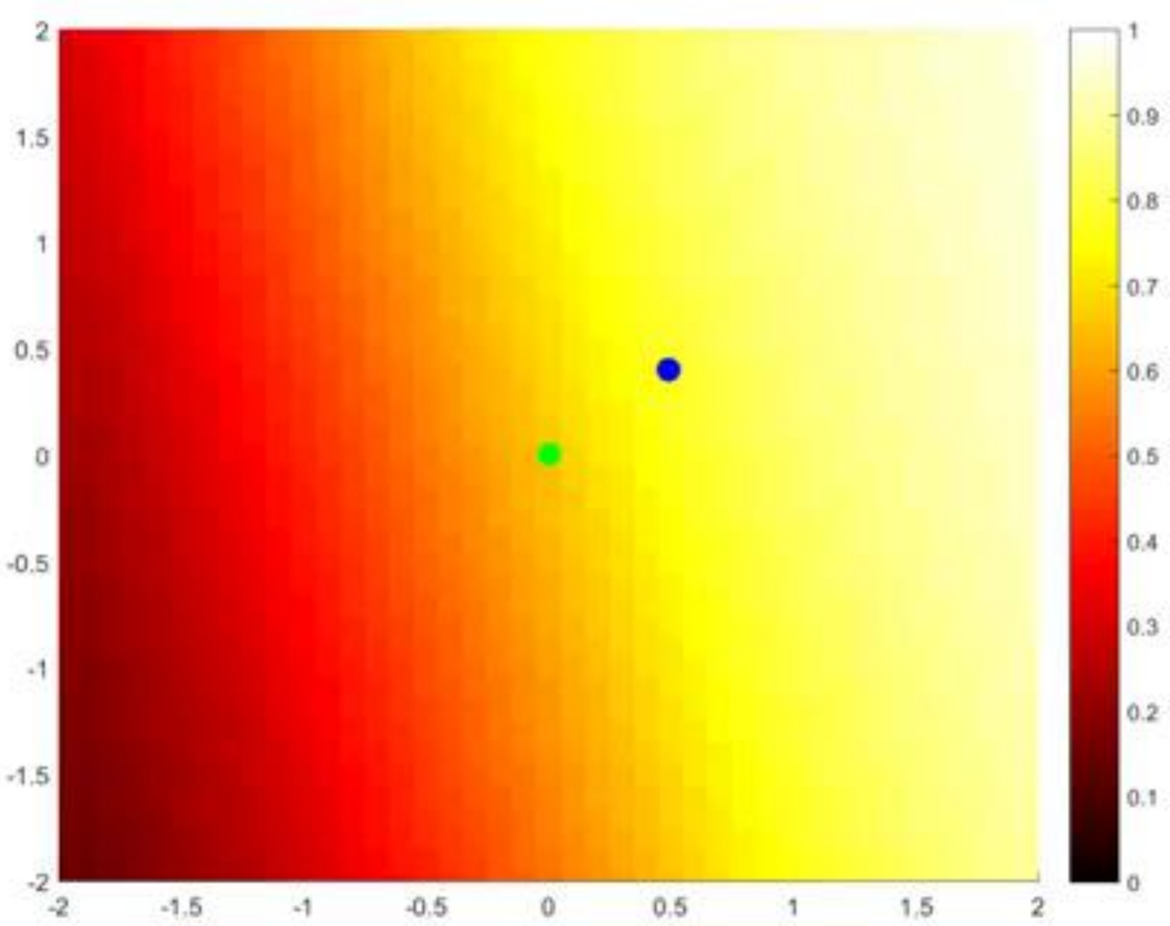
🟥 Discriminator
🟩 Data (target) distribution
🟦 Generated distribution

- 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

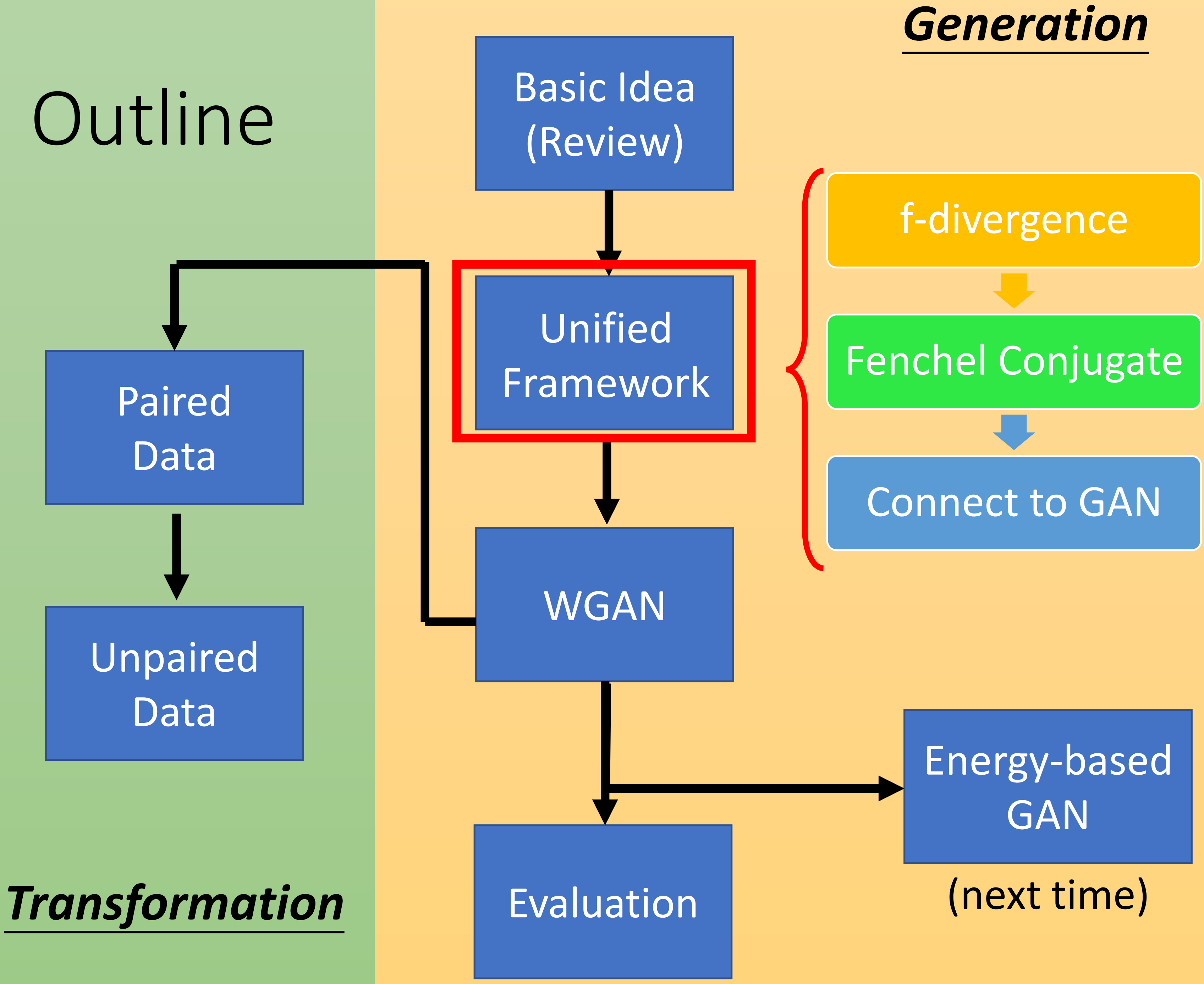
  - Sample m examples $\{x^1, x^2, \dots, x^m\}$ from data distribution $P_{data}(x)$
  - Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from the prior $P_{prior}(z)$
  - Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G\left(z^i\right)$
  - Update discriminator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

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
    - $\tilde{V} = \cancel{\frac{1}{m}\sum_{i=1}^{m} logD\left(x^i\right)} + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G\left(z^i\right)\right)\right)$
    - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}\left(\theta_g\right)$

Outline



Generation

Basic Idea (Review)

Unified Framework

- f-divergence
- Fenchel Conjugate
- Connect to GAN

WGAN

Evaluation

Energy-based GAN

(next time)

Transformation

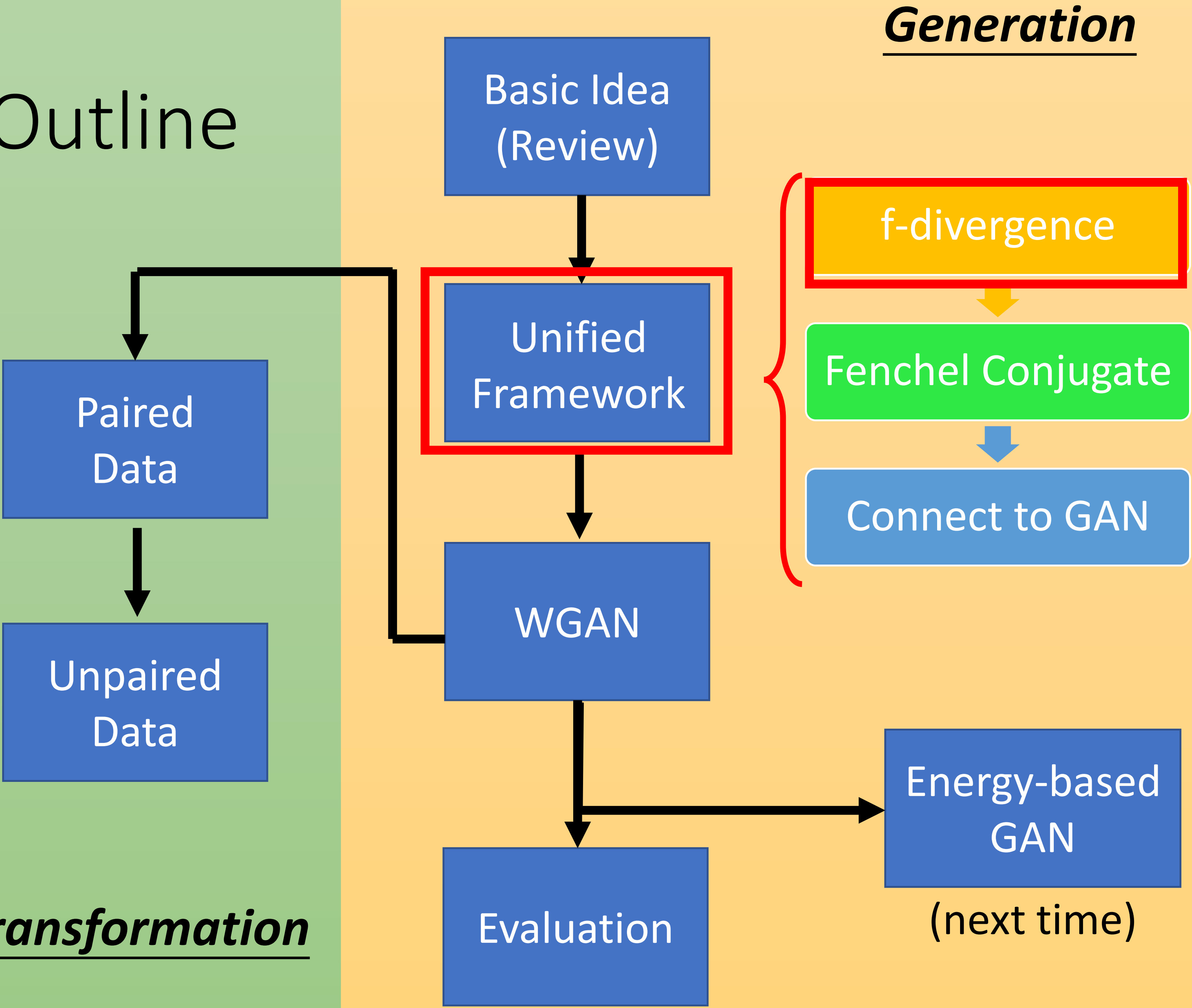
Paired Data

Unpaired Data

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

Basic Idea (Review)

Unified Framework

- f-divergence
- Fenchel Conjugate
- Connect to GAN

WGAN

Evaluation

Energy-based GAN

(next time)

Transformation

Paired Data

Unpaired Data

***f*-divergence**

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

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

f is convex

f(1) = 0

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

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

smallest

$$D_f(P||Q) = \int_x q(x) \underbrace{f\left(\overbrace{\boxed{\frac{p(x)}{q(x)}}}^{= 1}\right)}_{= 0} dx = 0$$

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

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

$$f(x) = xlogx$$

KL

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

$$f(x) = -logx$$

Reverse KL

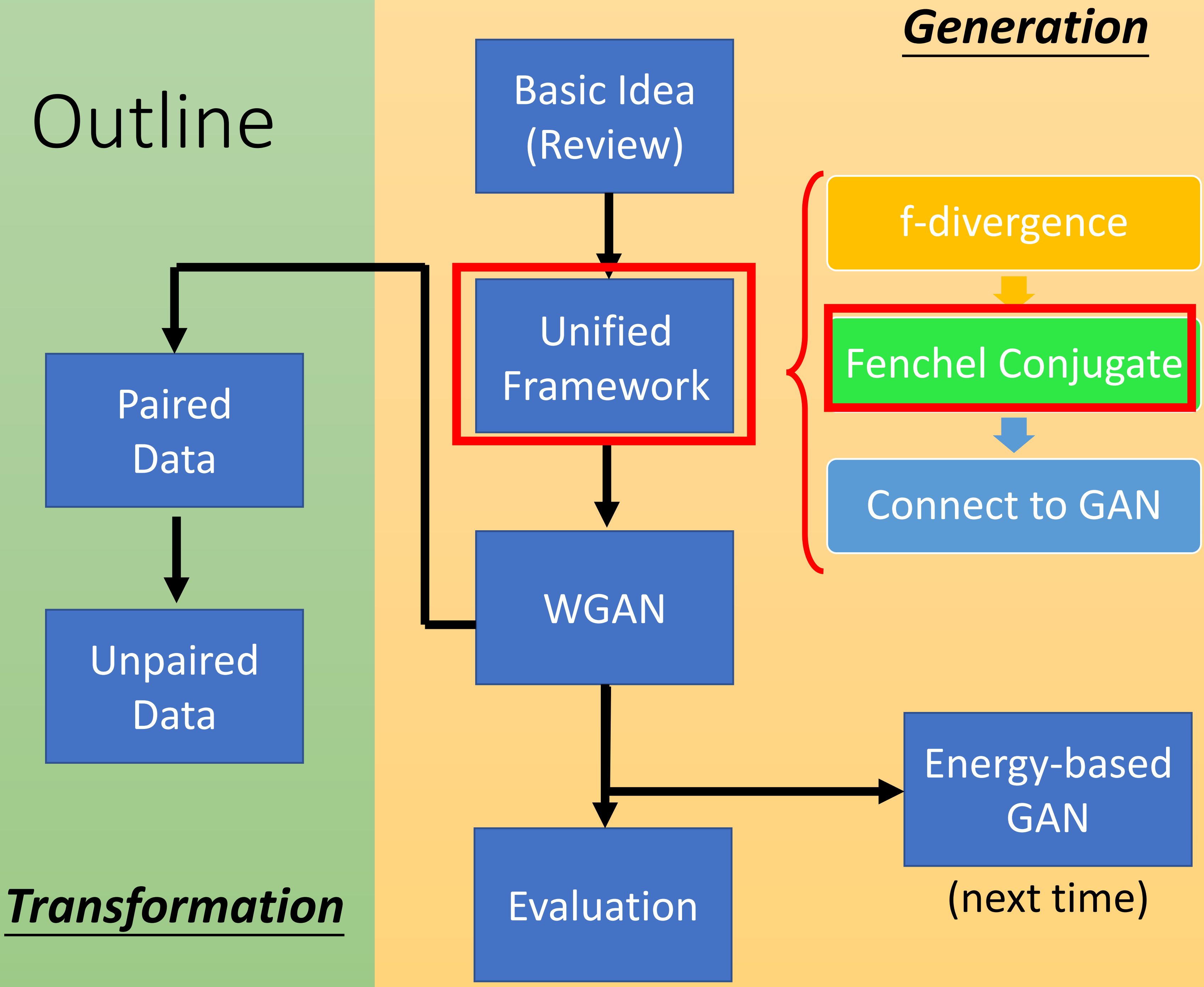
$$D_f(P||Q) = \int_x q(x)\left(-log\left(\frac{p(x)}{q(x)}\right)\right) dx = \int_x q(x) log\left(\frac{q(x)}{p(x)}\right) dx$$

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

Chi Square

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

Outline



Generation

Basic Idea (Review)

Unified Framework

f-divergence

Fenchel Conjugate

Connect to GAN

WGAN

Evaluation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

Transformation

Fenchel Conjugate

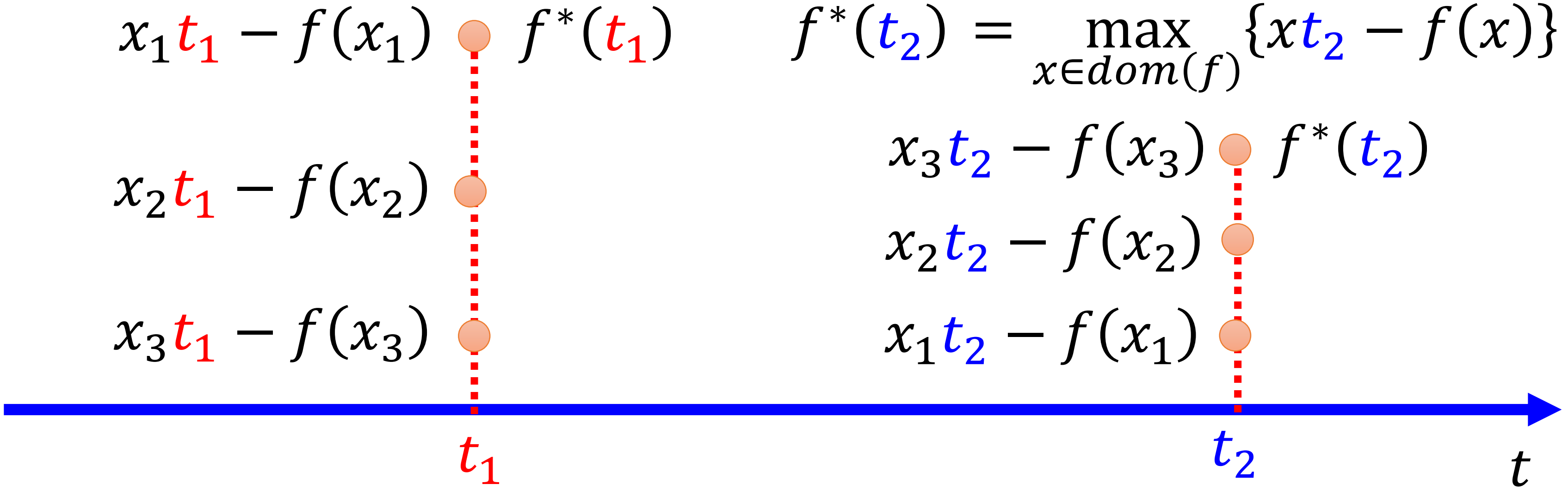
$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

f is convex, f(1) = 0

- Every convex function f has a conjugate function f*

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

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



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

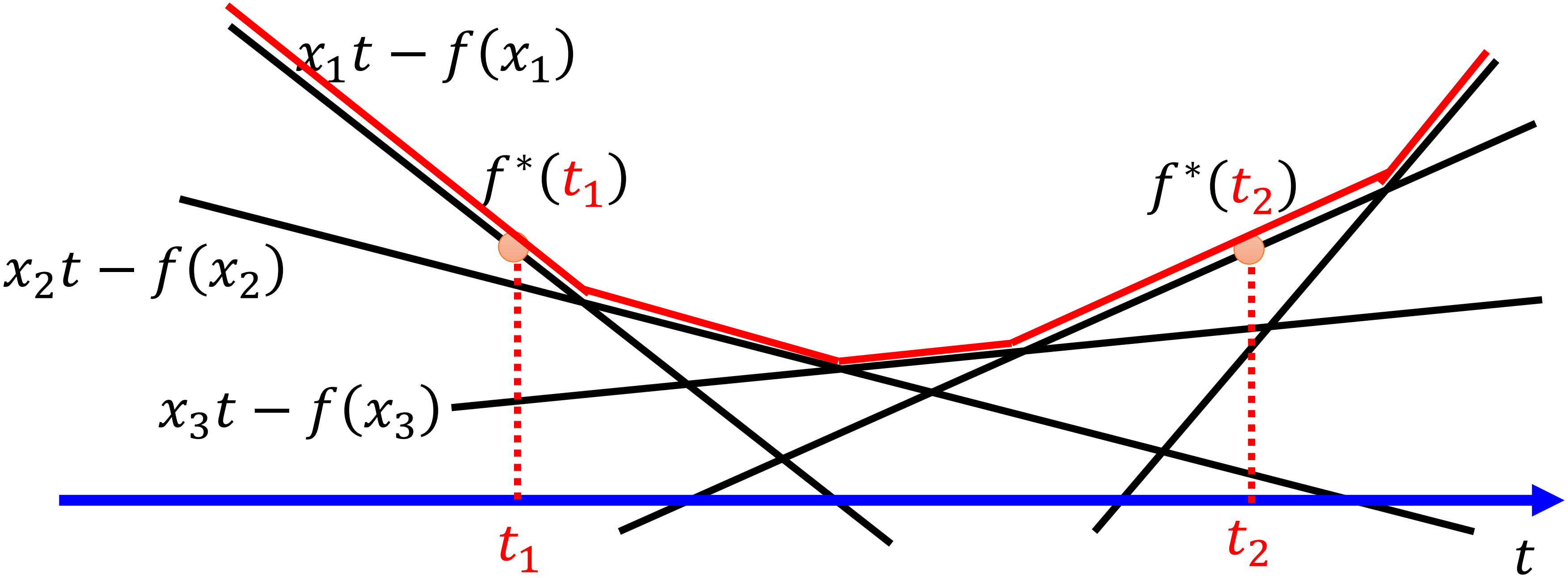
Fenchel Conjugate

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

f is convex, f(1) = 0

- Every convex function f has a conjugate function f*

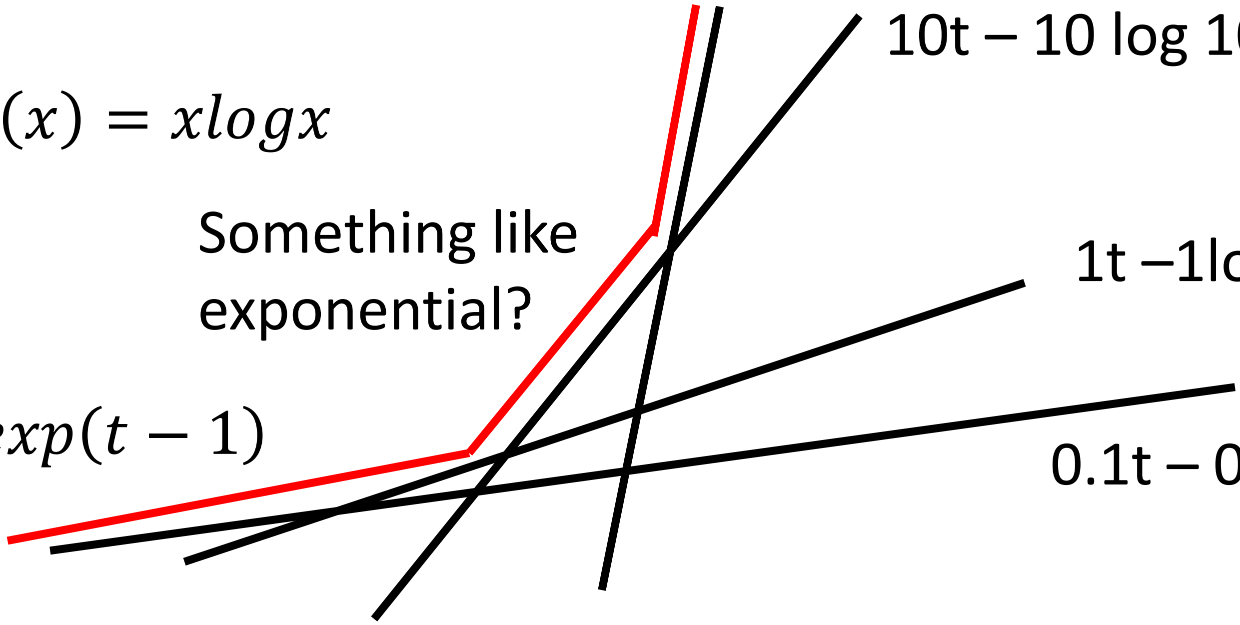
$$f^*(t) = \boxed{\max_{x \in dom(f)} \boxed{\{xt - f(x)\}}}$$



Fenchel Conjugate

- Every convex function f has a conjugate function f*

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



10t – 10 log 10

$f(x) = xlogx$

Something like exponential?

1t –1log1 = 1t – 0

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

0.1t – 0.1log0.1

Fenchel Conjugate

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

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

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

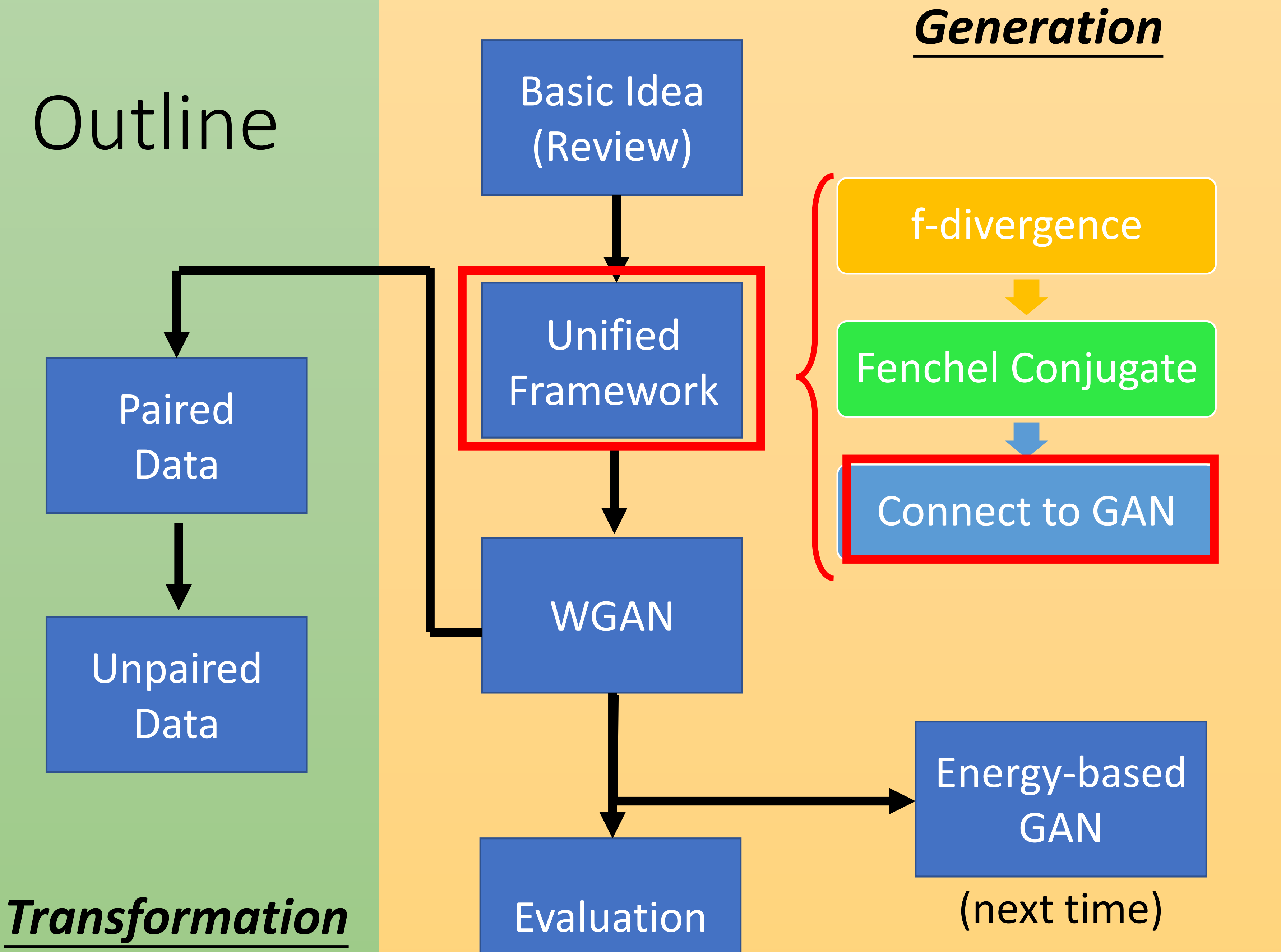
$$f^*(t) = \max_{x \in dom(f)} \{xt - xlogx\}$$

$g(x) = xt - xlogx$ Given t, find x maximizing $g(x)$

$t - logx - 1 = 0 \qquad x = exp(t-1)$

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

Outline



Generation

Basic Idea (Review)

Unified Framework

f-divergence

Fenchel Conjugate

Connect to GAN

WGAN

Evaluation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

Transformation

Connection with GAN

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

$$\boxed{\frac{p(x)}{q(x)}} \qquad \boxed{\frac{p(x)}{q(x)}}$$

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

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

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

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

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

$$= \int_x p(x) D(x) dx - \int_x q(x) f^*\big(D(x)\big) dx$$

Connection with GAN

$$D_f(P||Q) \approx \max_{\mathrm{D}} \int_x p(x)D(x)dx - \int_x q(x)f^*\big(D(x)\big)dx$$

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

Samples from P

Samples from Q

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

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

Original GAN has different V(G,D)

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

$$= arg \min_G \max_D V(G,D)$$

familiar? ☺

Double-loop v.s. Single-step

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

- 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

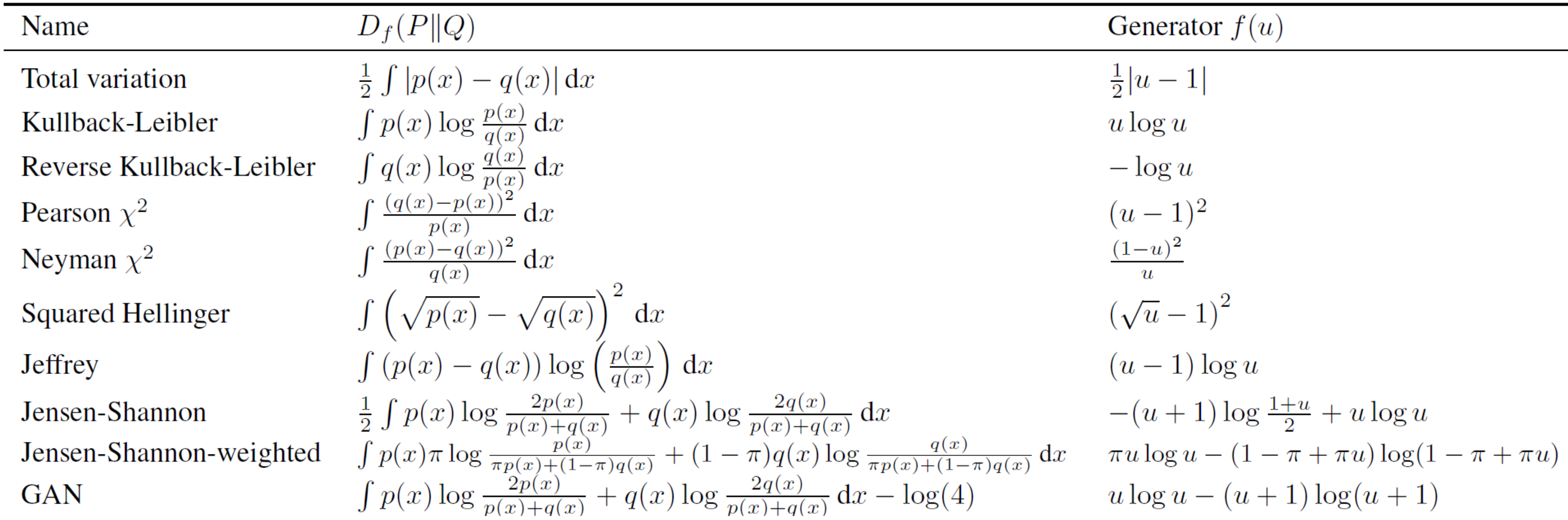
      Inner Loop
    - Update generator once:
      - $\theta_G^{t+1} \leftarrow \theta_G^t - \eta \nabla_{\theta_G} V(\theta_G^t, \theta_D^t)$

Outer Loop

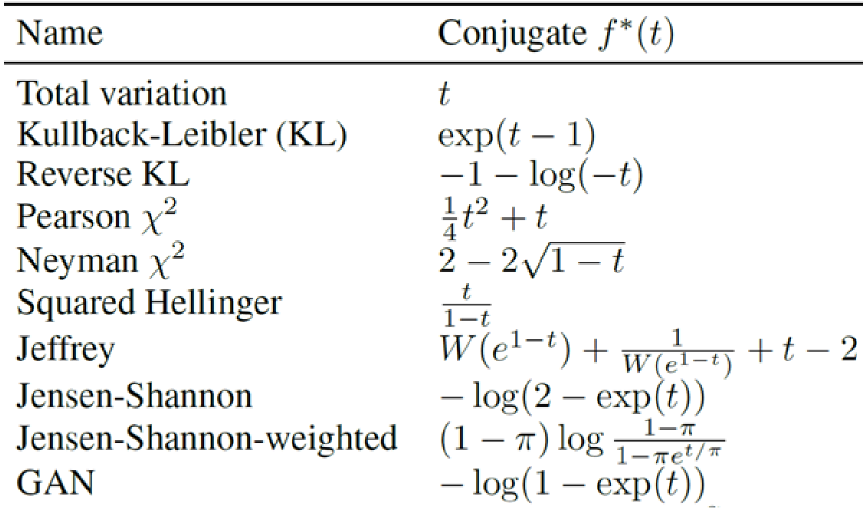
- Paper of f-GAN: Single-step algorithm
  - In each iteration, given θ_G^t and θ_D^t
    - $\theta_D^{t+1} \leftarrow \theta_D^t + \eta \nabla_{\theta_D} \underline{V(\theta_G^t, \theta_D^t)}$
    - $\theta_G^{t+1} \leftarrow \theta_G^t - \eta \nabla_{\theta_G} \underline{V(\theta_G^t, \theta_D^t)}$

One Backpropogation

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



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



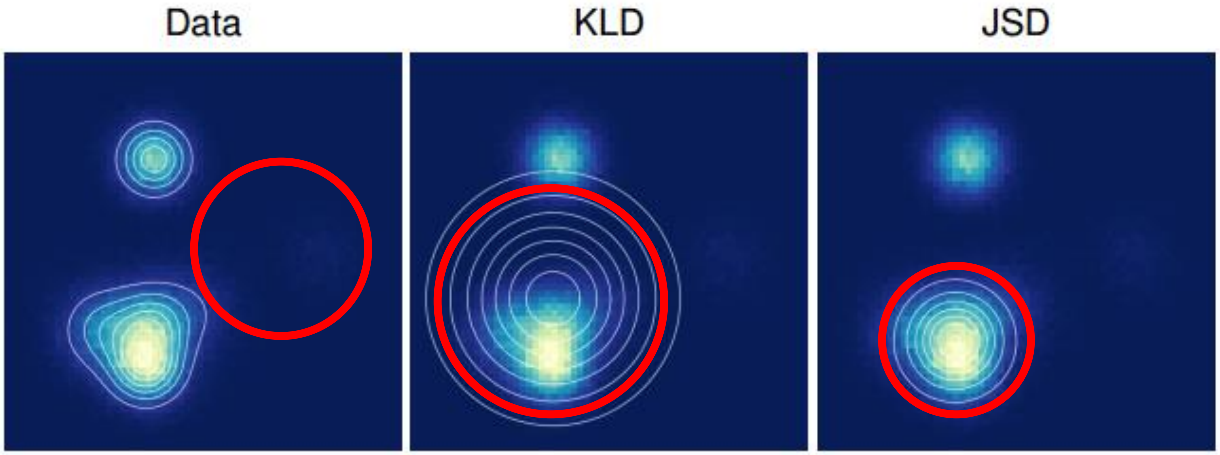
Name	Conjugate $f^*(t)$
Total variation	t
Kullback-Leibler (KL)	$\exp(t-1)$
Reverse KL	$-1-\log(-t)$
Pearson χ^2	$\frac{1}{4}t^2+t$
Neyman χ^2	$2-2\sqrt{1-t}$
Squared Hellinger	$\frac{t}{1-t}$
Jeffrey	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$
Jensen-Shannon	$-\log(2-\exp(t))$
Jensen-Shannon-weighted	$(1-\pi)\log\frac{1-\pi}{1-\pi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

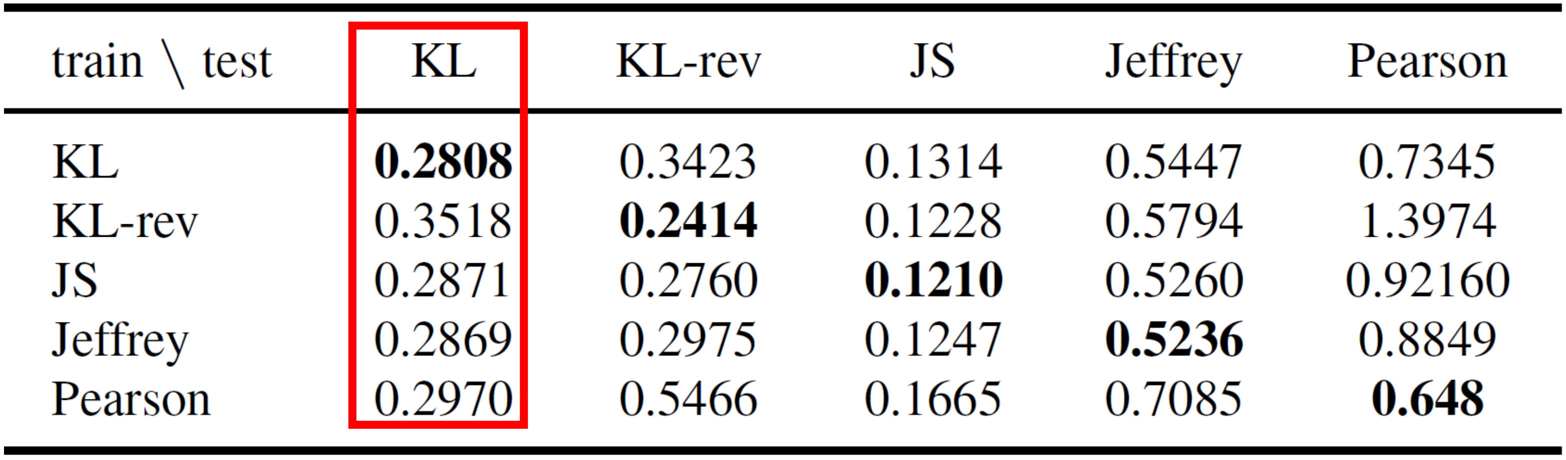
Using the f-divergence you like ☺

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

Experimental Results

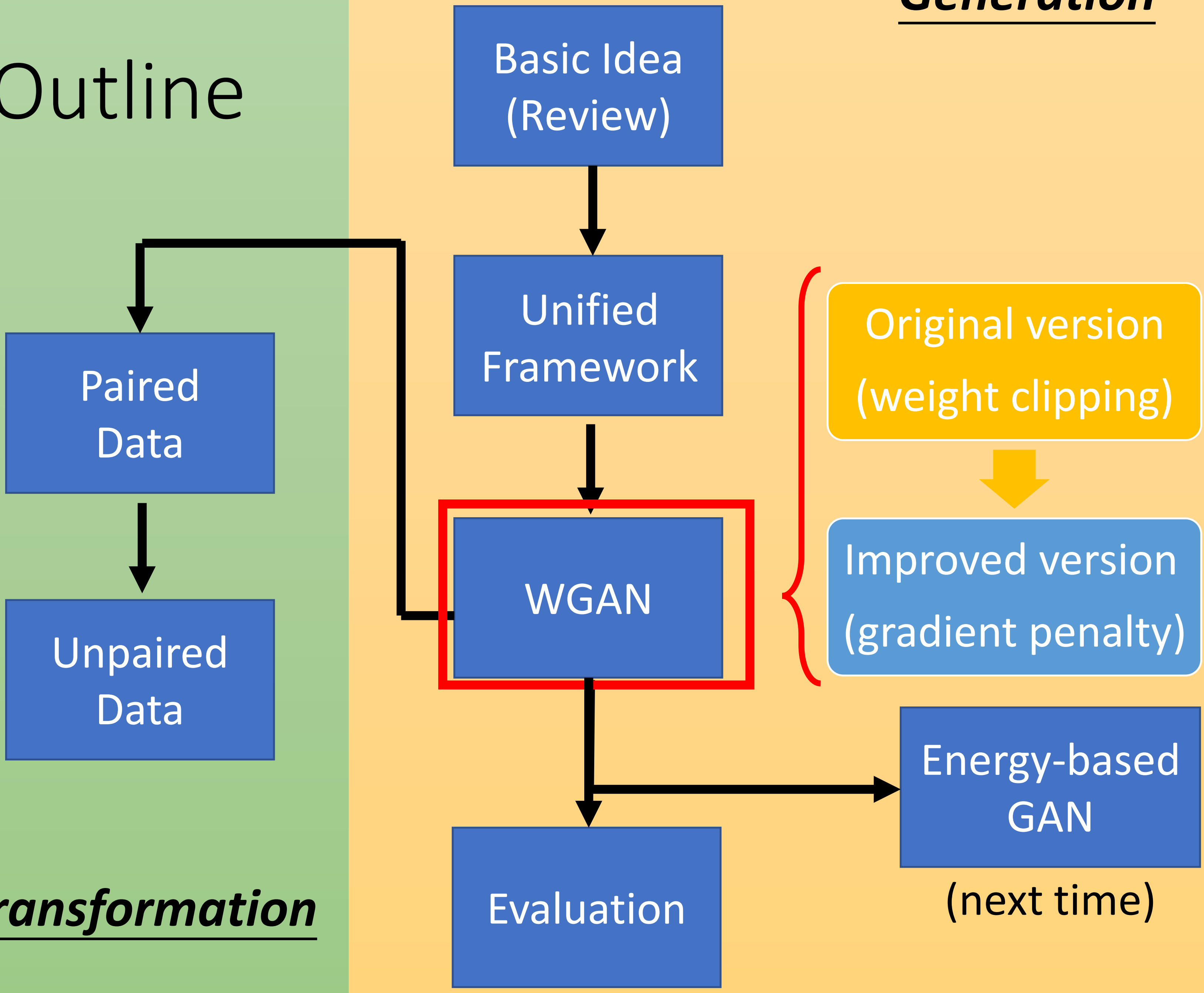
- Approximate a mixture of Gaussians by single mixture





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**

Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Original version (weight clipping)

Improved version (gradient penalty)

Energy-based GAN

(next time)

Evaluation

Paired Data

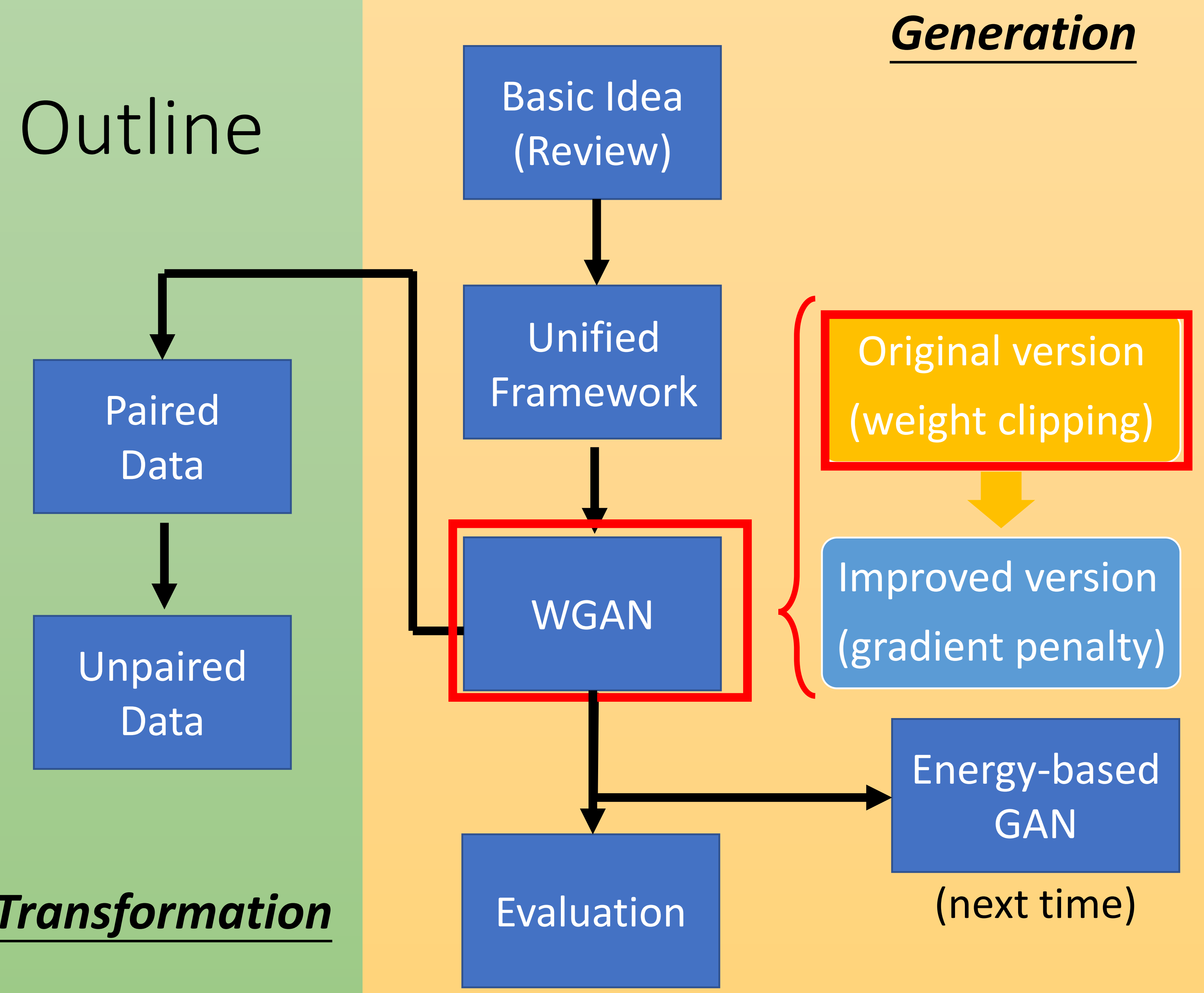
Unpaired Data

Transformation

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

Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Original version (weight clipping)

Improved version (gradient penalty)

Energy-based GAN

(next time)

Evaluation

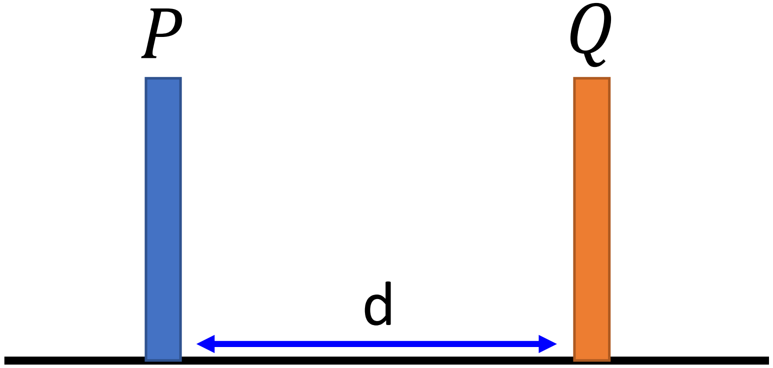
Paired Data

Unpaired Data

Transformation

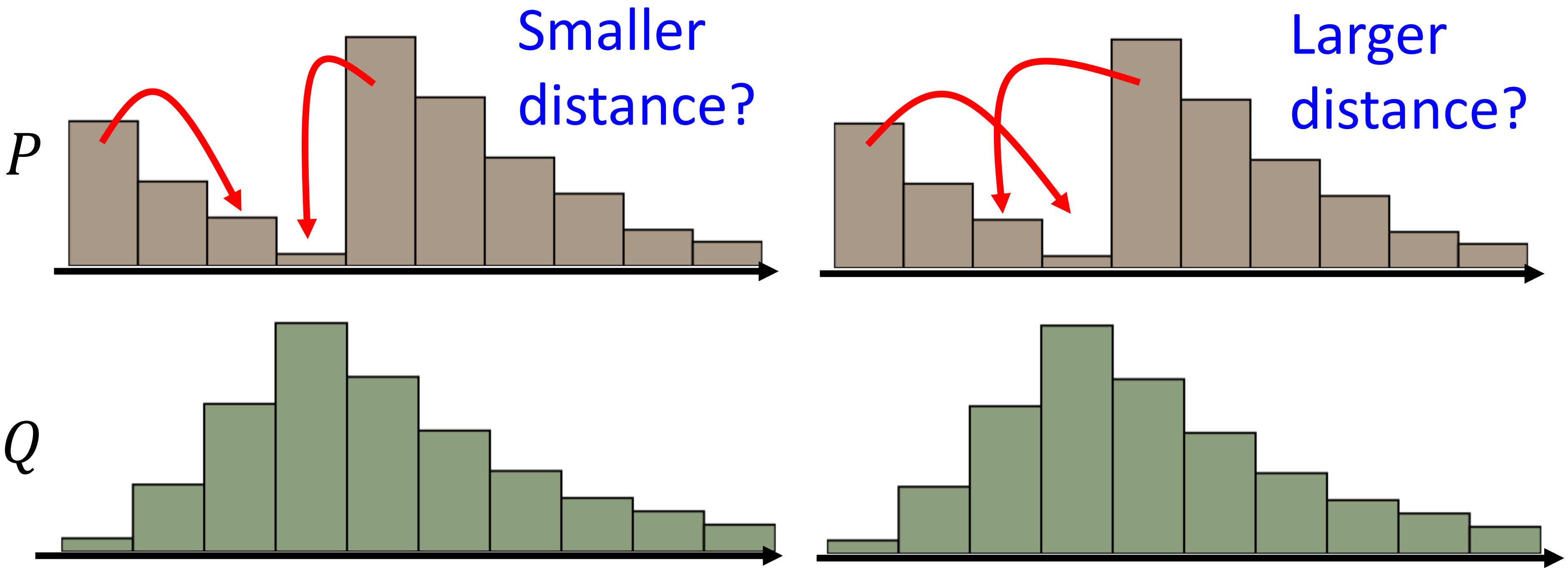
Earth Mover’s Distance

- 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



Smaller distance?

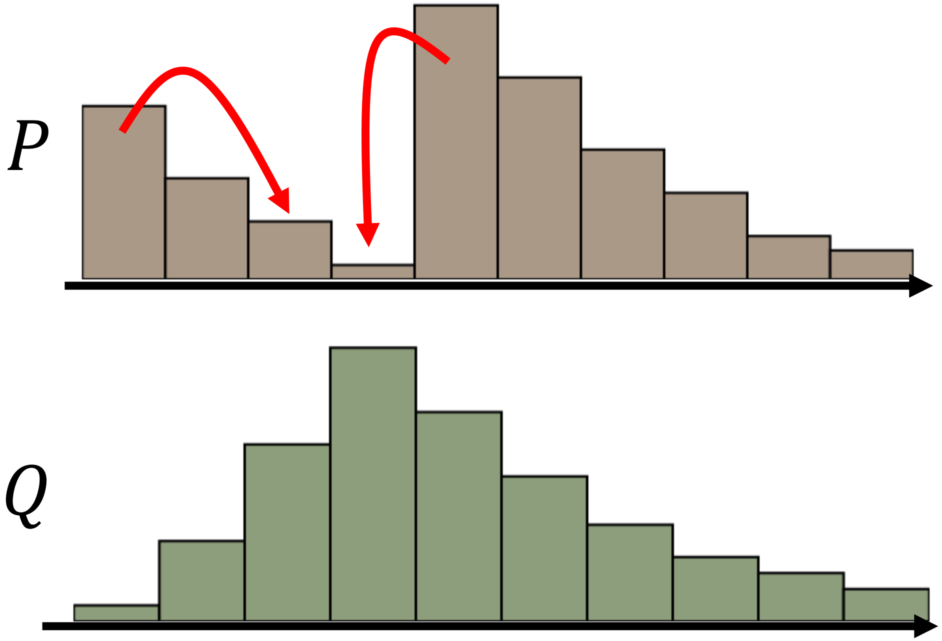
Larger distance?

There many possible “moving plans”.

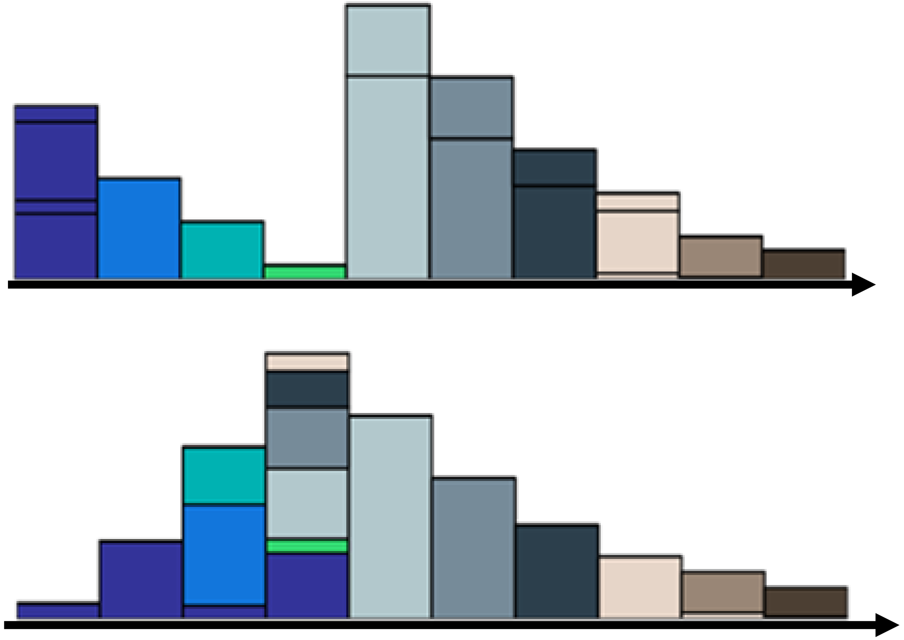
Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

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

Earth Mover’s Distance



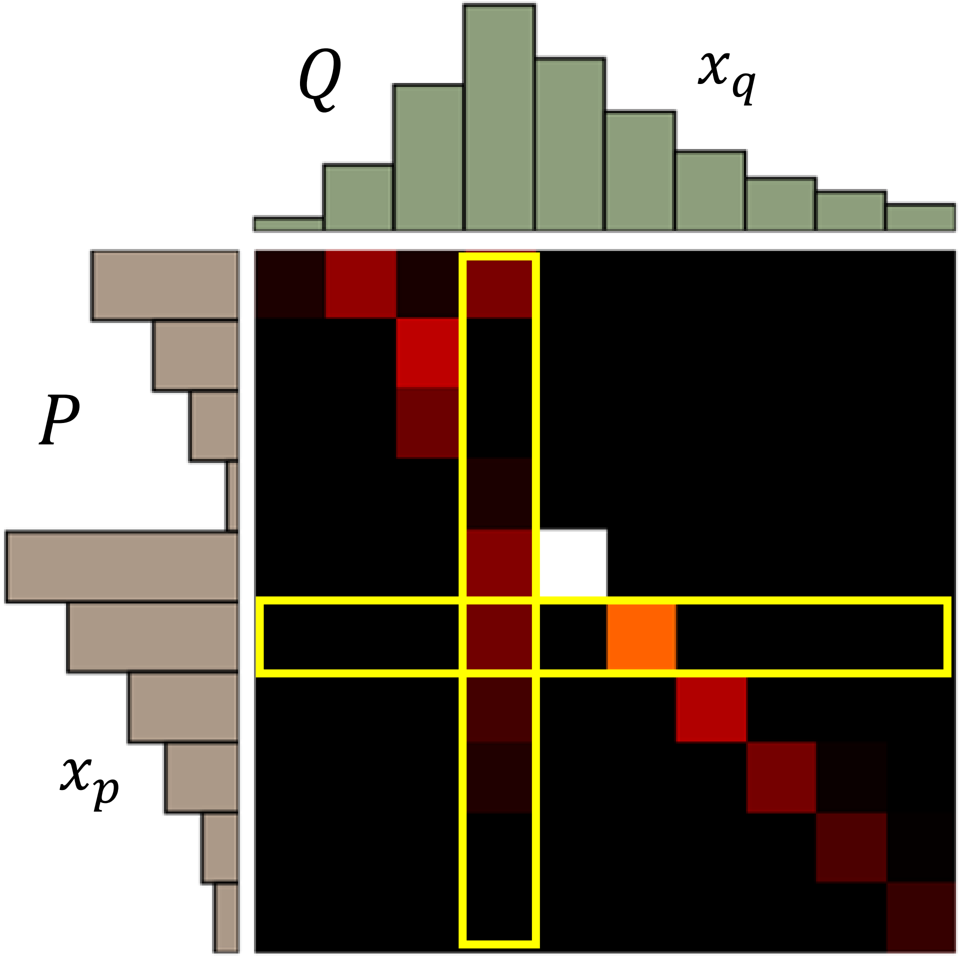
Best “moving plans” of this example



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/



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.

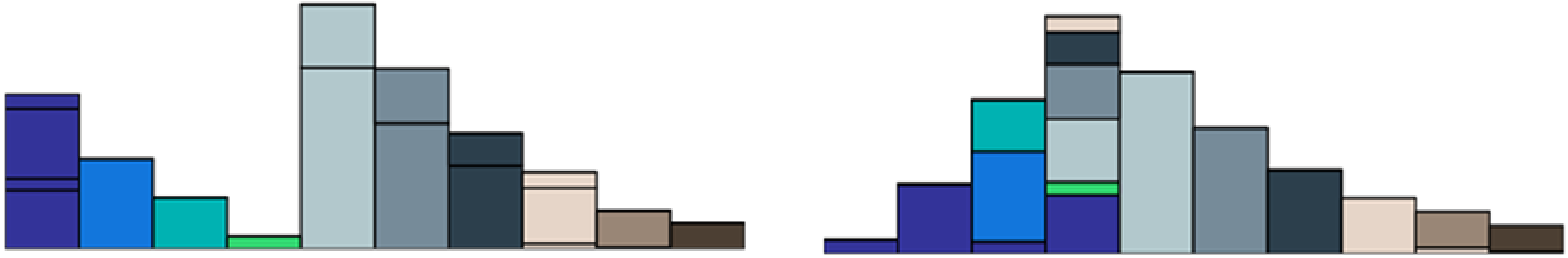
Average distance of a plan γ:

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

Earth Mover’s Distance:

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

The best plan

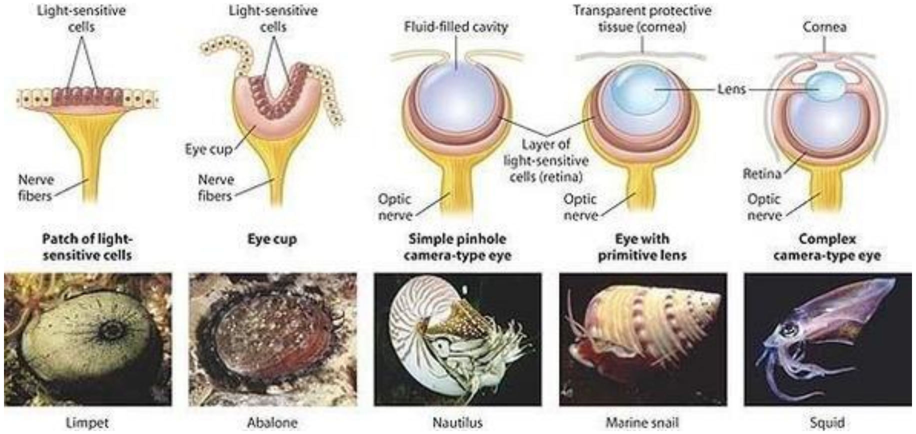


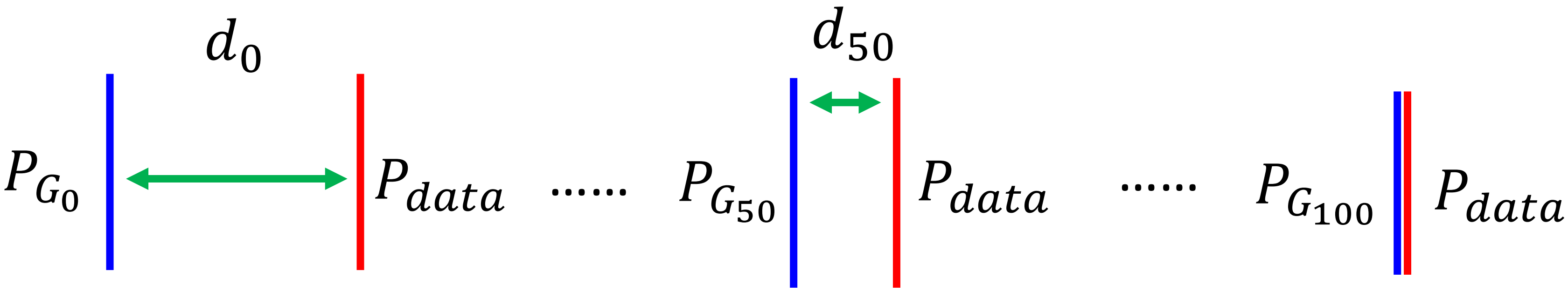
Why Earth Mover's Distance?

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

$$\Downarrow$$

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





$$JS(P_{G_0}, P_{data}) = log2$$

$$JS(P_{G_{50}}, P_{data}) = log2$$

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

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

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

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

Back to the GAN framework

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

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

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

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

Lipschitz Function

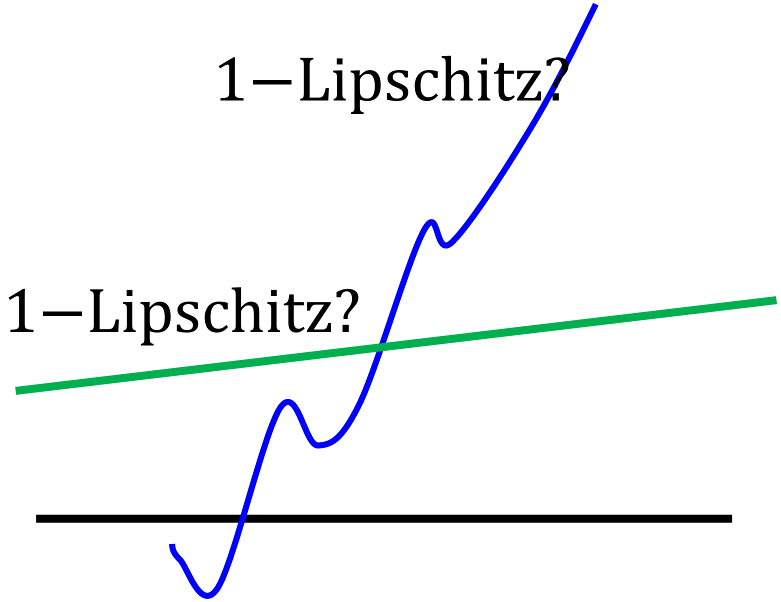
$$\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$$

Output change

Input change

K=1 for "$1 - Lipschitz$"

Do not change fast



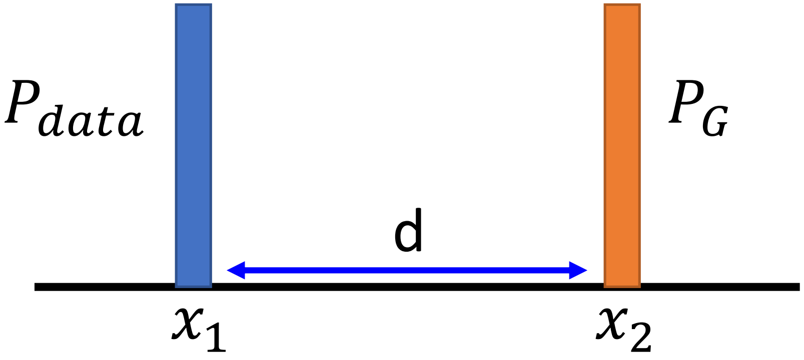
Back to the GAN framework

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

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

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

$D(x_2) = \cancel{-\infty}\ k$

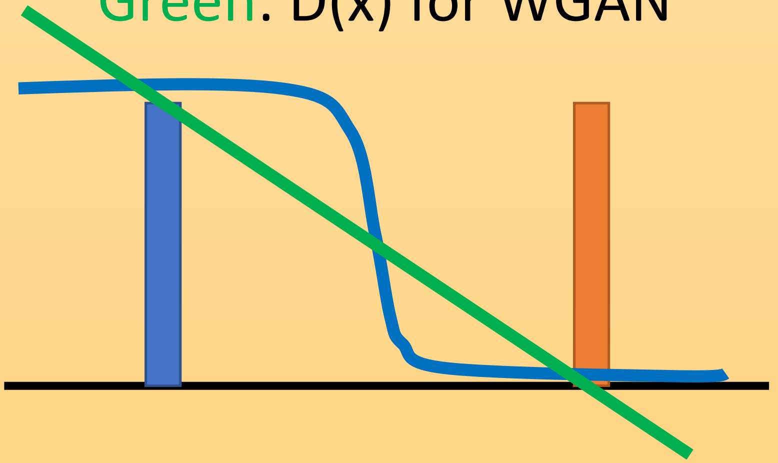


$$W(P_{data}, P_G) = d$$

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

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 $W(P_{data}, P_G)$

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

How to use gradient descent to optimize?

Weight clipping:

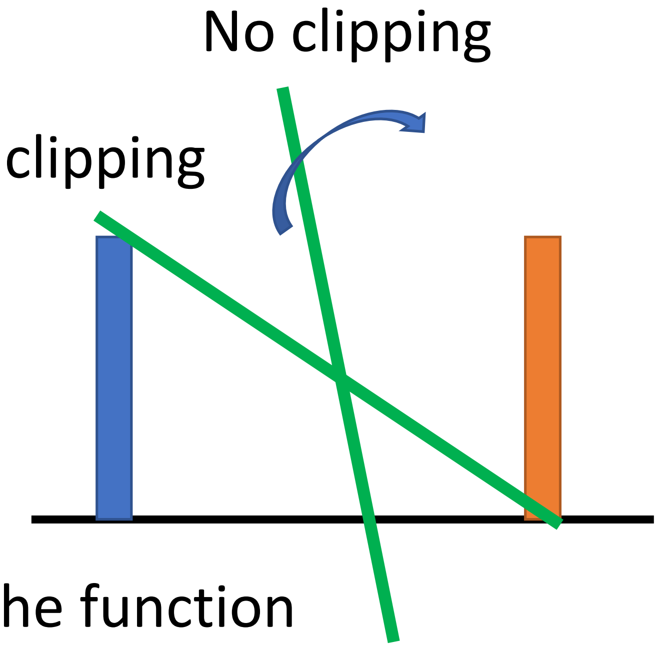
Force the weights w between c and -c

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
  - $\tilde{V} = \cancel{\frac{1}{m}\sum_{i=1}^{m} logD(x^i)} - \frac{1}{m}\sum_{i=1}^{m} D\left(G(z^i)\right)$
  - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

https://arxiv.org/abs/1701.07875

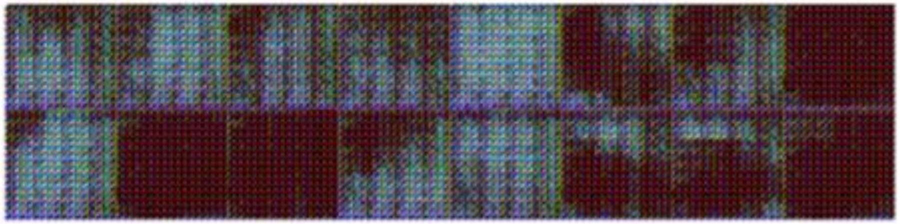
CNN generator:

W-GAN

GAN

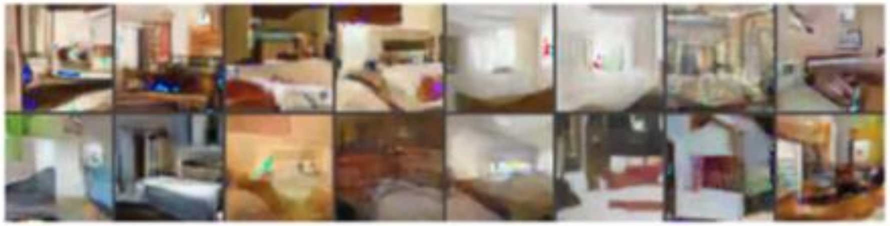
CNN generator (no batch normalization, bad structure):

W-GAN



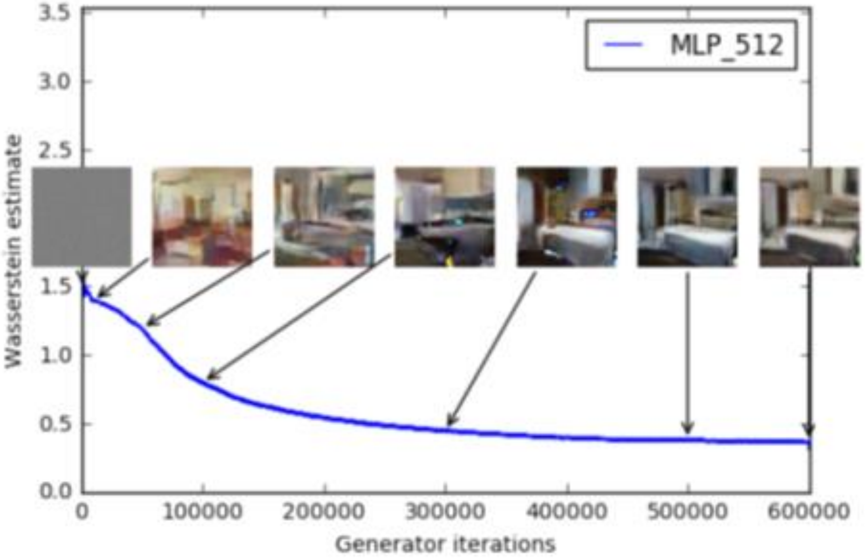
GAN

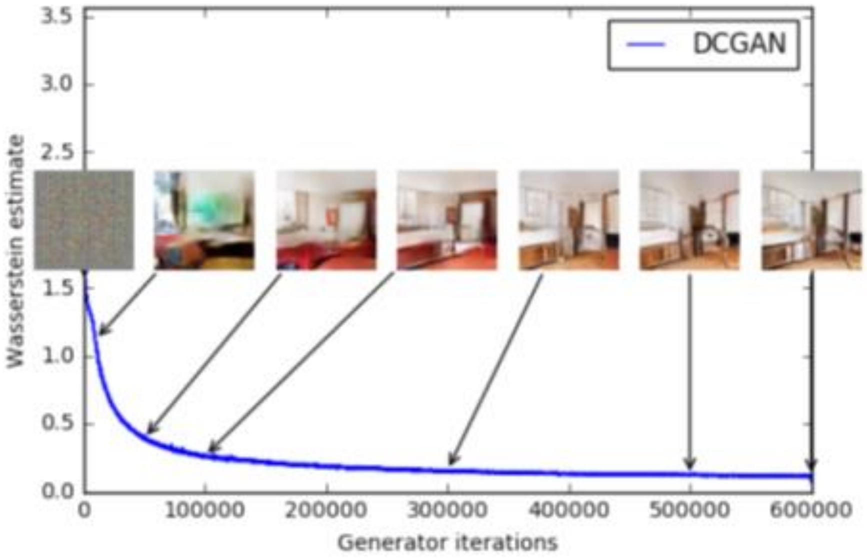
MLP generator:

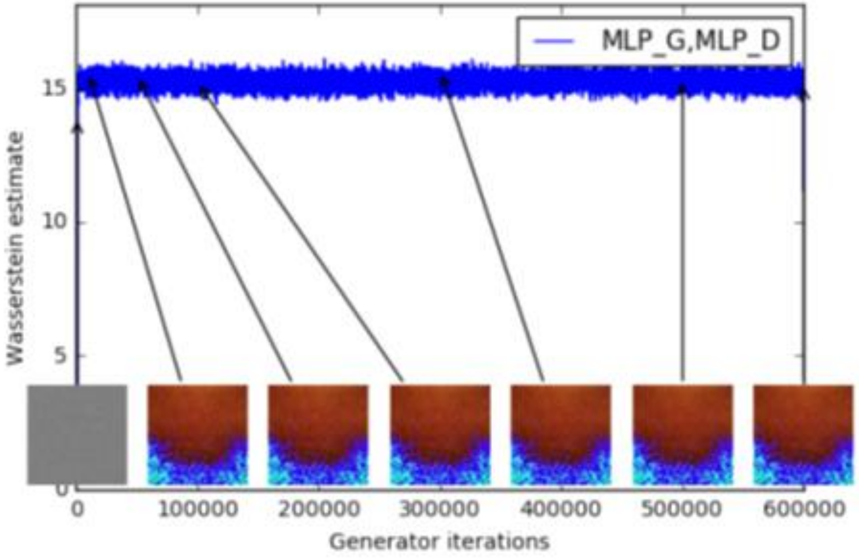


W-GAN

GAN





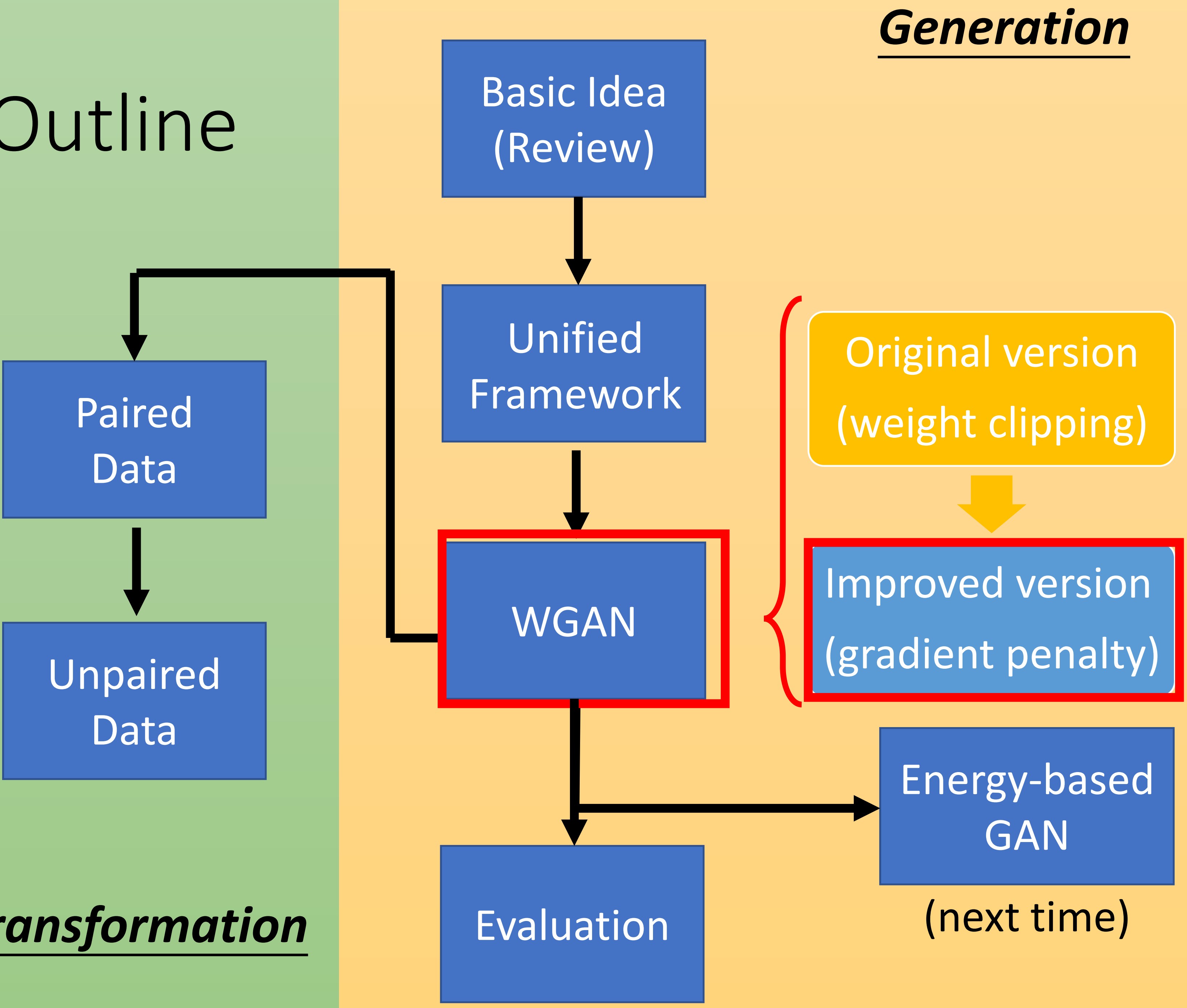


Vertical

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

https://arxiv.org/abs/1701.07875

Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Original version (weight clipping)

Improved version (gradient penalty)

Energy-based GAN

(next time)

Evaluation

Paired Data

Unpaired Data

Transformation

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.

$D \in 1 - Lipschitz$ ⟺ $\|\nabla_x D(x)\| \leq 1$ 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

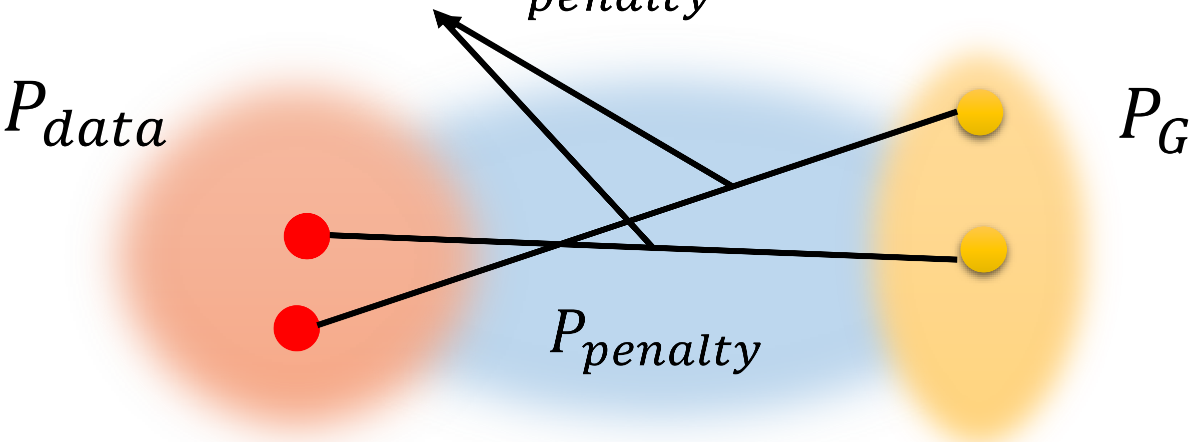
⬇

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

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.”

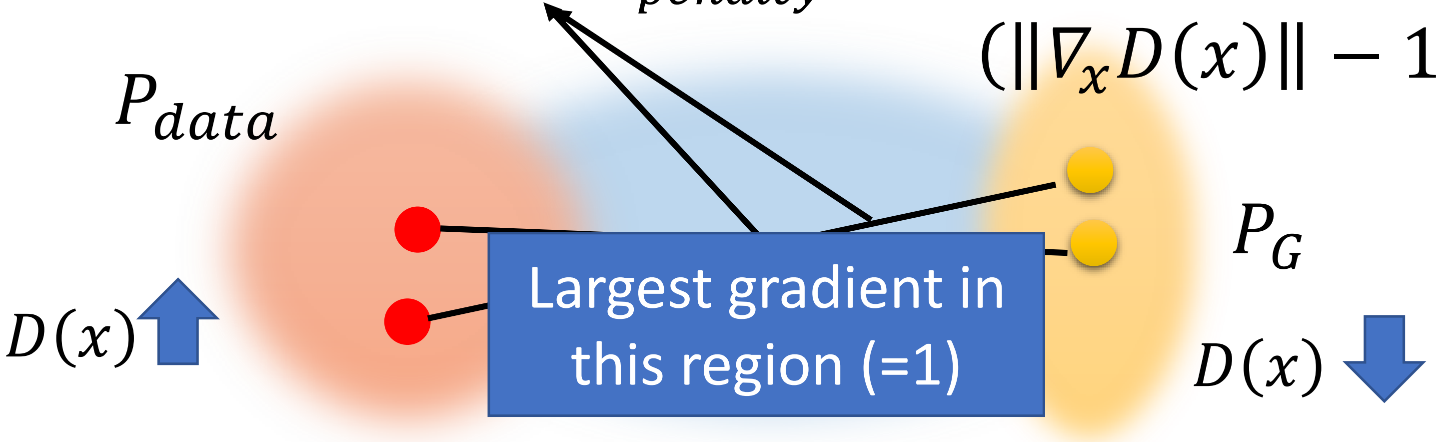
Only give gradient constraint to the region between P_{data} and P_G because they influence how P_G moves to P_{data}

Improved WGAN

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

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

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



Largest gradient in this region (=1)

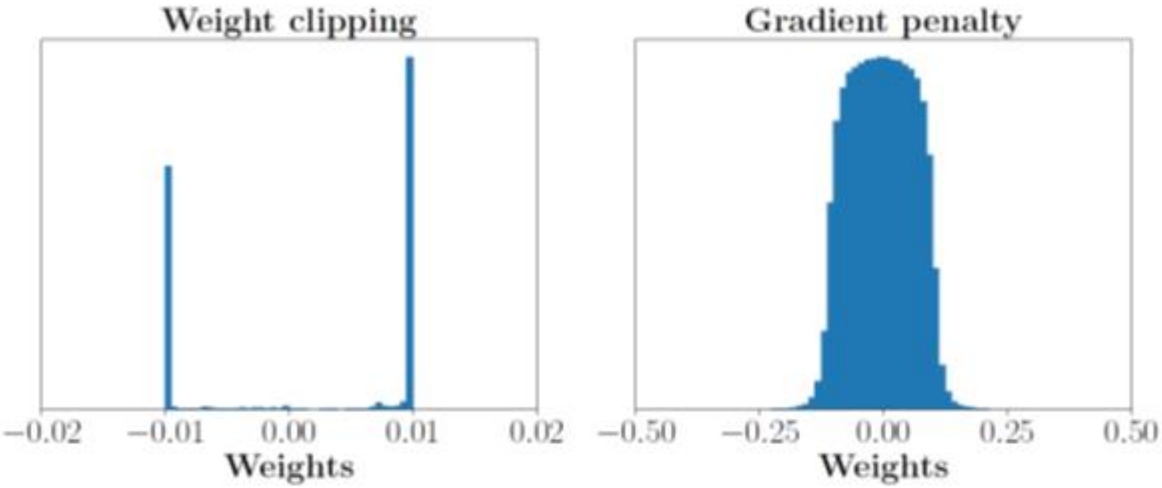
“One may wonder why we penalize the norm of the gradient for differing from 1, instead of just penalizing large gradients. The reason is that the optimal critic … actually has gradients with norm 1 almost everywhere under Pr and Pg”

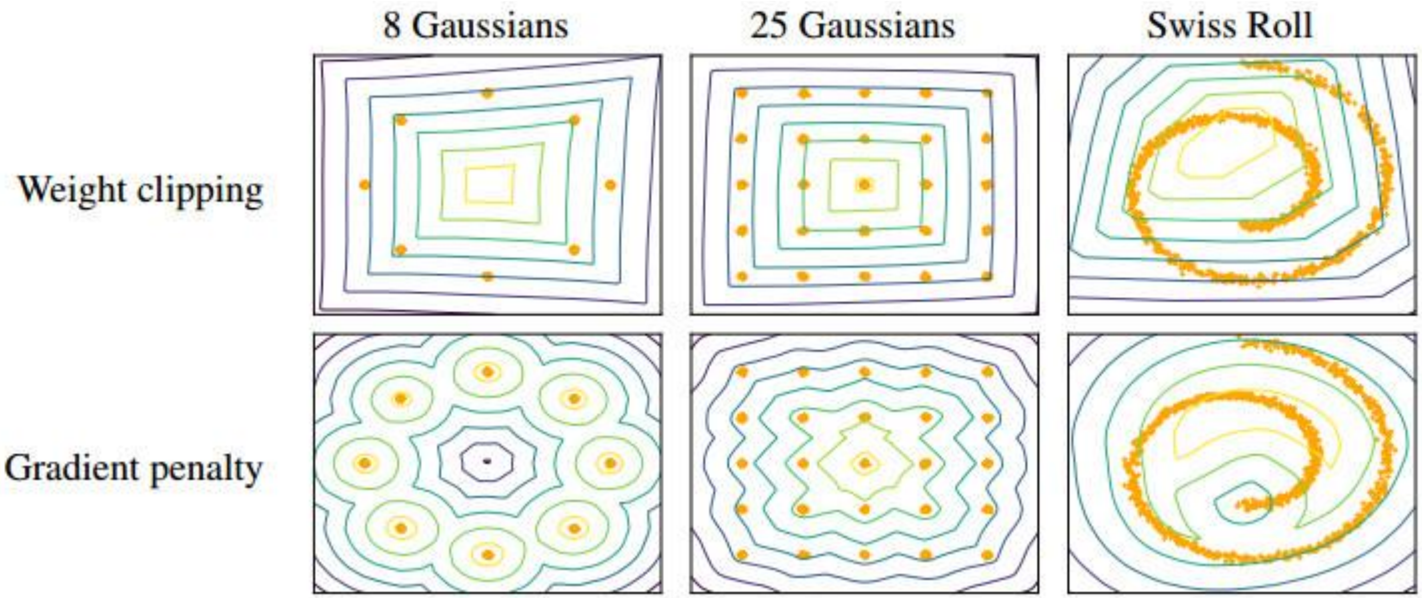
(check the proof in the appendix)

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

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)

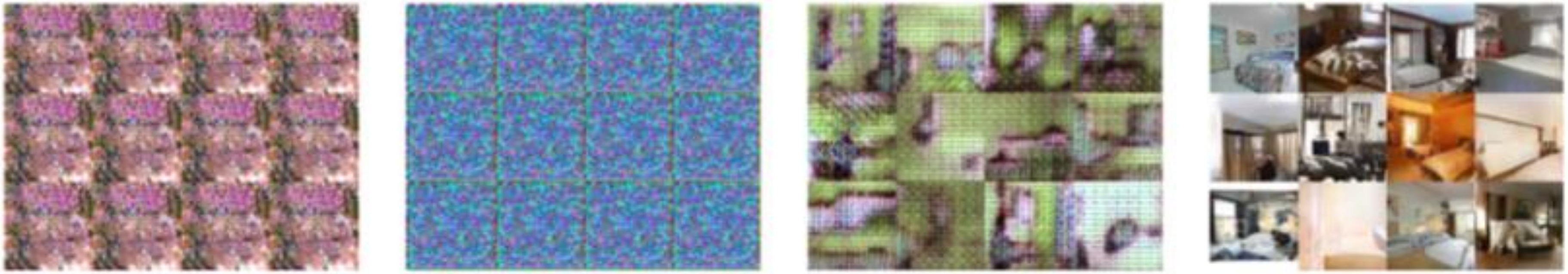
G: CNN (tanh), D: CNN(tanh)

DCGAN | LSGAN | Original WGAN | Improved WGAN

G: MLP, D: CNN

G: CNN (bad structure), D: CNN

G: 101 layer, D: 101 layer



Sentence Generation

- good bye.

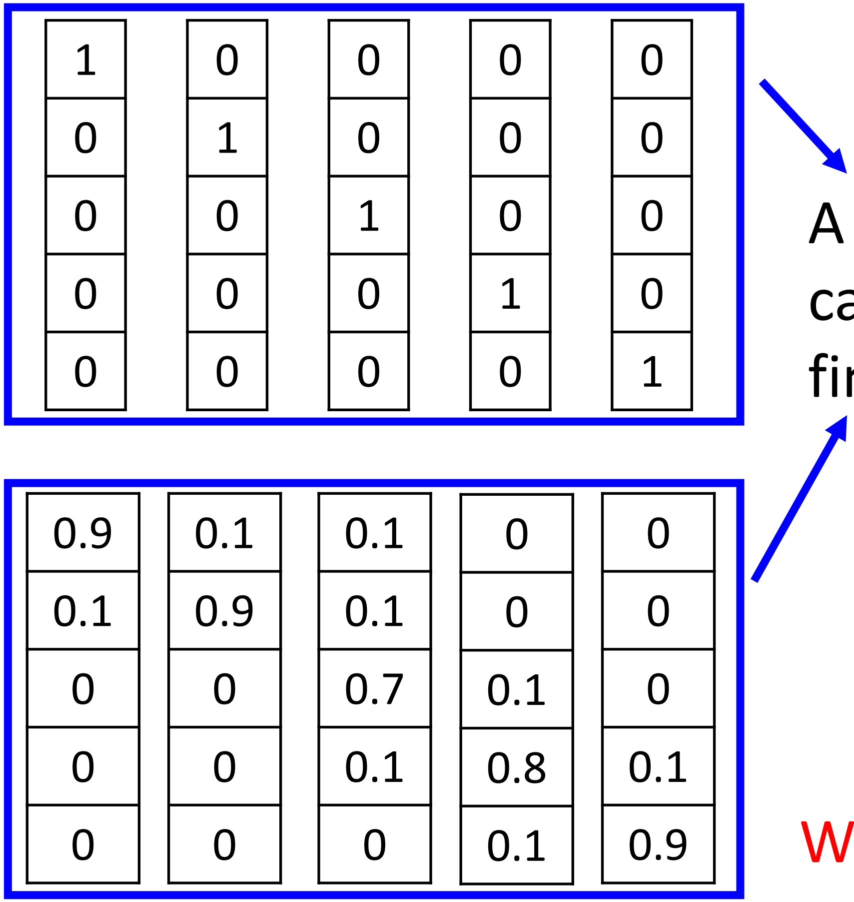
1-of-N Encoding

	g	o	o	d	<space>	……
	⋮					
d	0	0	0	1	1	
	⋮					
g	1	0	0	0	0	
	⋮					
o	0	1	1	0	0	
	⋮					
<space>	0	0	0	0	1	

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

- 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

Can never be 1-of-N

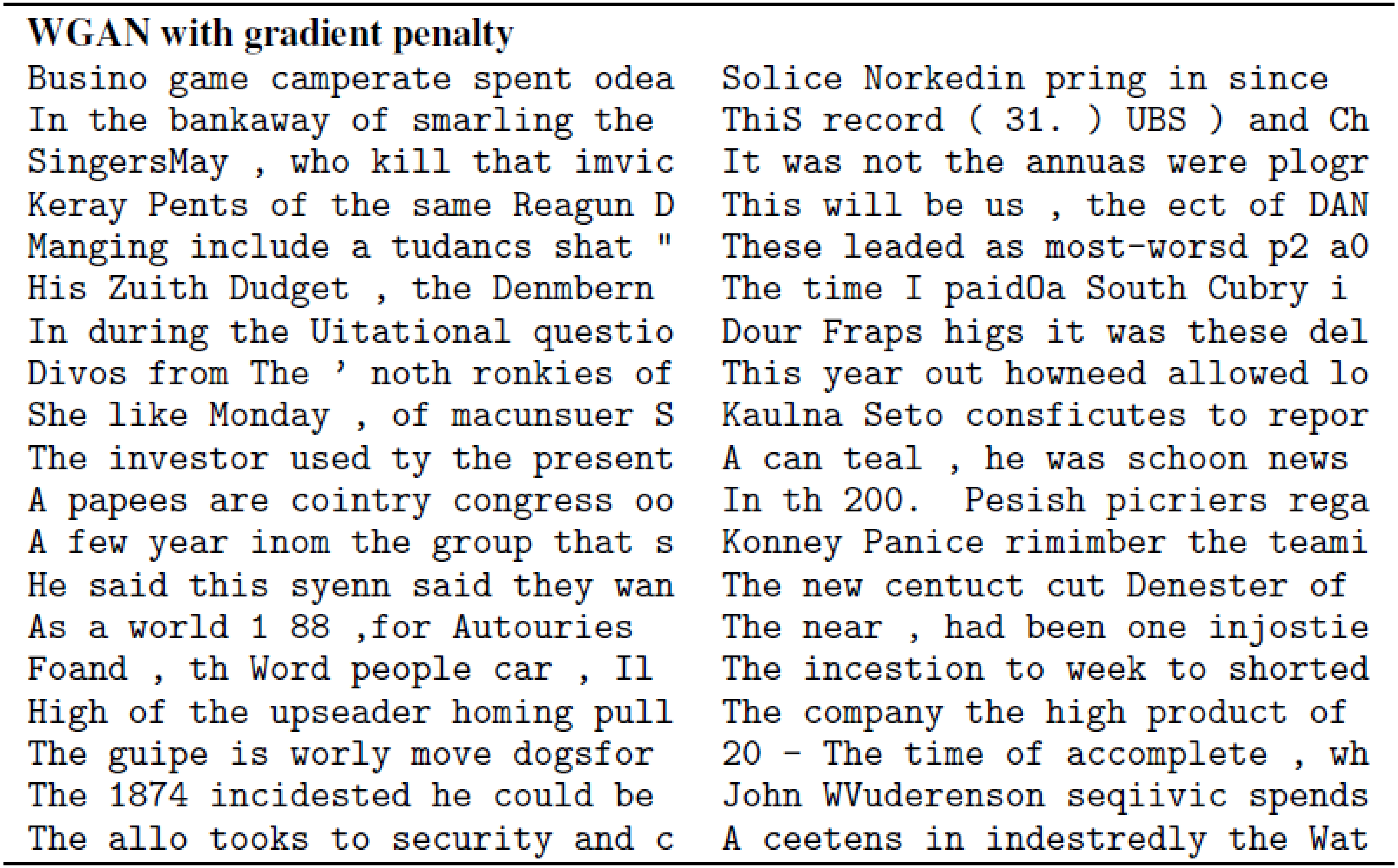
A binary classifier can immediately find the difference.

No overlap

WGAN is helpful

Improved WGAN successfully generating sentences

https://arxiv.org/abs/1704.00028



WGAN with gradient penalty

```
Busino game camperate spent odea     Solice Norkedin pring in since
In the bankaway of smarling the      ThiS record ( 31. ) UBS ) and Ch
SingersMay , who kill that imvic     It was not the annuas were plogr
Keray Pents of the same Reagun D     This will be us , the ect of DAN
Manging include a tudancs shat "     These leaded as most-worsd p2 a0
His Zuith Dudget , the Denmbern      The time I paid0a South Cubry i
In during the Uitational questio     Dour Fraps higs it was these del
Divos from The ' noth ronkies of     This year out howneed allowed lo
She like Monday , of macunsuer S     Kaulna Seto consficutes to repor
The investor used ty the present     A can teal , he was schoon news
A papees are cointry congress oo     In th 200.  Pesish picriers rega
A few year inom the group that s     Konney Panice rimimber the teami
He said this syenn said they wan     The new centuct cut Denester of
As a world 1 88 ,for Autouries       The near , had been one injostie
Foand , th Word people car , Il      The incestion to week to shorted
High of the upseader homing pull     The company the high product of
The guipe is worly move dogsfor      20 - The time of accomplete , wh
The 1874 incidested he could be      John WVuderenson seqiivic spends
The allo tooks to security and c     A ceetens in indestredly the Wat
```

W-GAN – 唐詩鍊成

感謝 李仲翊 同學提供實驗結果

輸出 32 個字 (包含標點)

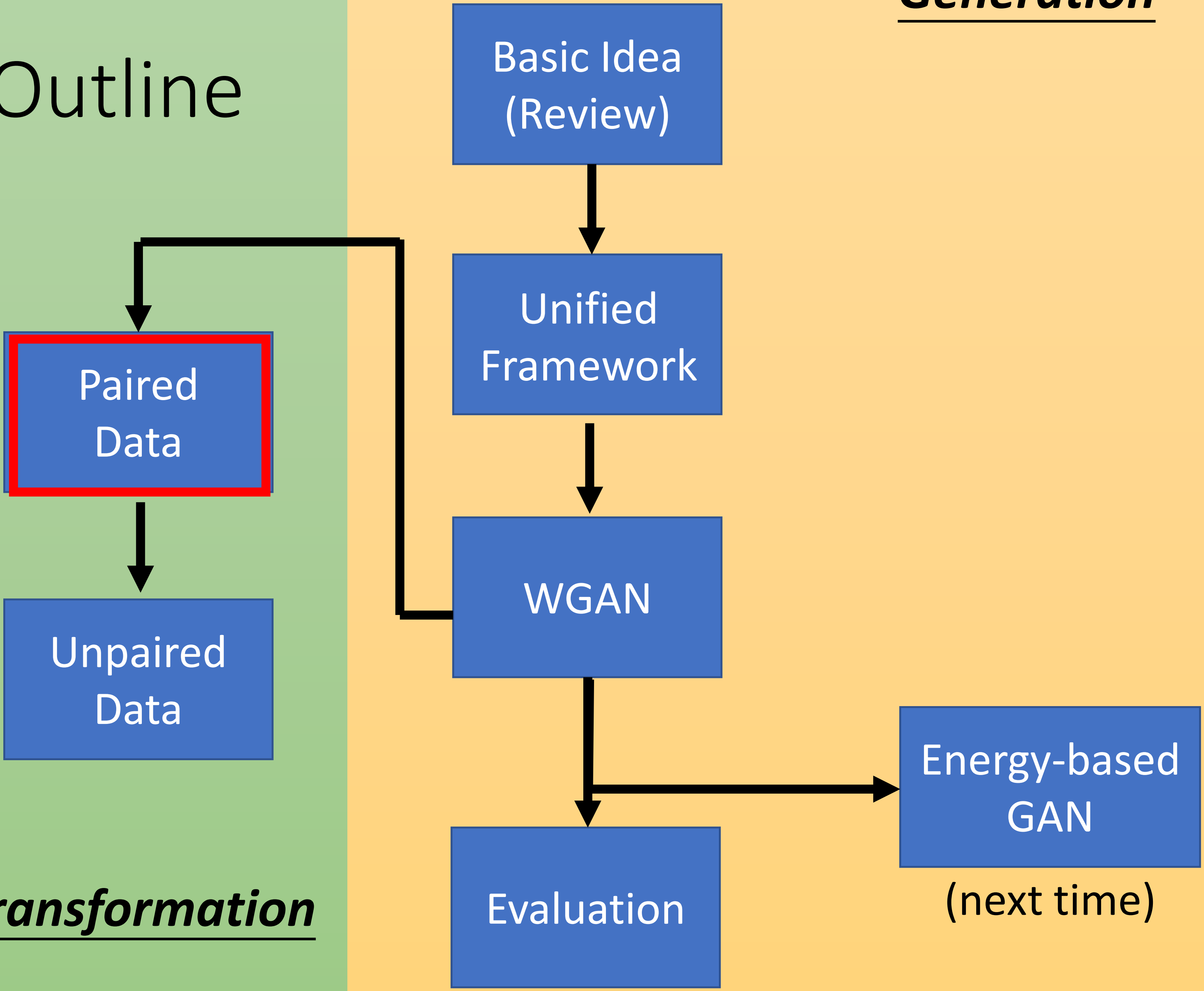
- 升雲白遲丹齋取，此酒新巷市入頭。黃道故海歸中後，不驚入得韻子門。
- 摢口容章蓄翎翎，邦貸無遊隔將毬。外蕭曾臺遶出畧，此計推上呂天夢。
- 新來寶伎泉，手雪泓臺蓑。曾子花路魏，不謀散薦船。
- 功持牧度機邀爭，不躚官嬉牧涼散。不迎白旅今掩冬，盡蘸金衹可停。
- 玉十洪沄爭春風，溪子風佛挺橫鞋。盤盤稅焰先花齋，誰過飄鶴一丞幢。
- 海人依野庇，為阻例沉迴。座花不佐樹，弟闌十名儂。
- 入維當興日世瀕，不評皺。頭醉空其杯，駸園凋送頭。
- 鉢笙動春枝，寶絫潔長知。官爲窓爛去，絆粒薛一靜。
- 吾涼腕不楚，縱先待旅知。楚人縱酒待，一蔓飄聖猜。
- 折幕故癘應韻子，徑頭霜瓊老徑徑。尚錯春鏘熊悽梅，去吹依能九將香。
- 通可矯目鷚須淨，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，帛陽舊摢甿婷儻。

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

Outline

Generation



Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

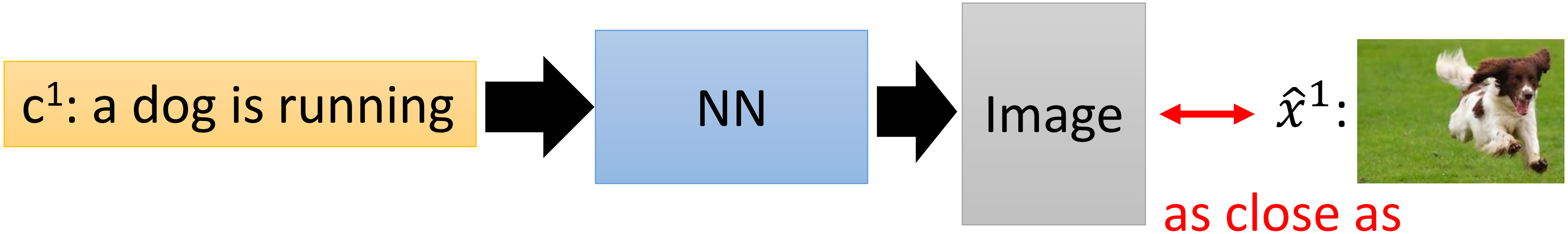
Transformation

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



c^1: a dog is running

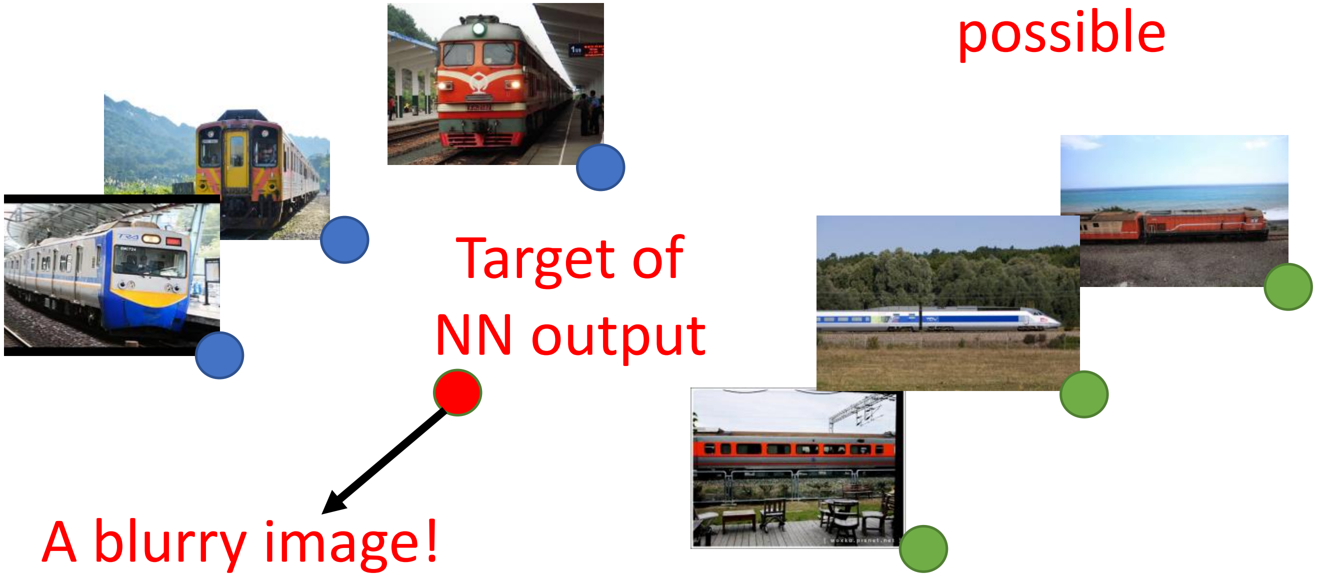
NN

Image

$\hat{x}^1$:

as close as possible

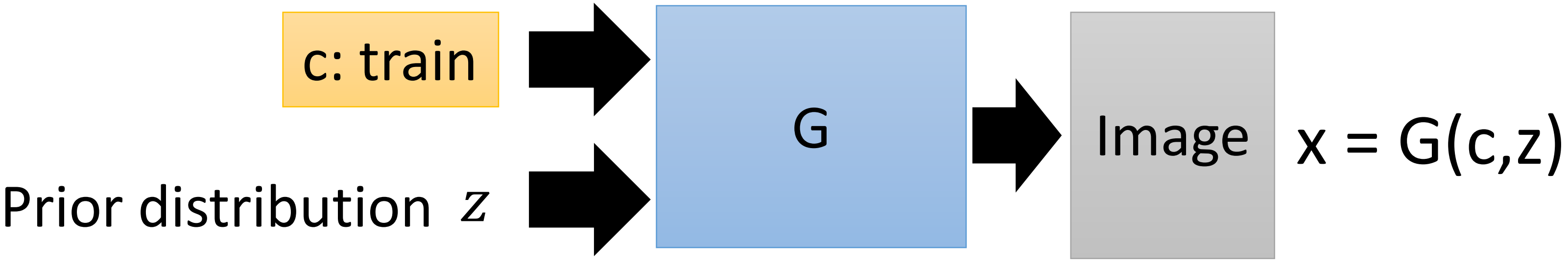
Text: “train”



Target of NN output

A blurry image!

Conditional GAN



c: train

Prior distribution z

G

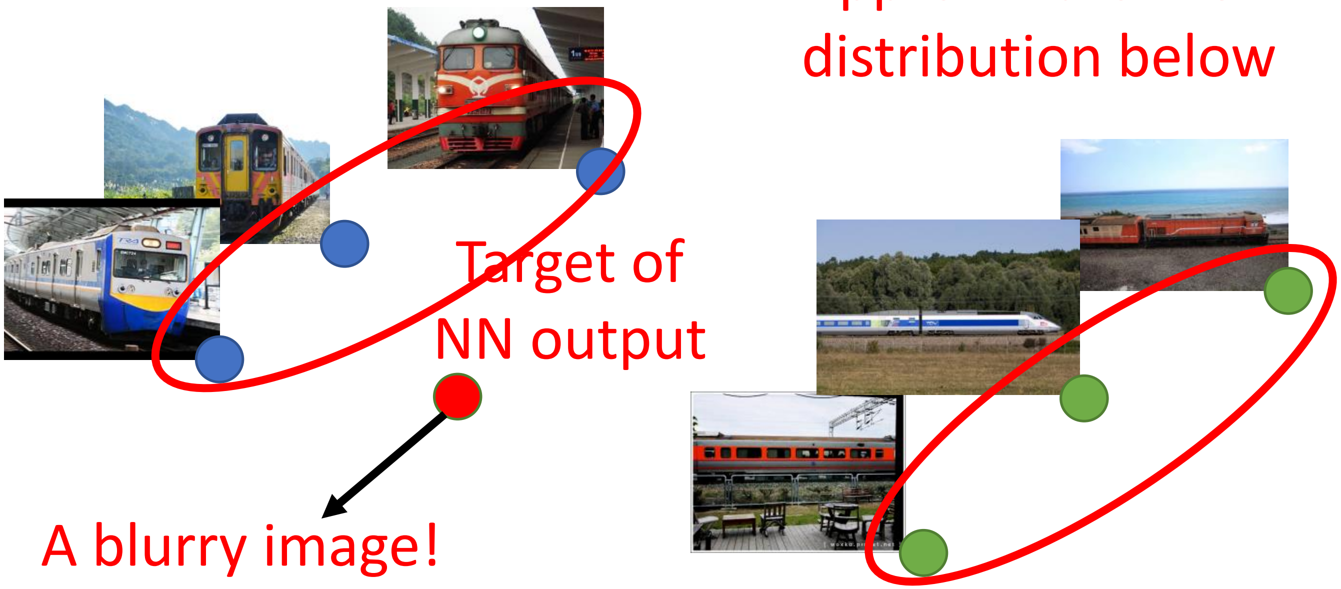
Image

$x = G(c,z)$

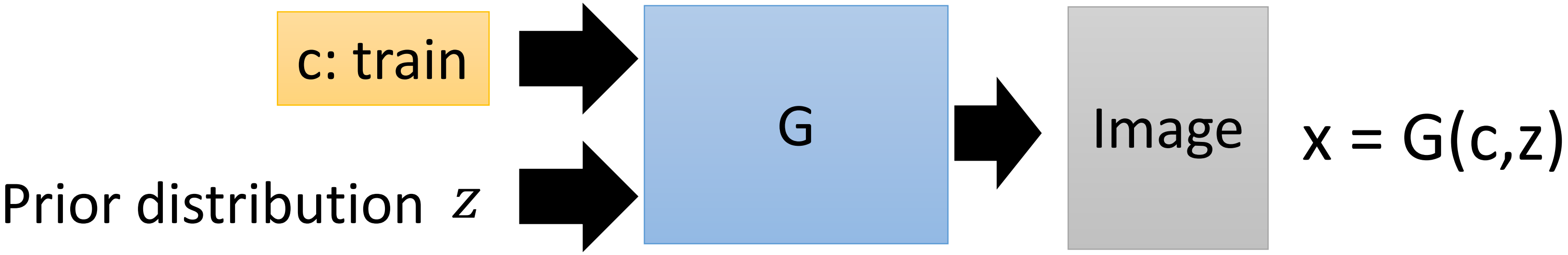
It is a distribution.

Approximate the distribution below

Text: “train”



Conditional GAN



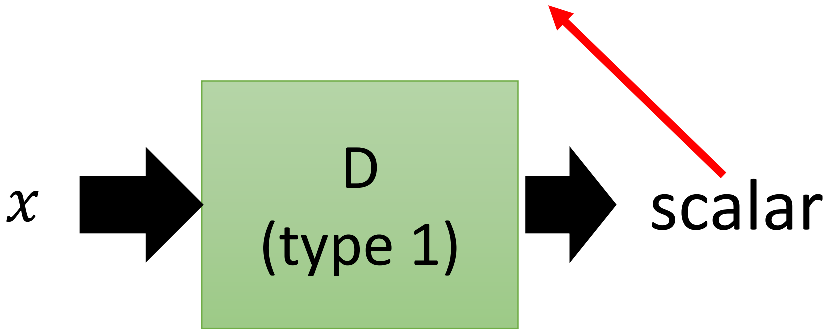
c: train

Prior distribution z

G

Image

x = G(c,z)

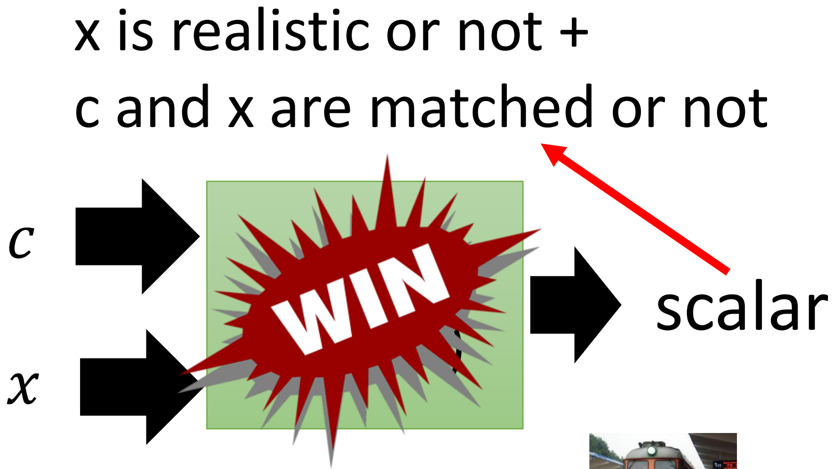


x is realistic or not

x

D
(type 1)

scalar



x is realistic or not +
c and x are matched or not

c

x

scalar

Positive example: 

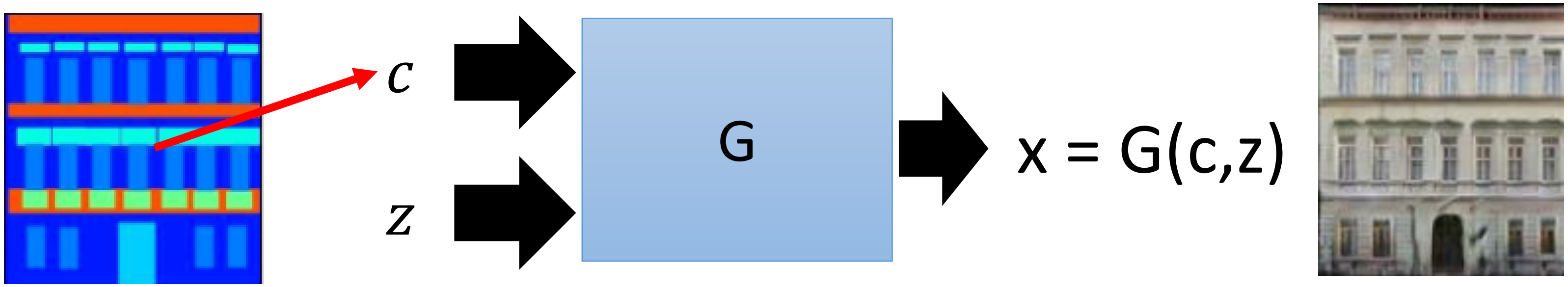
Negative example: Image

Positive example: (train , )

Negative example:

(train , Image) (cat , )

Image-to-image

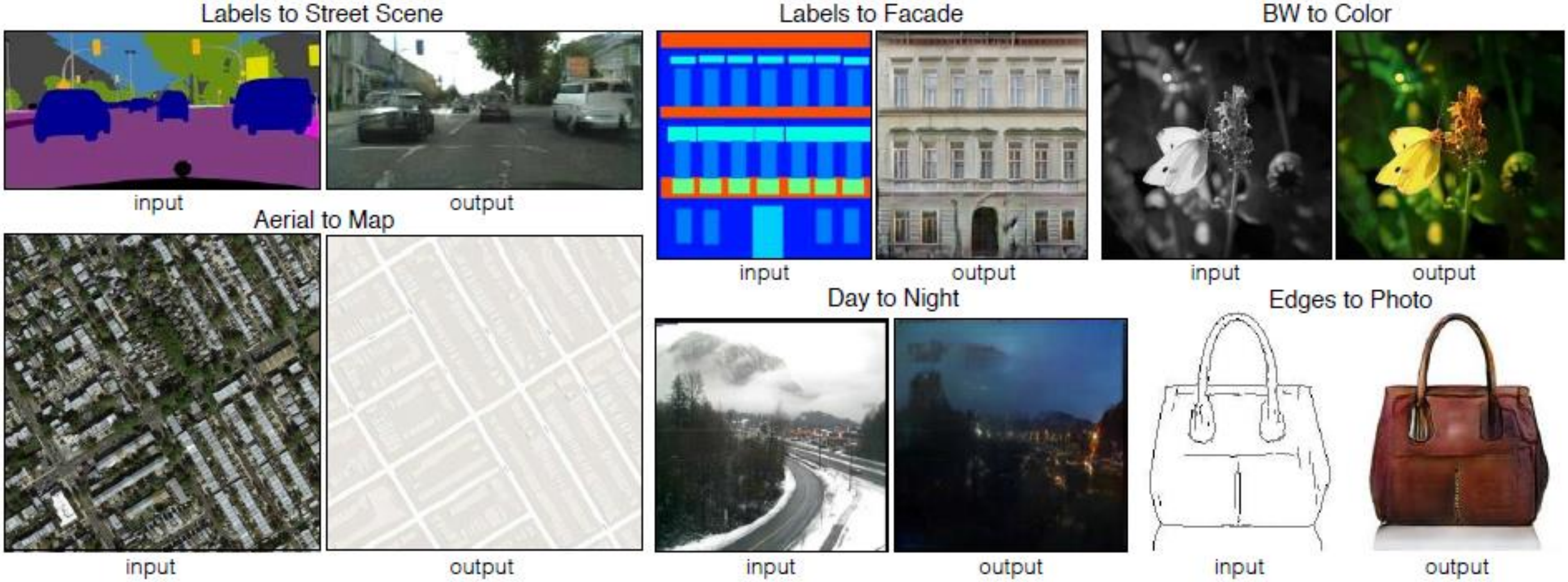


c

z

G

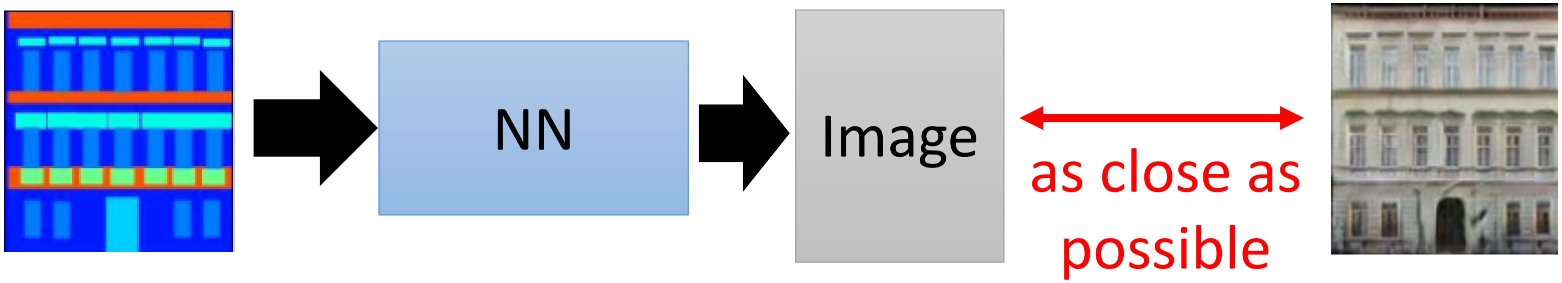
x = G(c,z)



https://arxiv.org/pdf/1611.07004

Image-to-image

- Traditional supervised approach

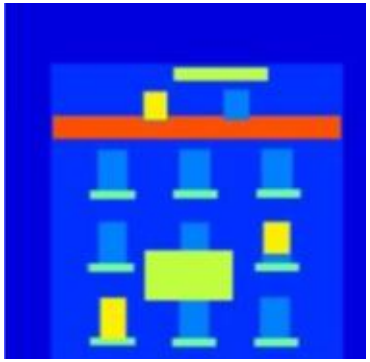


NN

Image

as close as possible

Testing:



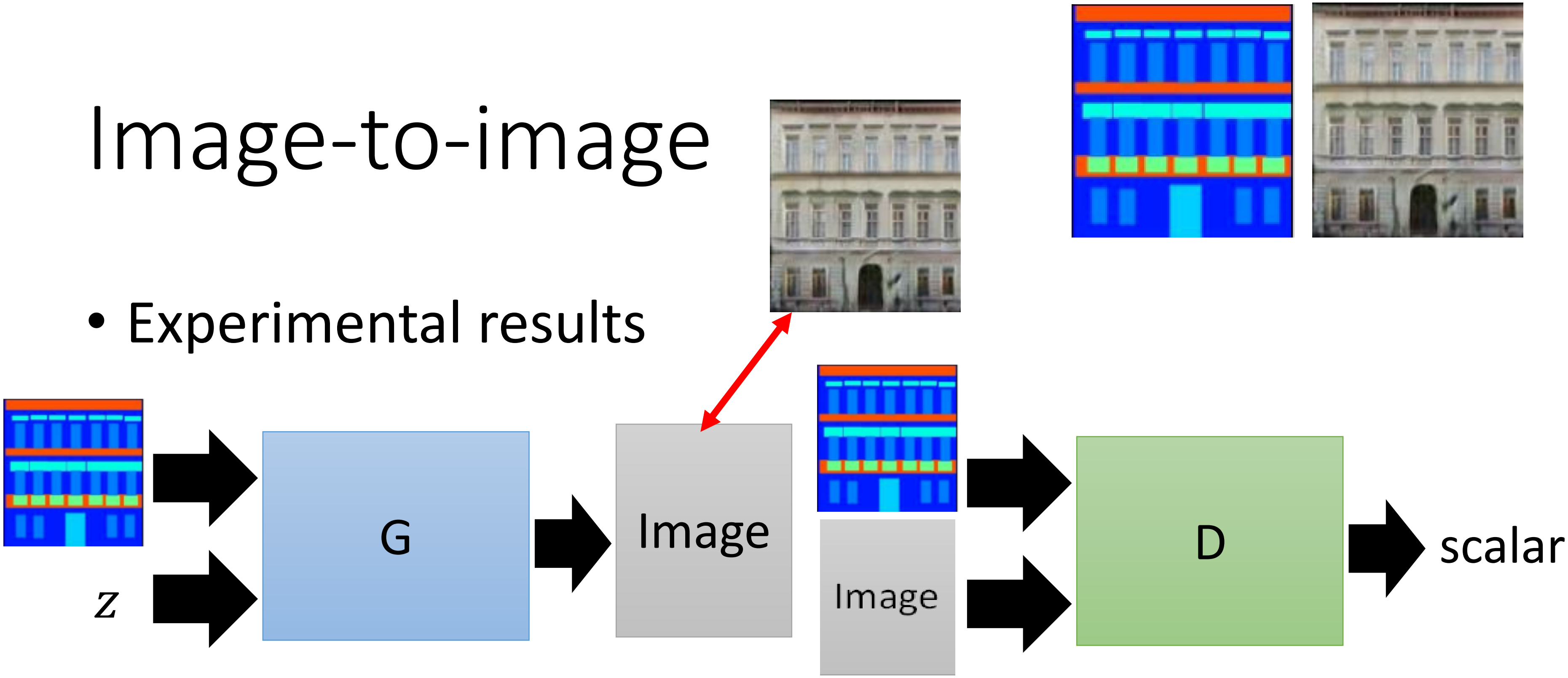
input

close

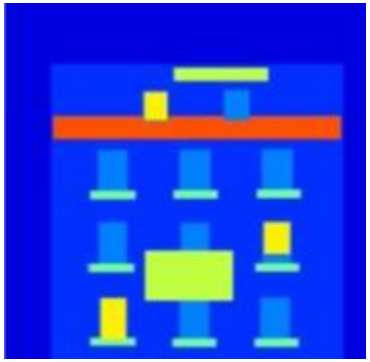
It is blurry because it is the average of several images.

Image-to-image

- Experimental results



Testing:



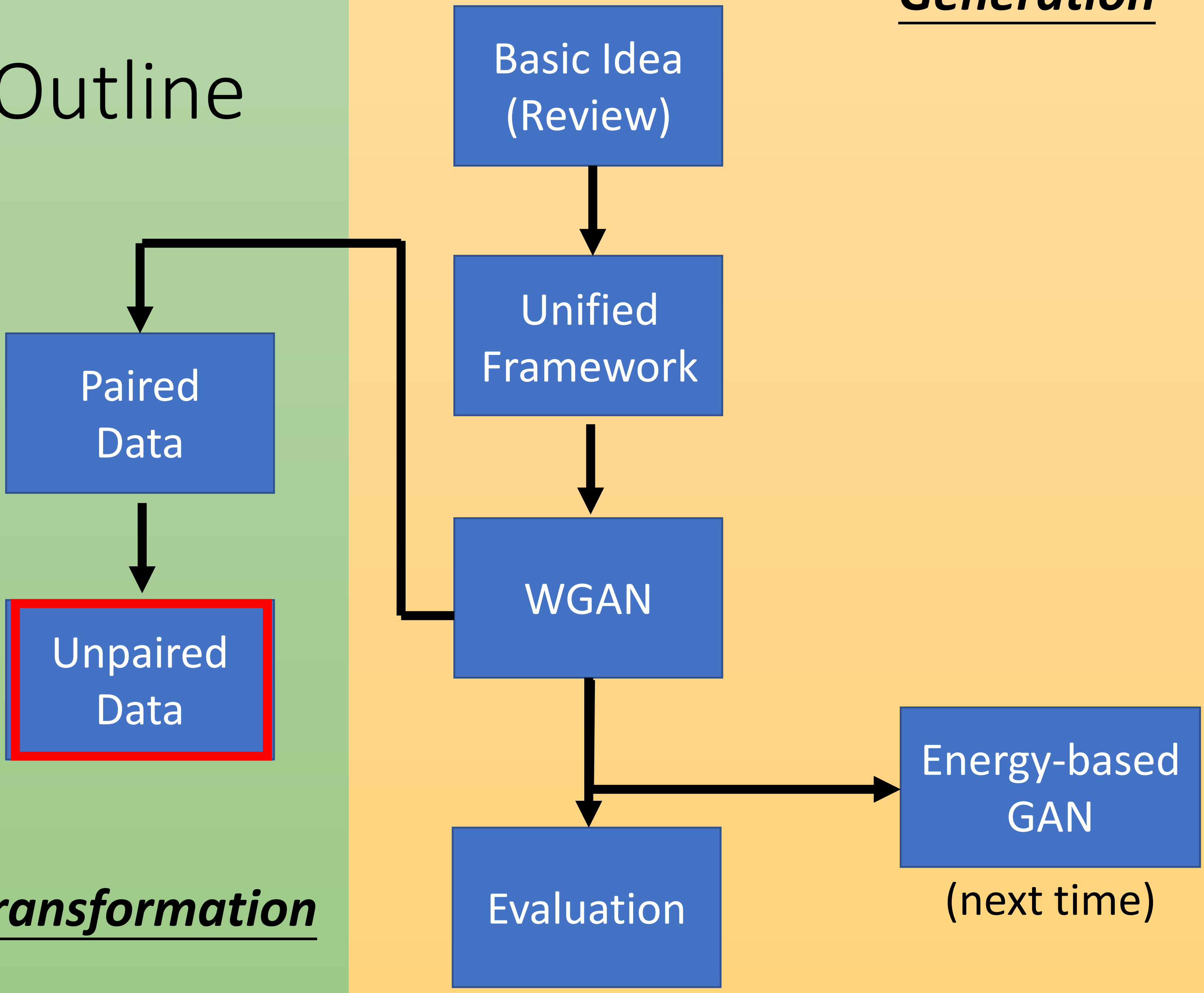
input

close

GAN

GAN + close

Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

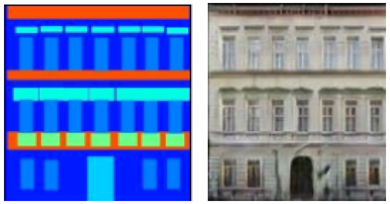
Transformation

Paired Data

Unpaired Data

Unpaired Transformation
- Cycle GAN, Disco GAN

paired data



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

Domain X

Domain Y

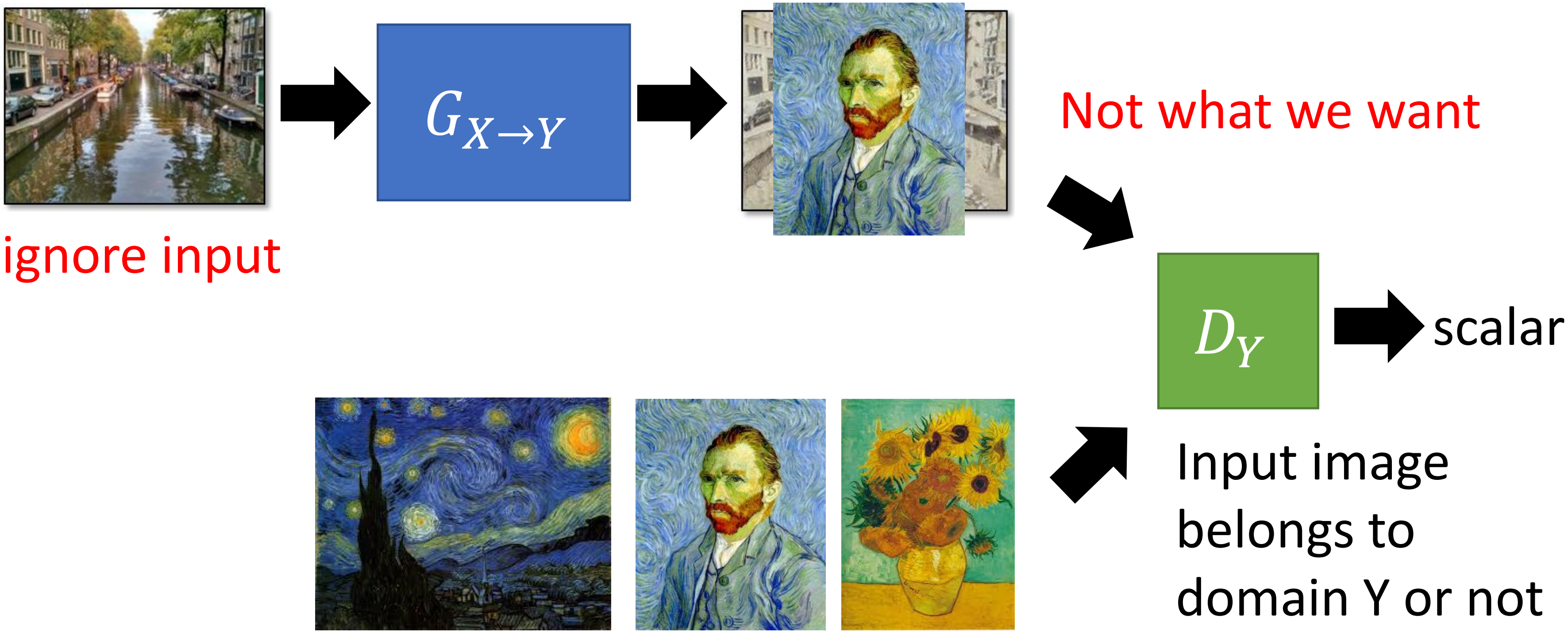
Cycle GAN

https://arxiv.org/abs/1703.10593
https://junyanz.github.io/CycleGAN/

Domain X

Domain Y

Domain X



$G_{X\to Y}$

Become similar to domain Y

Not what we want

ignore input

D_Y

scalar

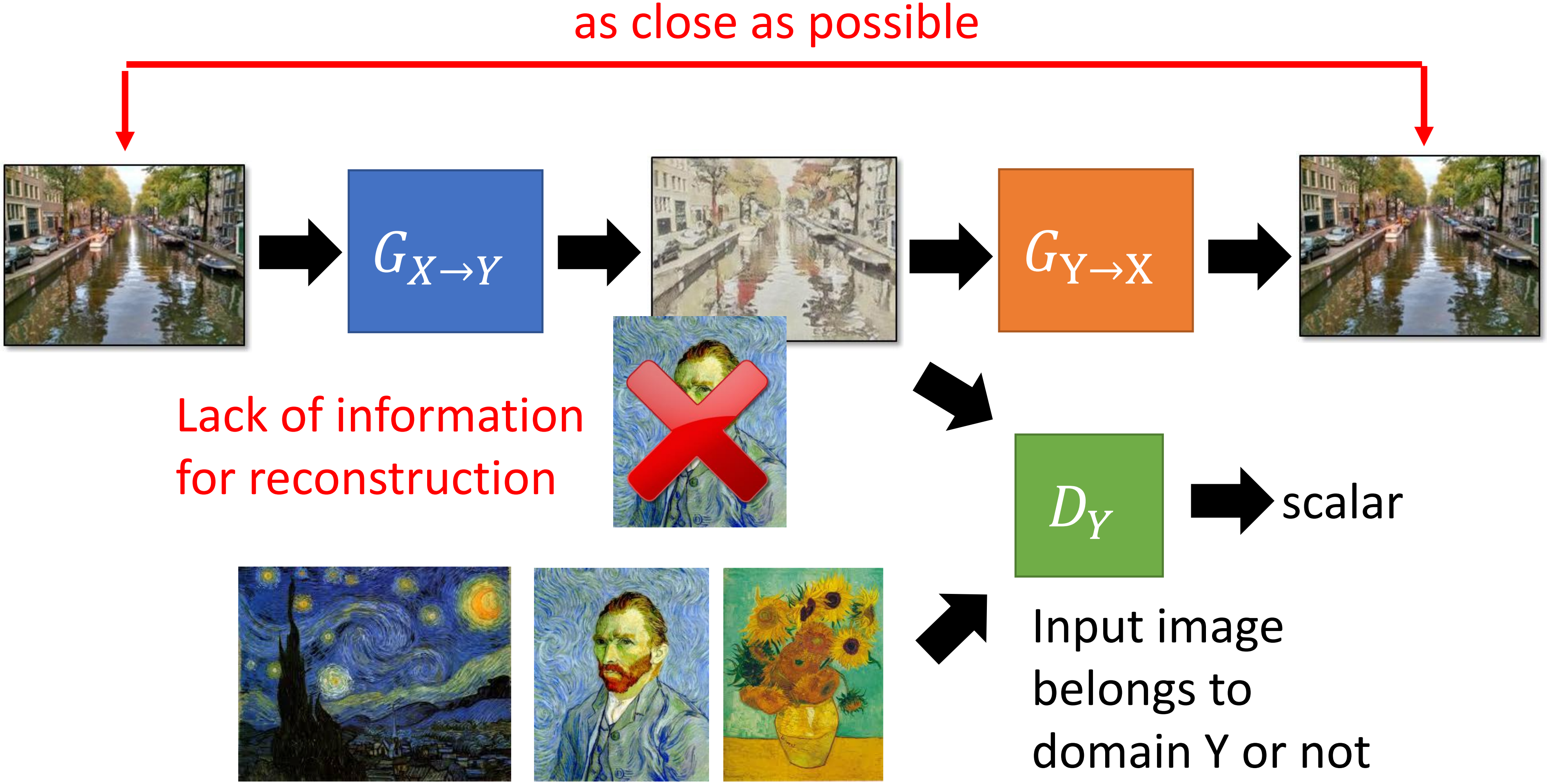
Input image belongs to domain Y or not

Domain Y

Cycle GAN

Domain X

Domain Y



as close as possible

$G_{X\to Y}$

$G_{Y\to X}$

Lack of information for reconstruction

D_Y

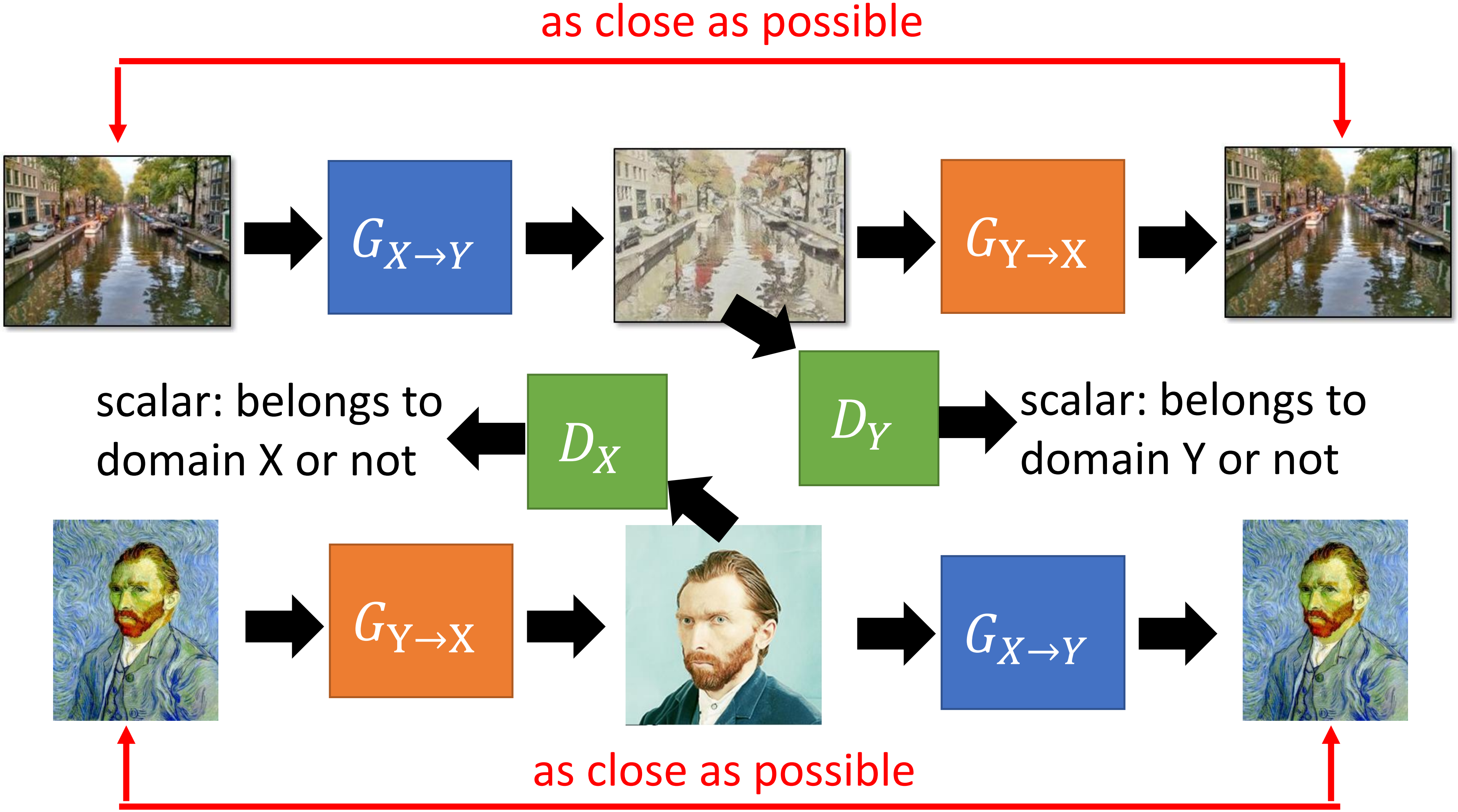
scalar

Input image belongs to domain Y or not

Domain Y

Cycle GAN


Domain X
Domain Y


as close as possible
$G_{X\rightarrow Y}$
$G_{Y\rightarrow X}$
D_Y
scalar: belongs to domain Y or not
D_X
scalar: belongs to domain X or not
$G_{Y\rightarrow X}$
$G_{X\rightarrow Y}$
as close as possible

Unpaired Transformation – 真人動畫化

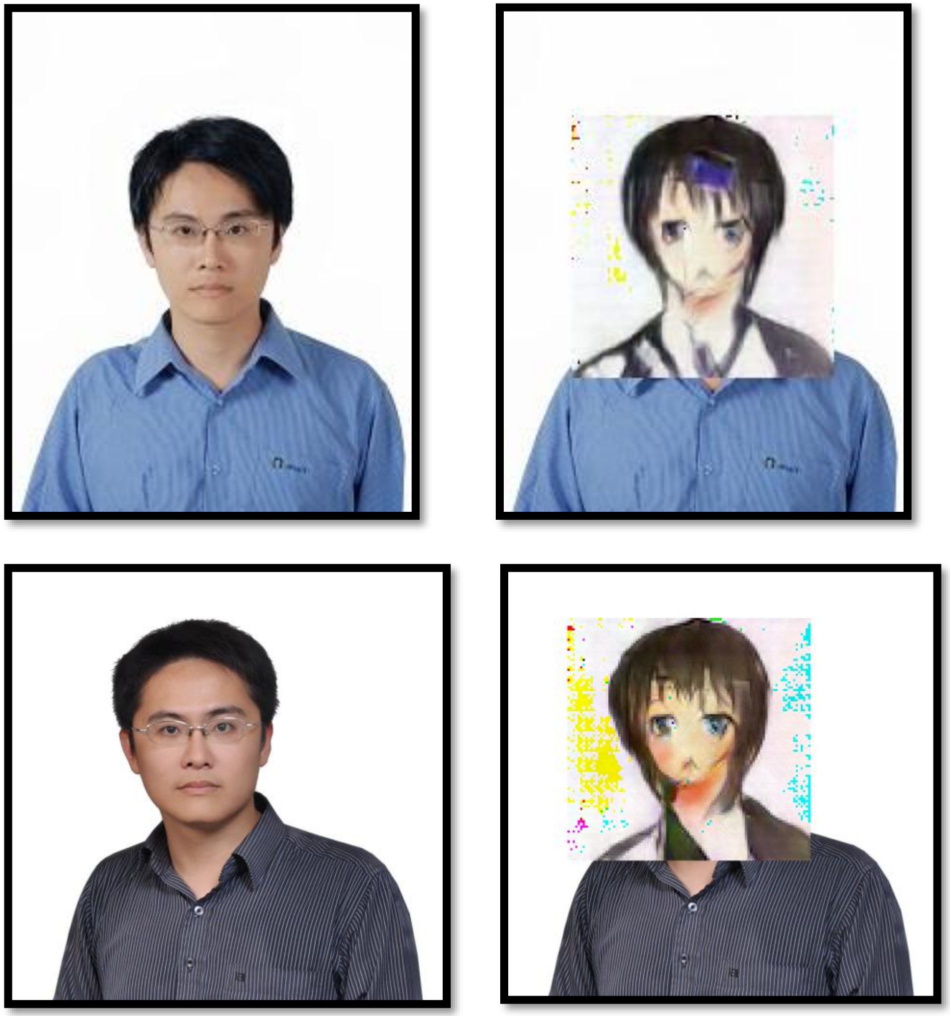
- http://qiita.com/Hi-king/items/8d36d9029ad1203aac55

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

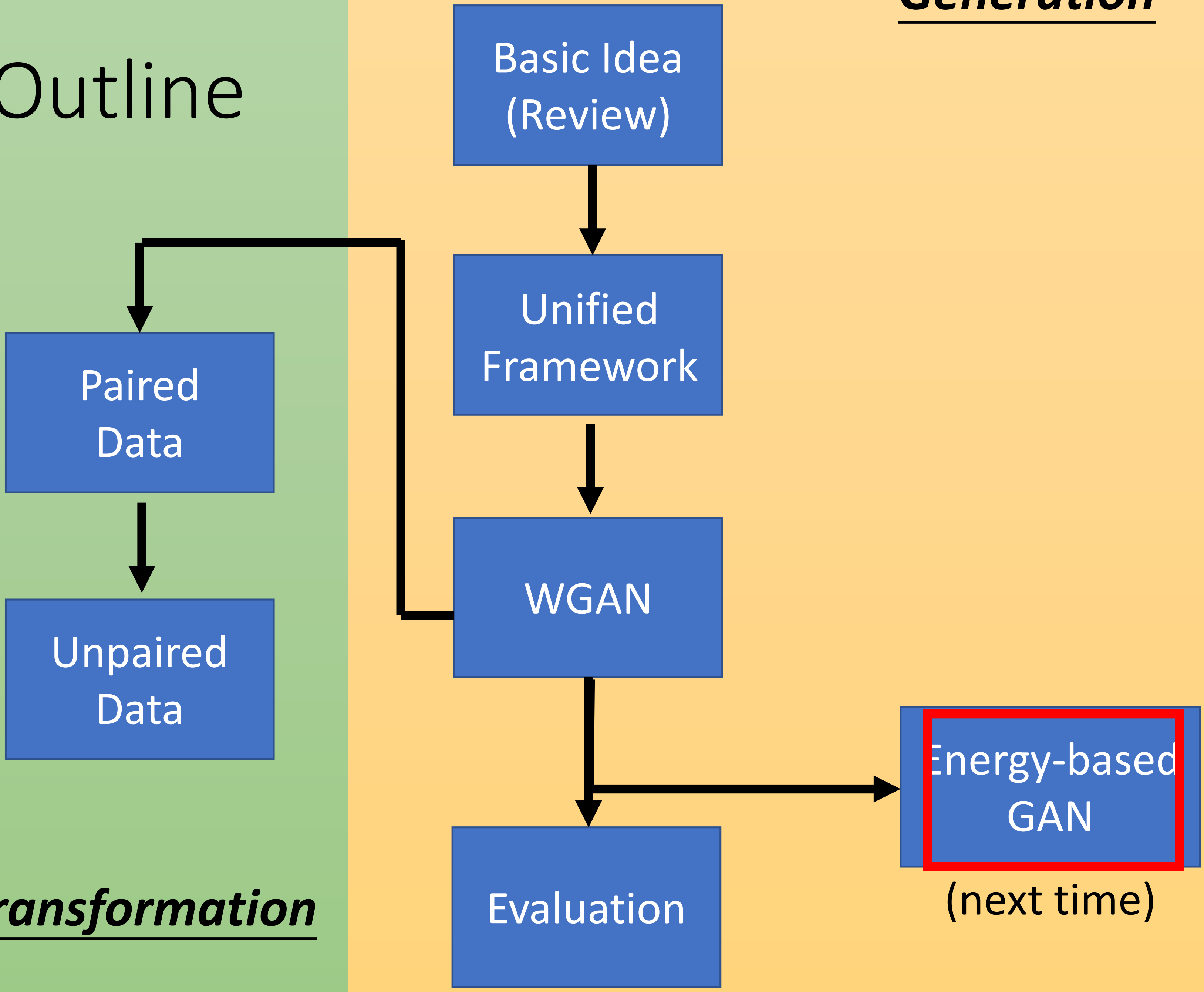
把真人頭像變成動漫人物

Unpaired Transformation – 真人動畫化

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



Outline



Generation

Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

Transformation

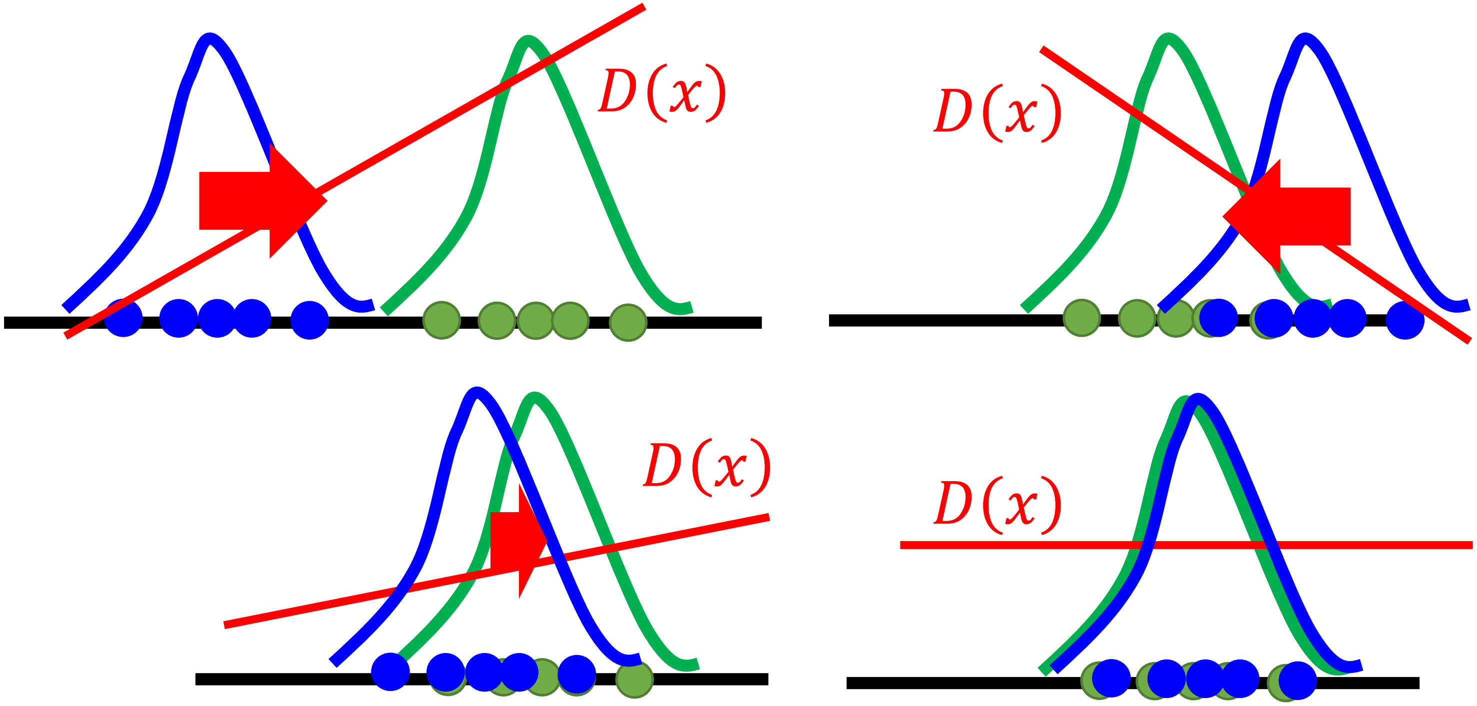
Review

🟥 Discriminator

🟩 Data (target) distribution

🟦 Generated distribution

- Role of discriminator: lead the generator

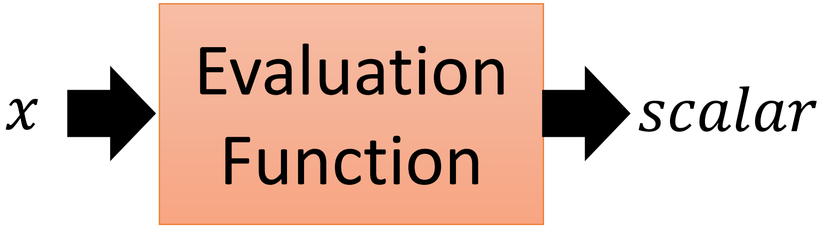


When the data distribution and generated distribution is the same

The discriminator will be useless (flat).

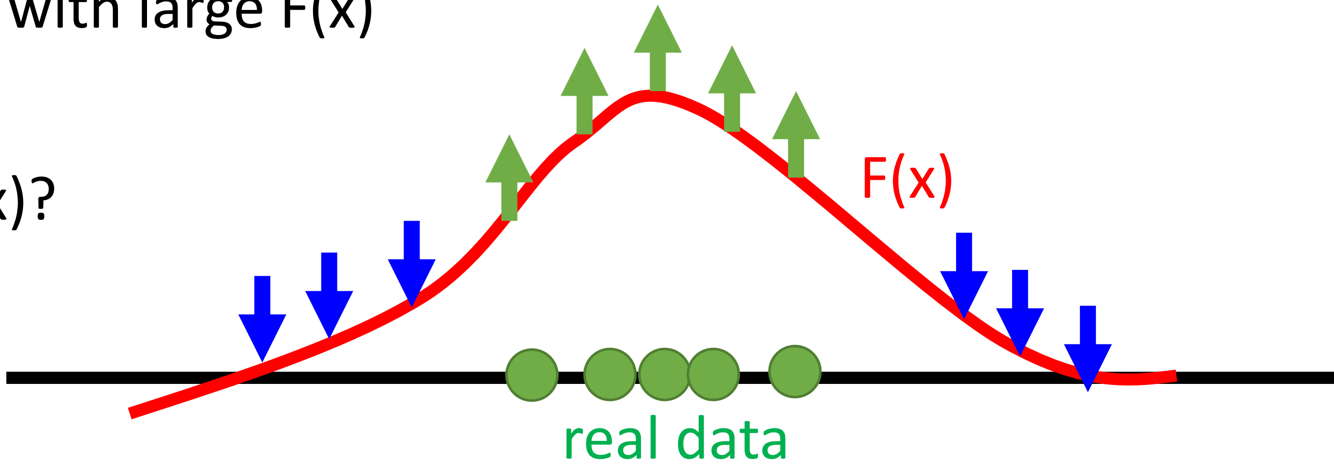
Energy-based Model

- 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



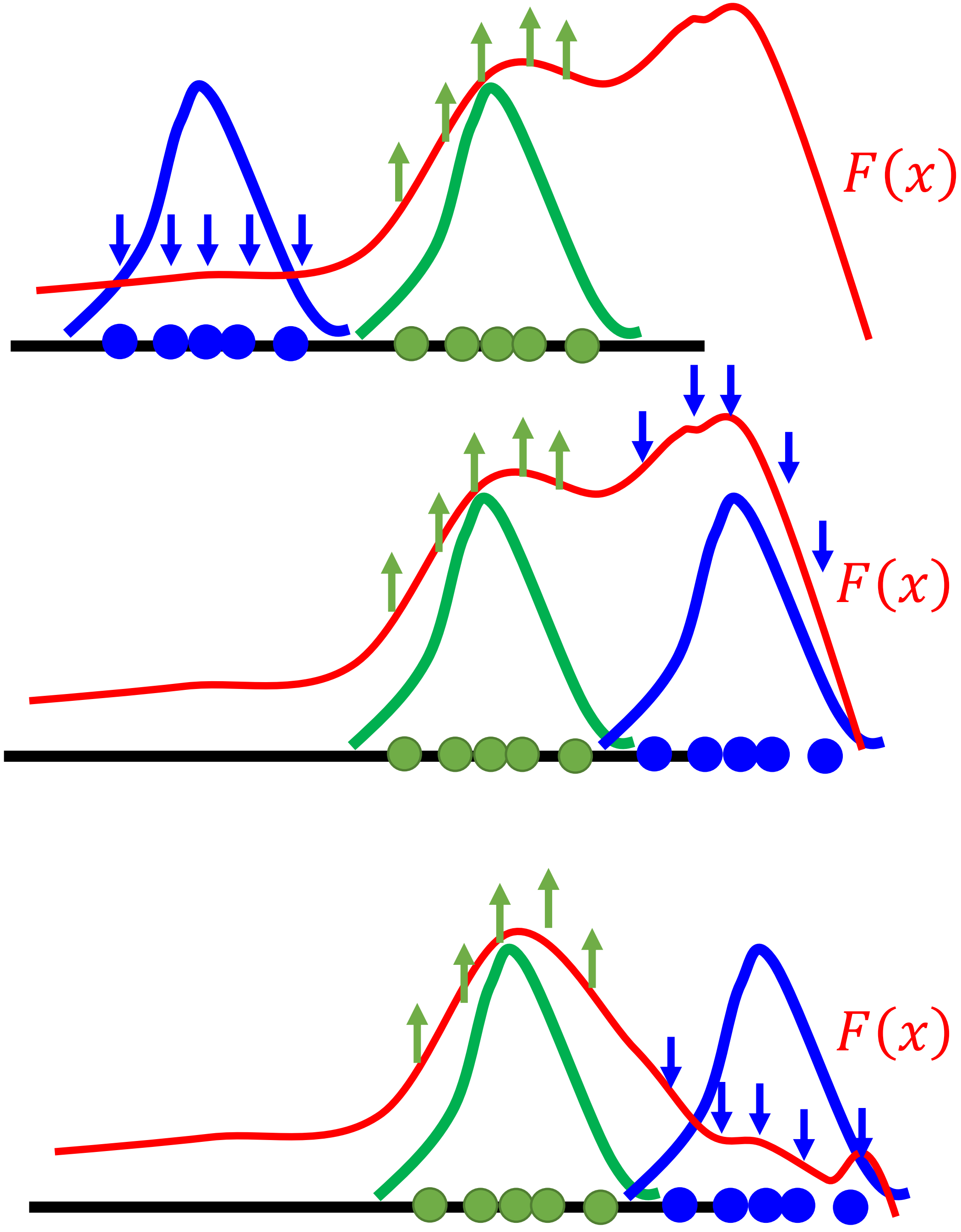
x → Evaluation Function → $scalar$

- We can find good x by F(x):
  - Generate x with large F(x)
- How to find F(x)?

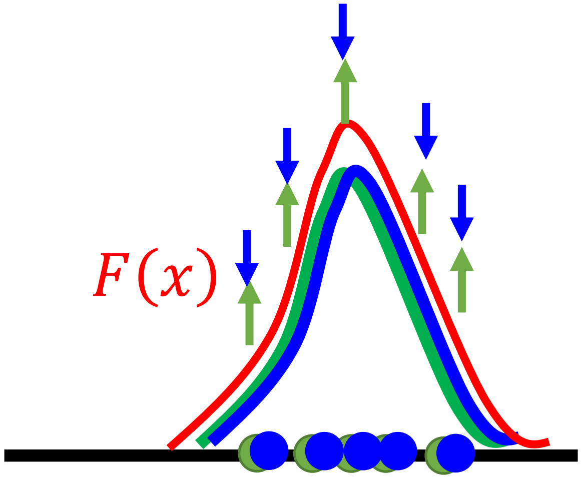


Energy-based GAN

- 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