

MAE 688: Machine Learning for Mechanical Engineers

Lecture 7- Generative Adversarial Network (GAN) Part 1



Dr. Bing Dong

Director, Built Environment Science and Technology (BEST) Lab

Professor

Mechanical and Aerospace Engineering

Syracuse University

Generative Adversarial Network



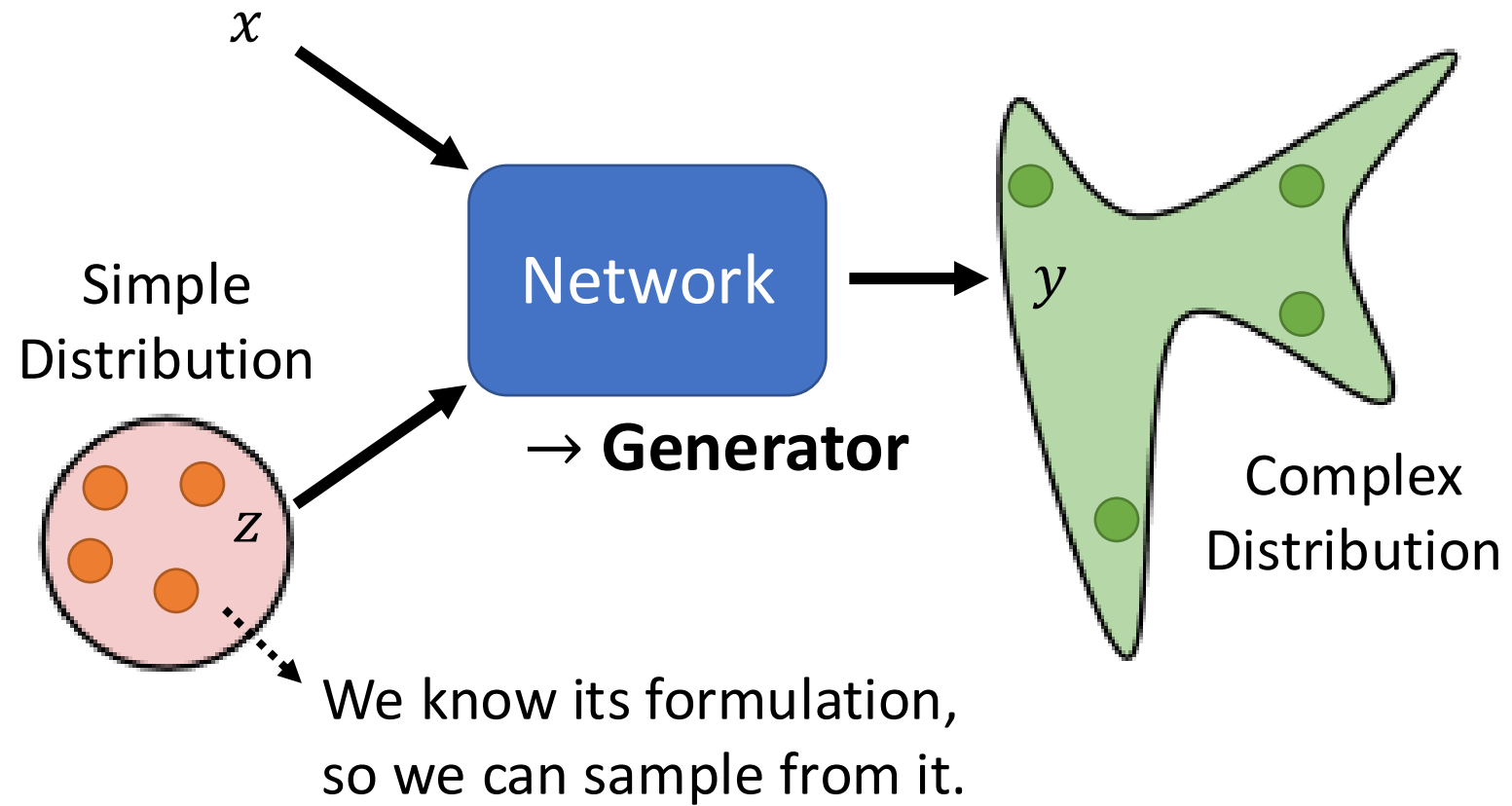
Introduction to Deep Learning
Deep Learning-Training (1)
Recurrent Neural Network
LSTM
Transformer

Generative Adversarial Network

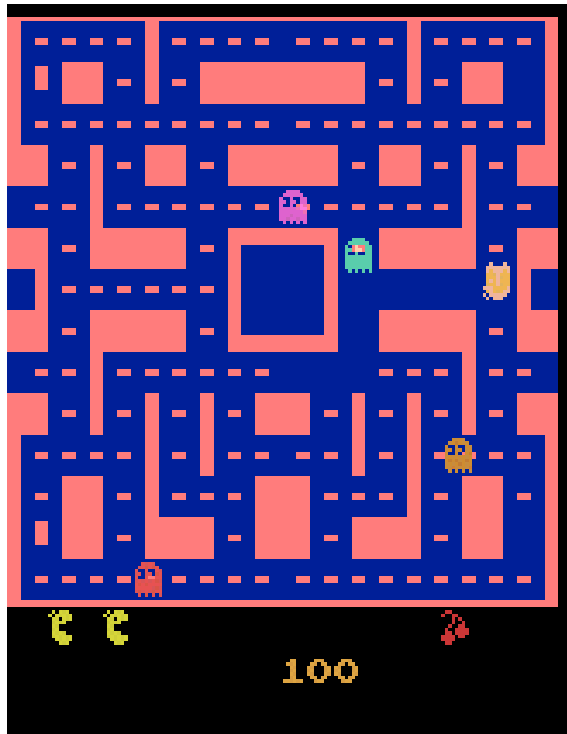


In mechanical engineering, sometime the measurement is very expensive, and you want to have more data...

Network as Generator

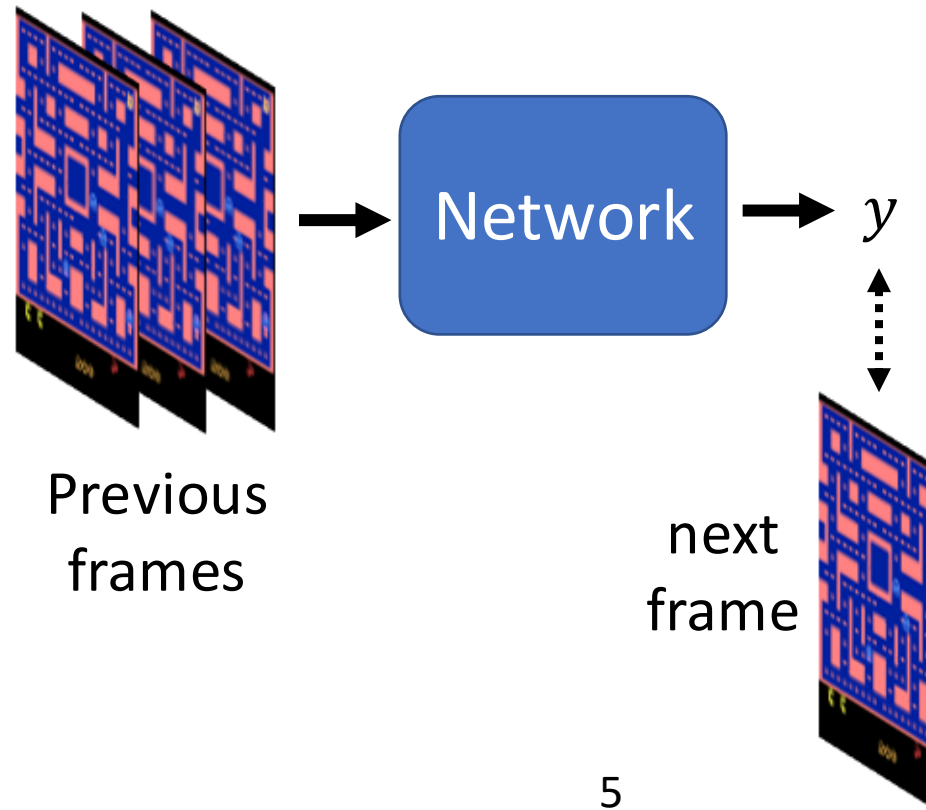


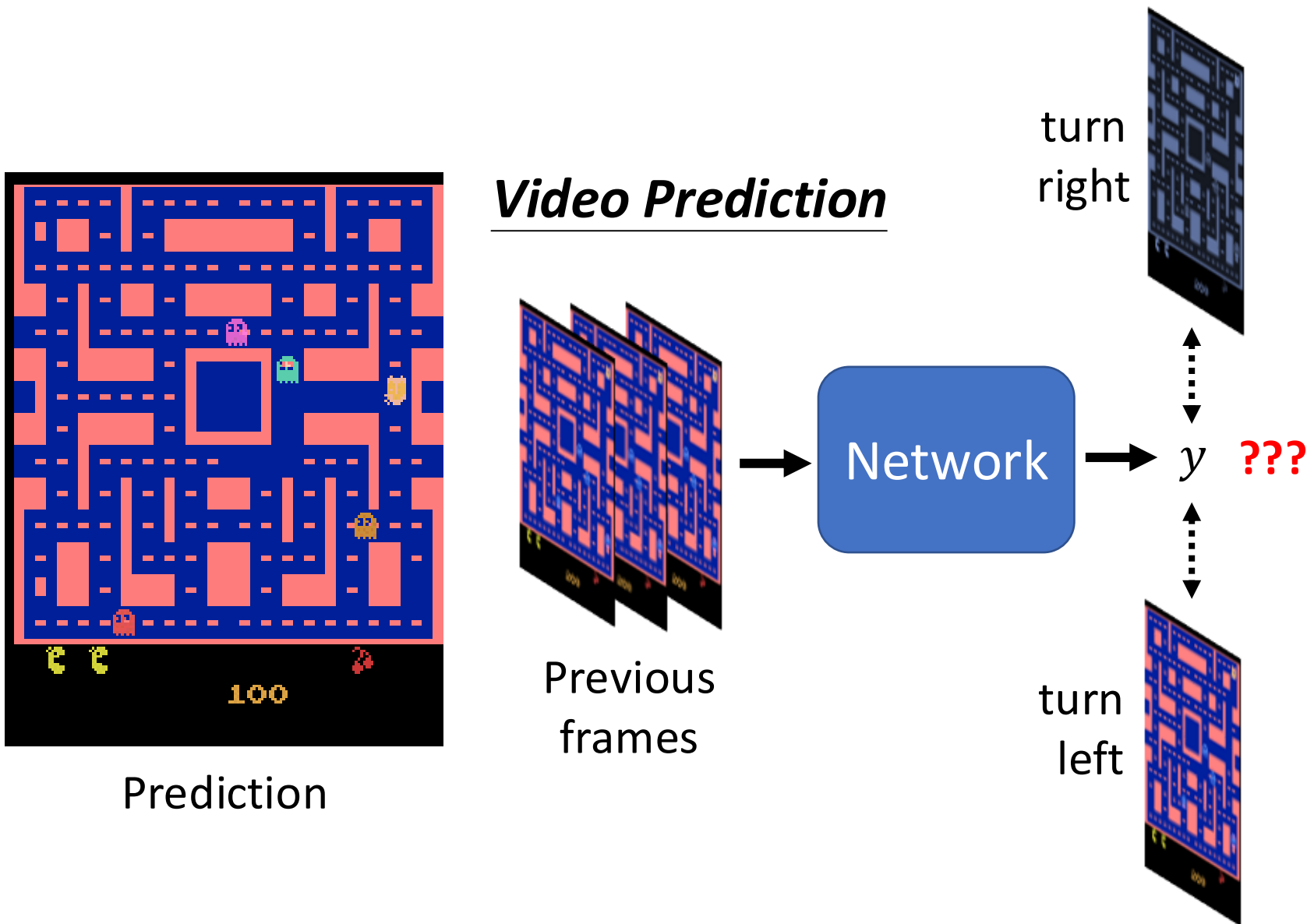
Why distribution?

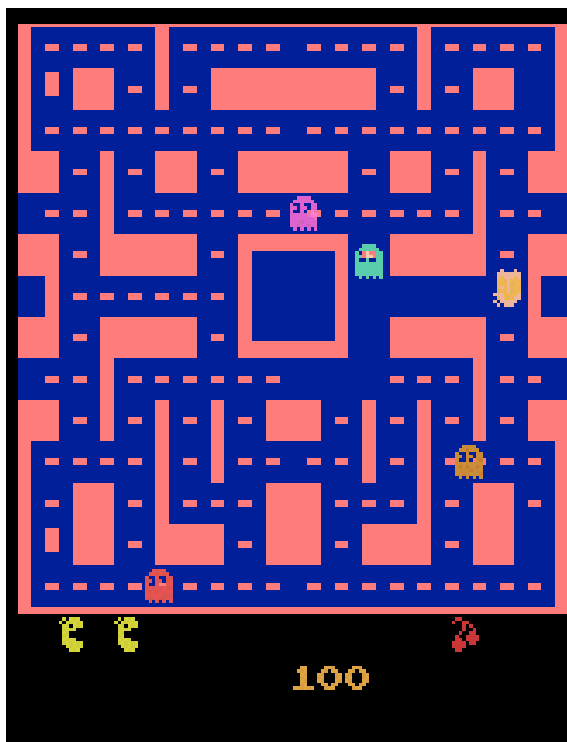


Real Video

Video Prediction

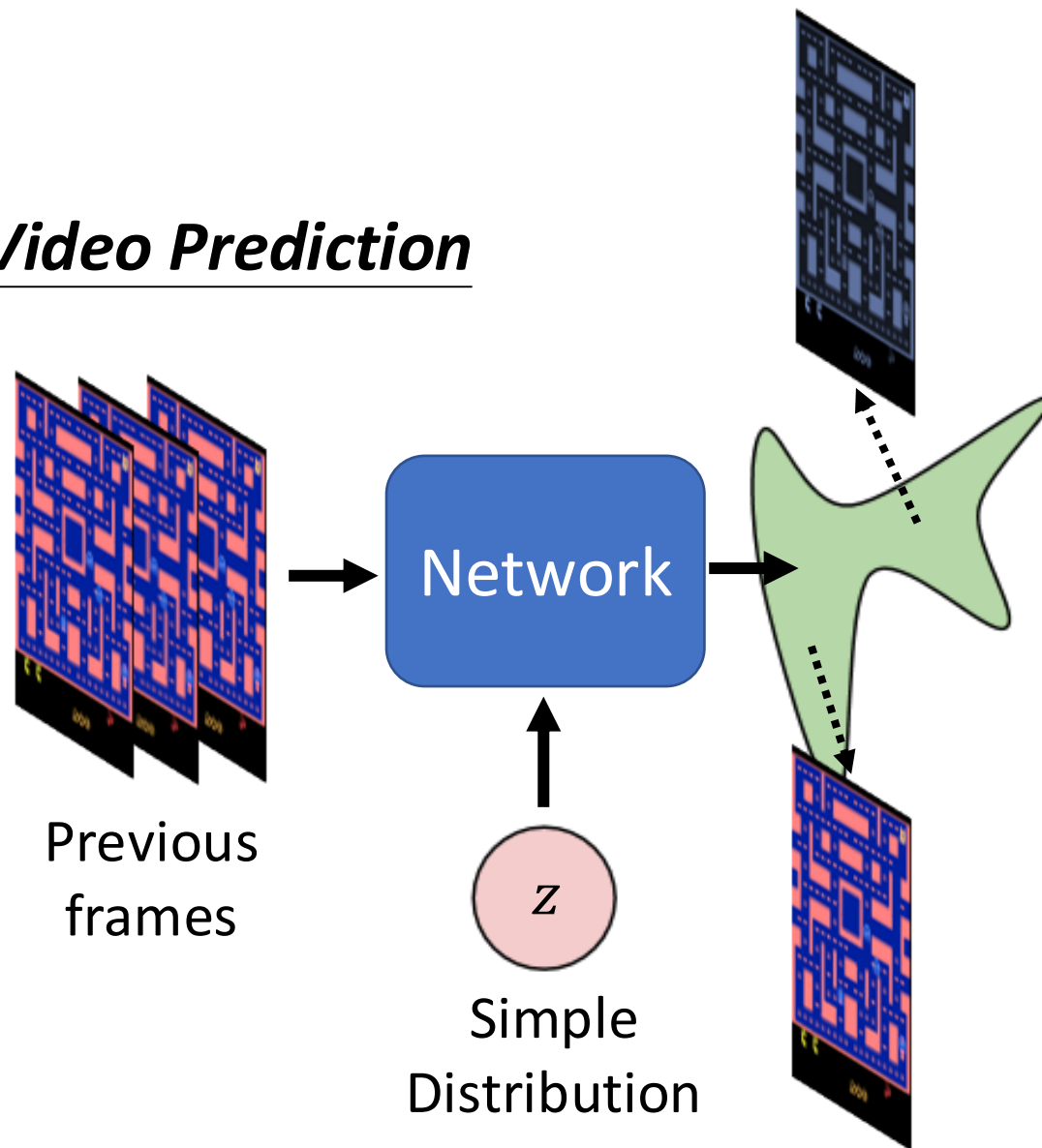






Prediction

Video Prediction



Why Distribution

(The same input has different outputs.)

- Especially for the tasks needs “*creativity*”

Drawing

Character
with red eyes



Chatbot

Who is Bing
Dong



A professor at SU

A professor at COE

Generative Adversarial Network (GAN)

GAN

How to pronounce “GAN”?



Google Says

All Kinds of GAN ...

<https://github.com/hindupuravinash/the-gan-zoo>

GAN

ACGAN

BGAN

CGAN

DCGAN

EBGAN

fGAN

GoGAN

⋮

- SeUDA - [Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation in Chest X-ray Segmentation](#)
- SG-GAN - [Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption \(github\)](#)
- SG-GAN - [Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attribute Manipulation](#)
- SGAN - [Texture Synthesis with Spatial Generative Adversarial Networks](#)
- SGAN - [Stacked Generative Adversarial Networks \(github\)](#)
- SGAN - [Steganographic Generative Adversarial Networks](#)
- SGAN - [SGAN: An Alternative Training of Generative Adversarial Networks](#)
- SGAN - [CT Image Enhancement Using Stacked Generative Adversarial Networks and Transfer Learning for Lesion Segmentation Improvement](#)
- sGAN - [Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast MRI](#)
- SiftingGAN - [SiftingGAN: Generating and Sifting Labeled Samples to Improve the Remote Sensing Image Scene Classification Baseline in vitro](#)
- SiGAN - [SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face Hallucination](#)
- SimGAN - [Learning from Simulated and Unsupervised Images through Adversarial Training](#)
- SisGAN - [Semantic Image Synthesis via Adversarial Learning](#)

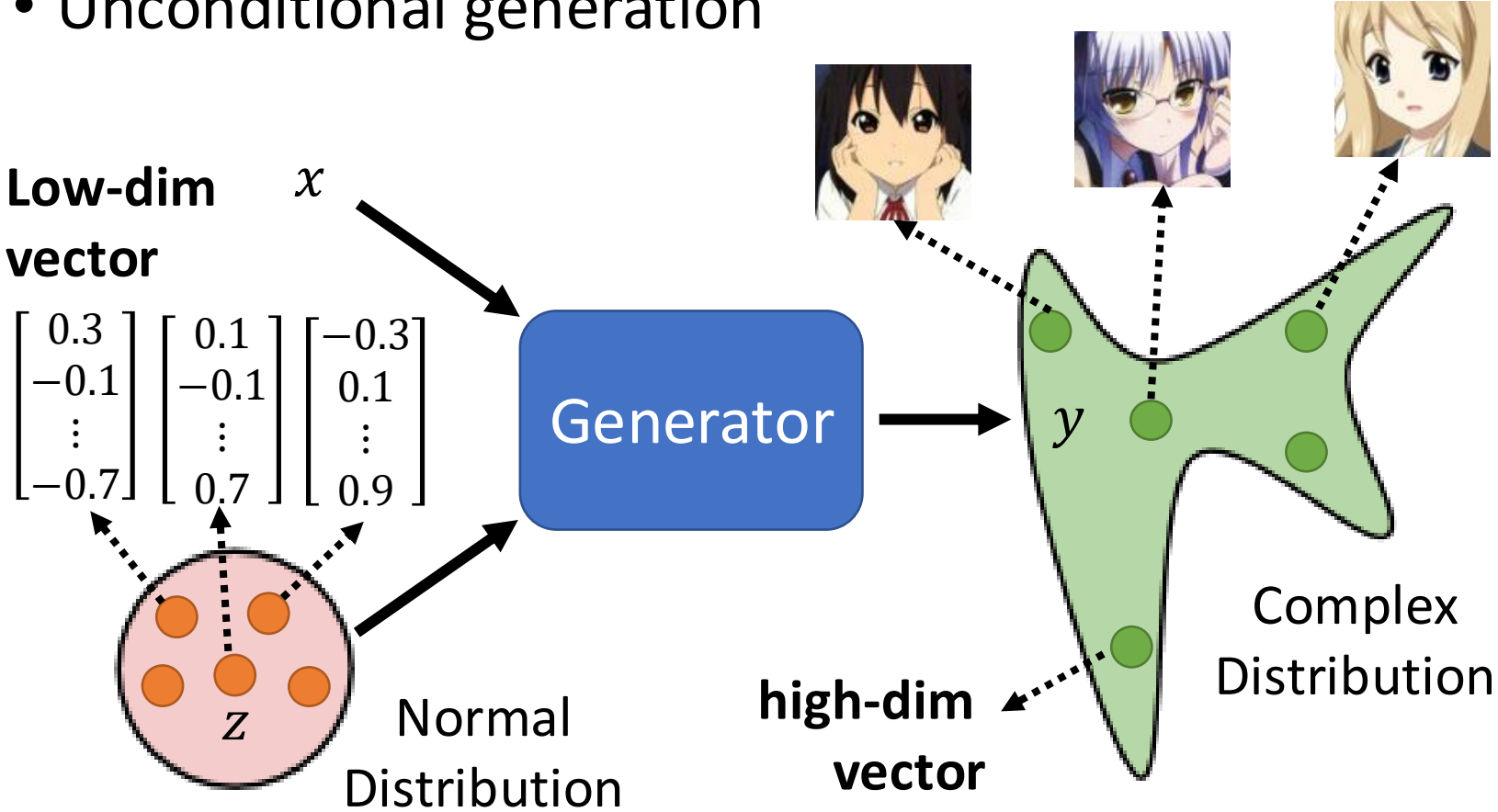
Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, "Variational Approaches for Auto-Encoding Generative Adversarial Networks", arXiv, 2017

²We use the Greek α prefix for α -GAN, as AEGAN and most other Latin prefixes seem to have been taken <https://deephunt.in/the-gan-zoo-79597dc8c347>.



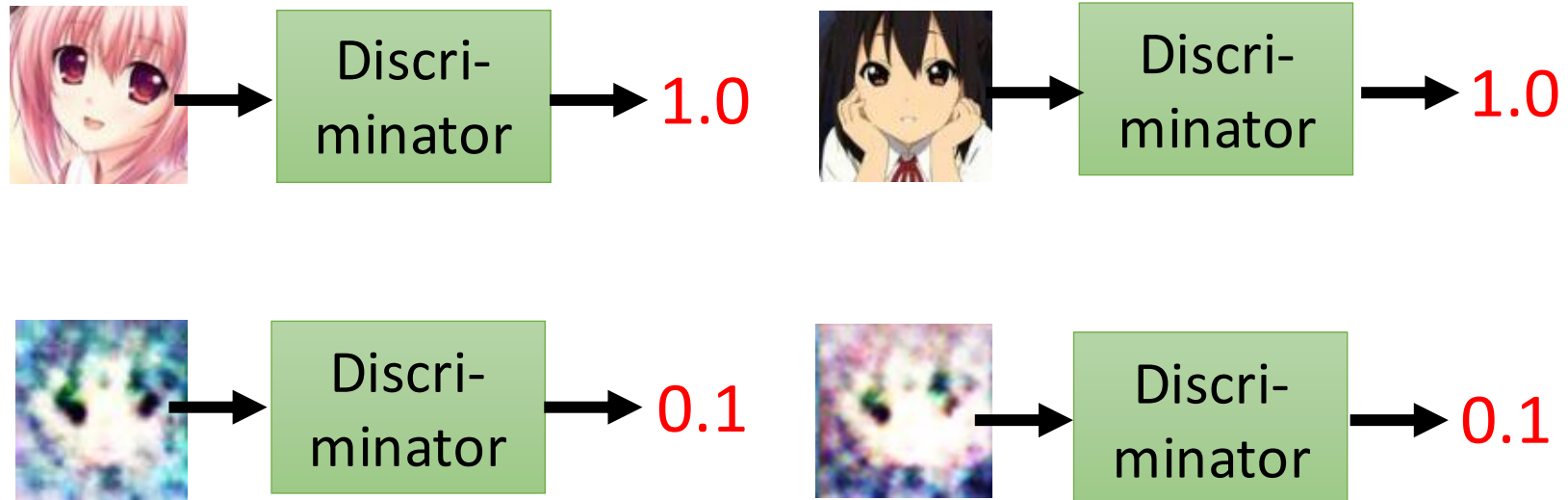
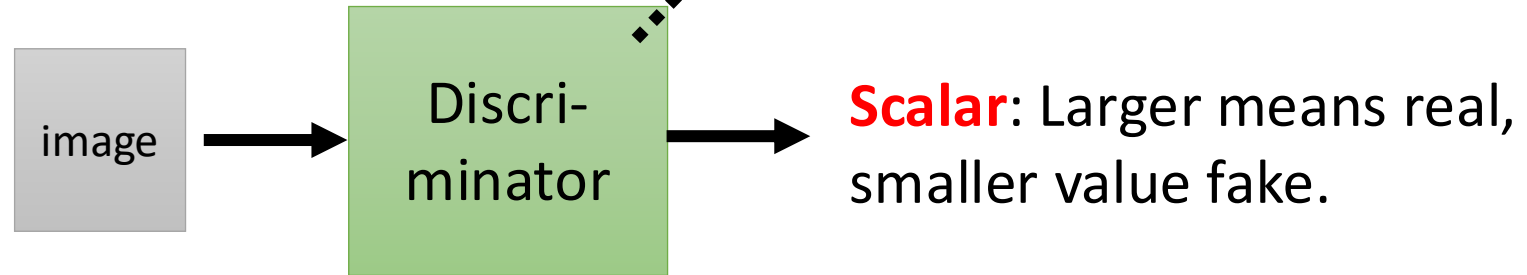
Anime Face Generation

- Unconditional generation



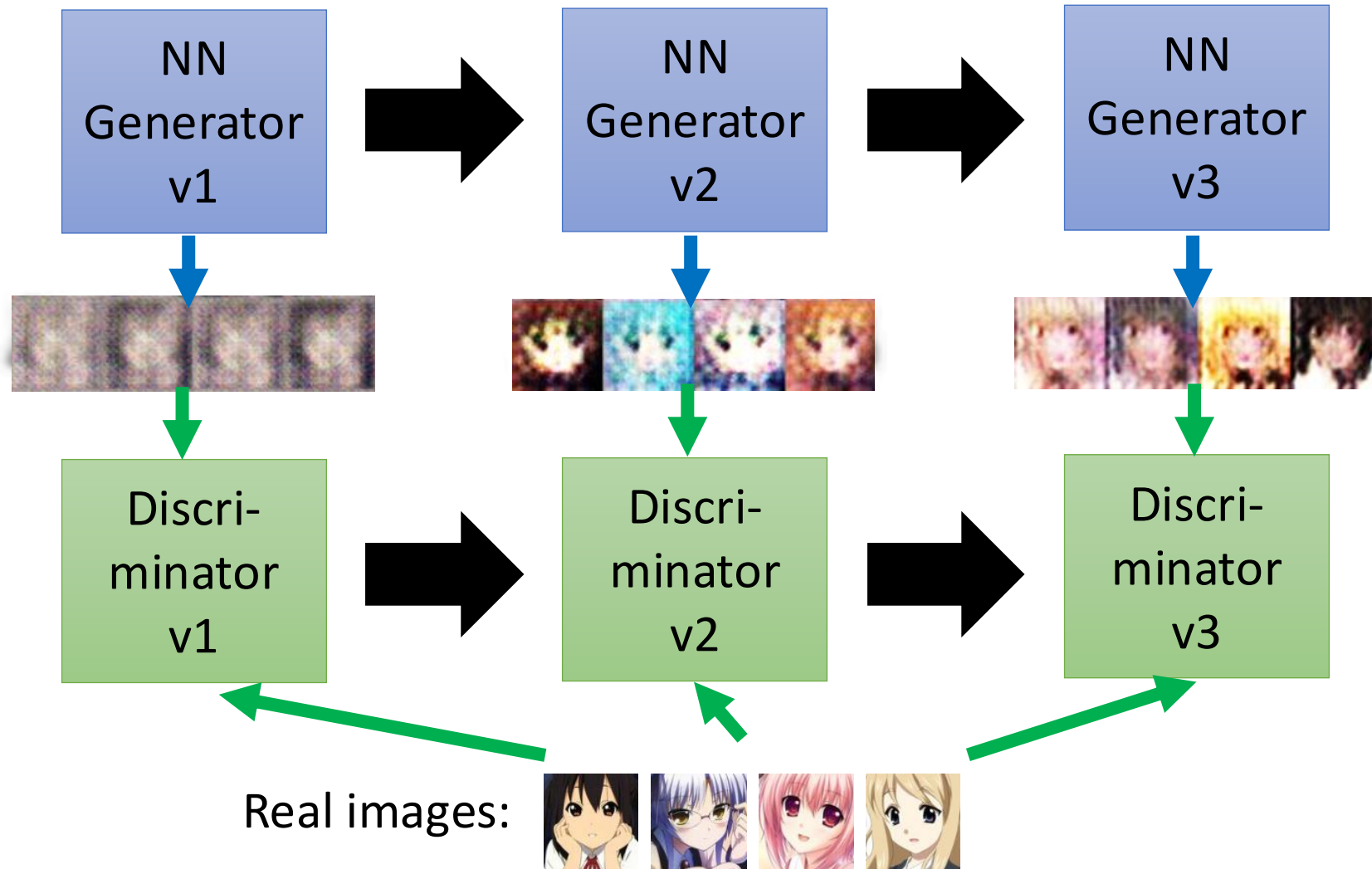
Discriminator

It is a neural network
(that is, a function).



Basic Idea of GAN

This is where the term “*adversarial*” comes from.

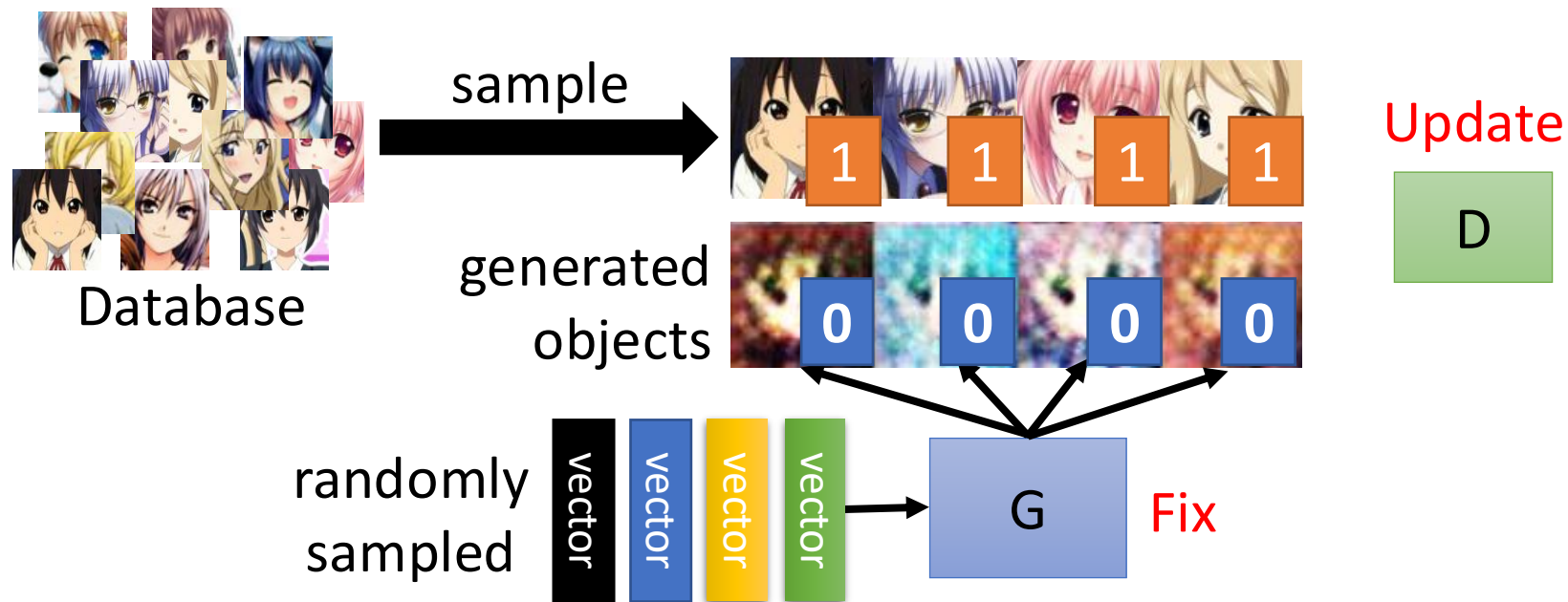


Algorithm

- Initialize generator and discriminator
- In each training iteration:



Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.

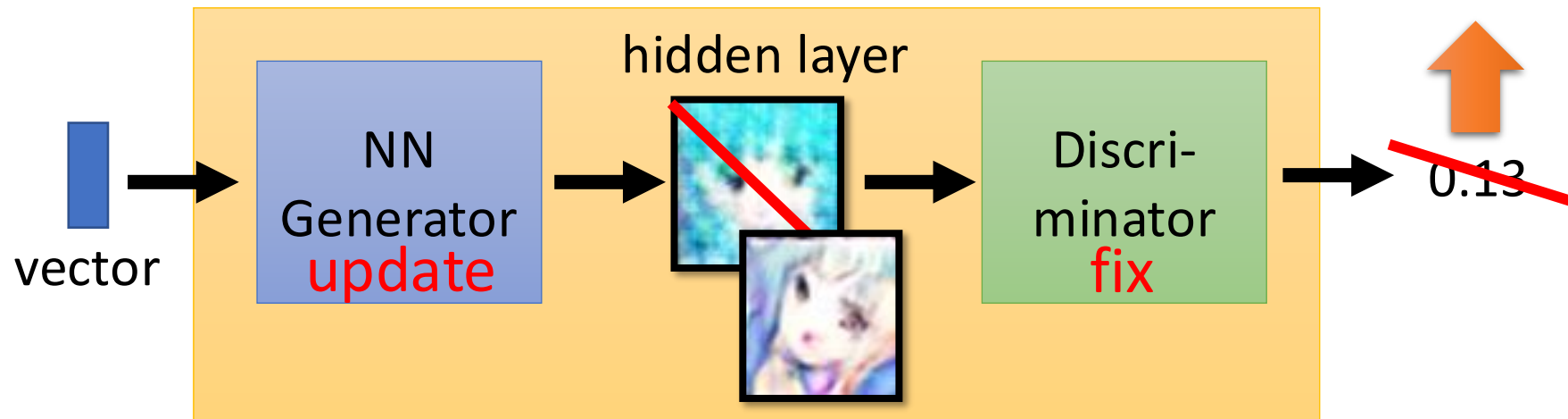
Algorithm

- Initialize generator and discriminator
- In each training iteration:



Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



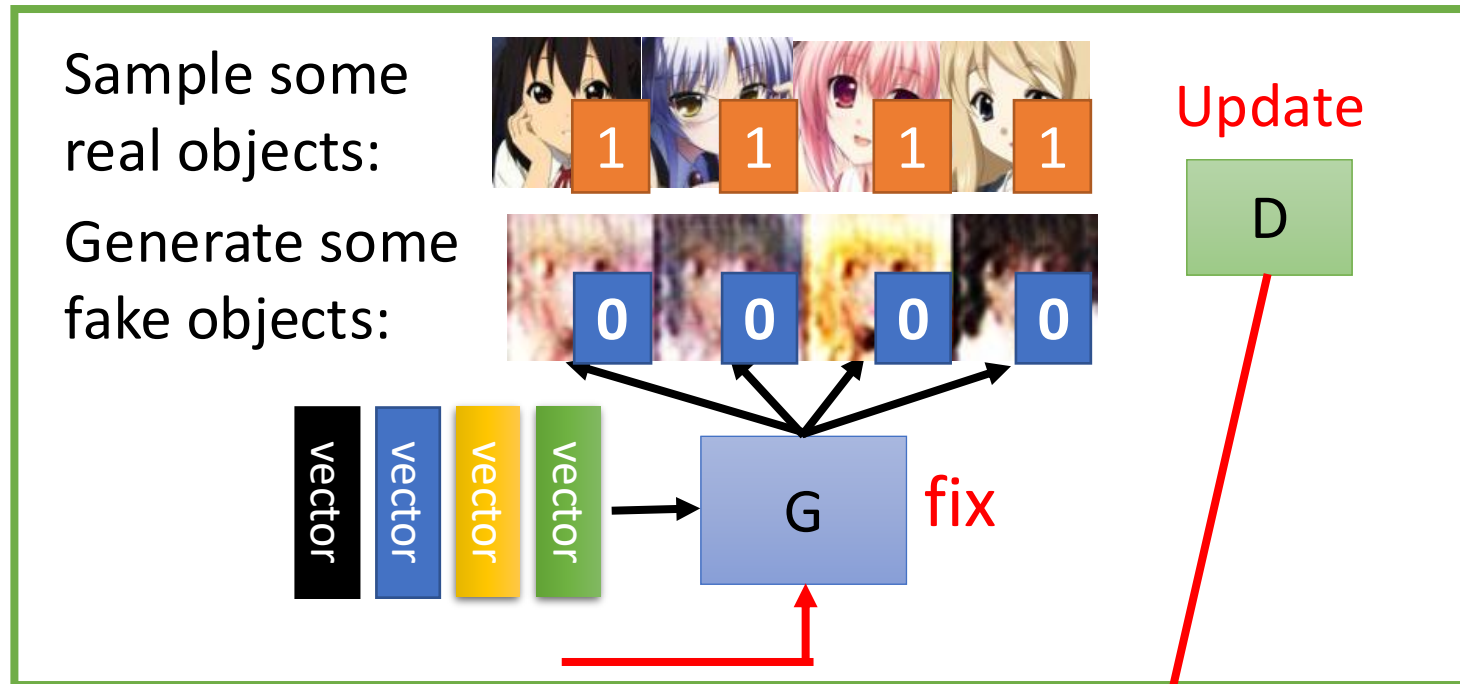
large network

Algorithm

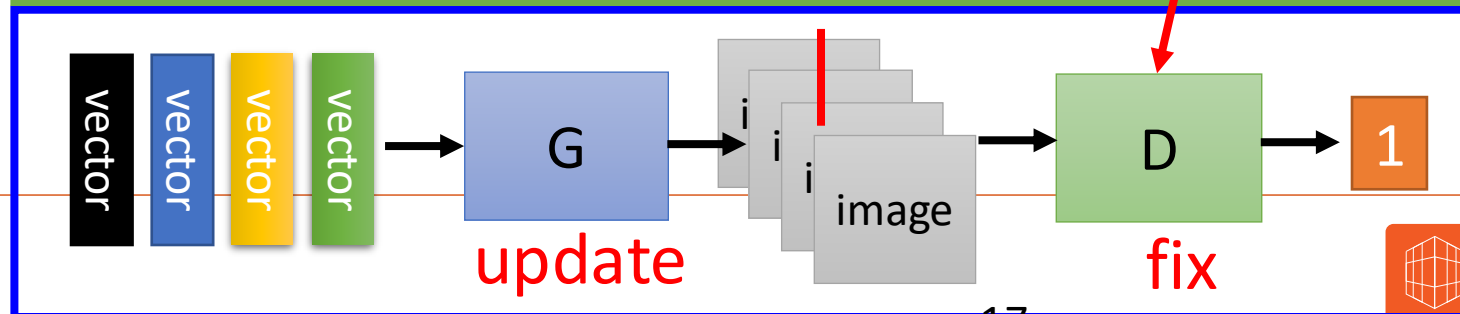
- Initialize generator and discriminator
- In each training iteration:



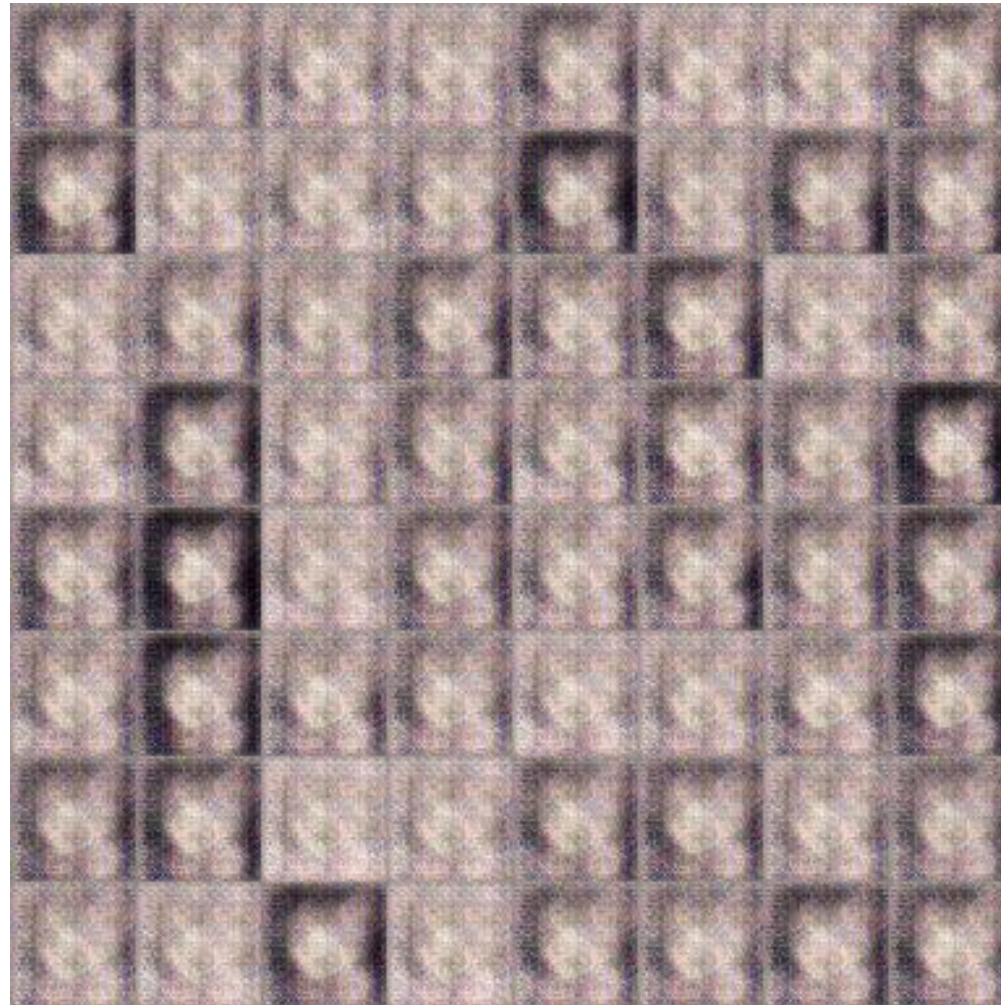
Learning
D



Learning
G



Anime Face Generation



100 updates



1000 updates

2000 updates





5000 updates

20,000 updates



50,000 updates





The faces
generated by
machine.

In 2019, with StyleGAN

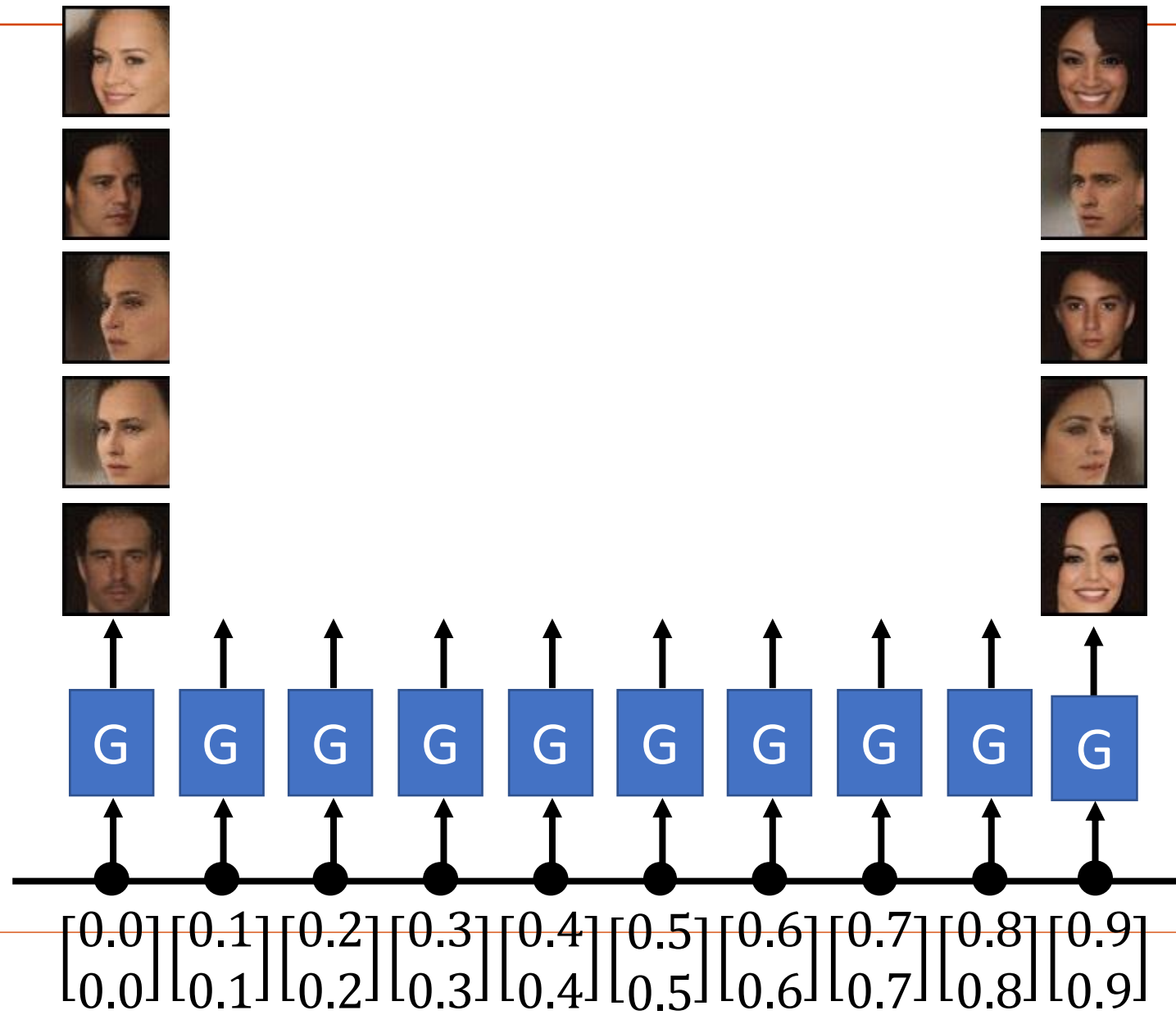


Source of video:
<https://www.gwern.net/Faces>

Progressive GAN



<https://arxiv.org/abs/1710.10196>



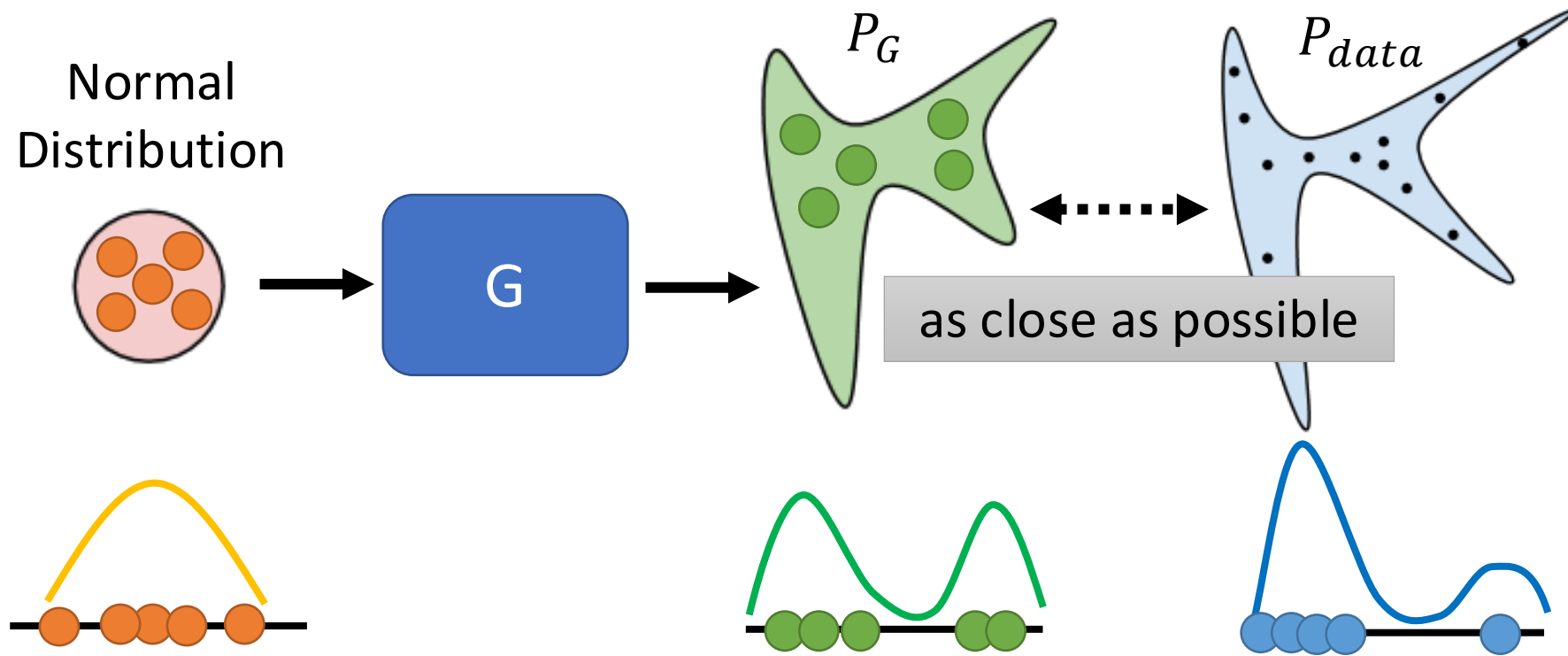
Today BigGAN



Theory behind GAN

Our Objective

$$\text{c.f. } w^*, b^* = \arg \min_{w, b} L$$



$$G^* = \arg \min_G \underline{Div}(P_G, P_{data})$$

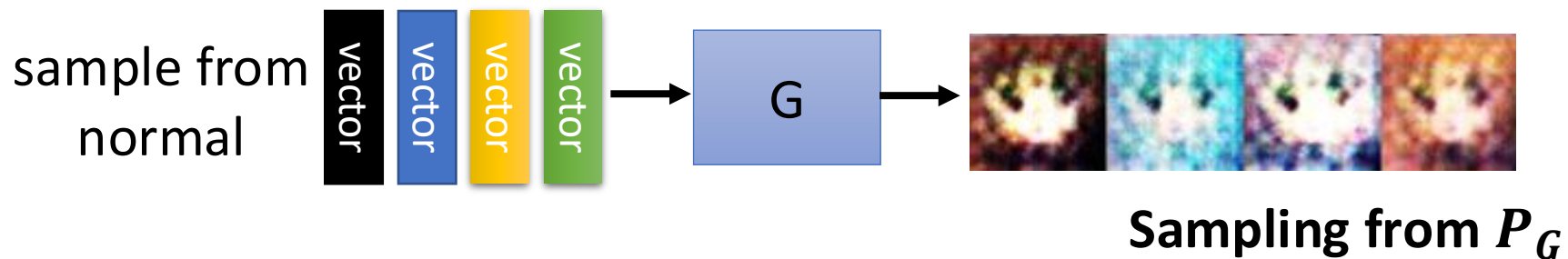
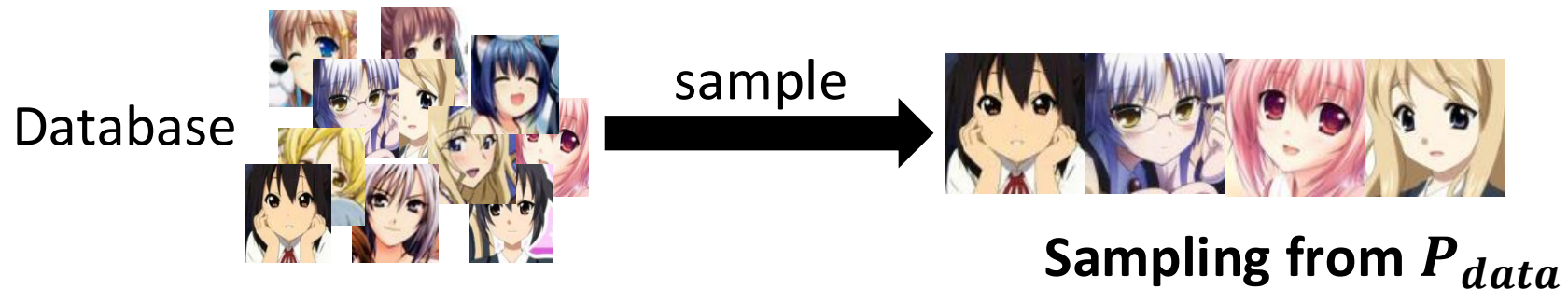
Divergence between distributions P_G and P_{data}

How to compute the divergence?

Sampling is good enough

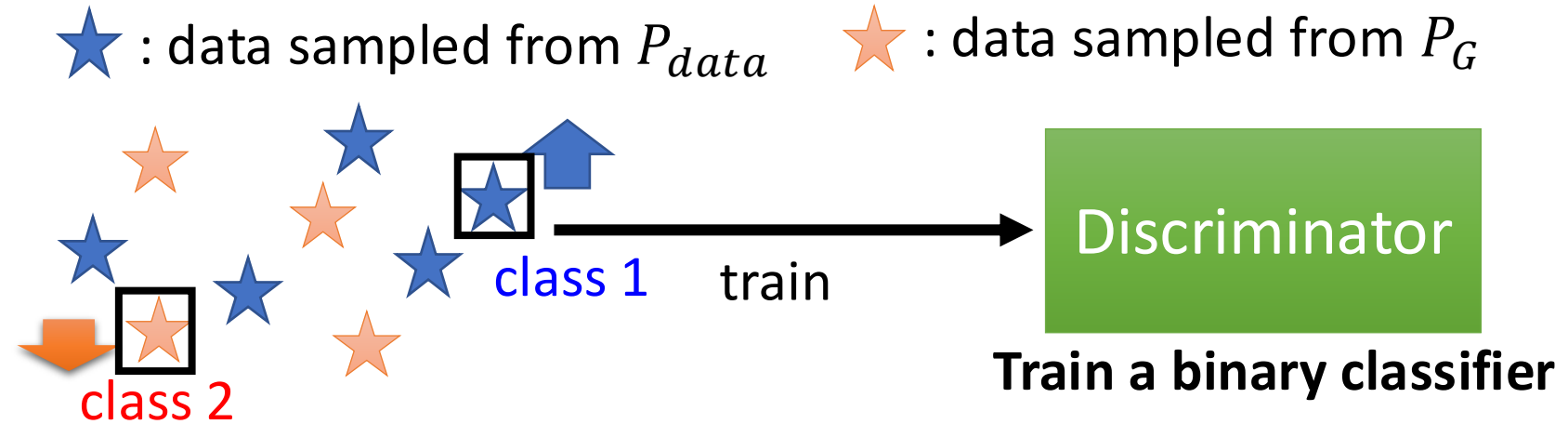
$$G^* = \arg \min_G \text{Div}(P_G, P_{data})$$

Although we do not know the distributions of P_G and P_{data} , we can sample from them.



Discriminator

$$G^* = \arg \min_G \text{Div}(P_G, P_{data})$$



Training: $D^* = \arg \max_D V(D, G)$ The value is related to Jensen-Shannon (JS) divergence.

Objective Function for D

$$V(G, D) = E_{y \sim P_{data}} [\log D(y)] + E_{y \sim P_G} [\log(1 - D(y))]$$

$D^* = \arg \max_D V(D, G)$ = Training classifier:
 negative cross entropy minimize cross entropy

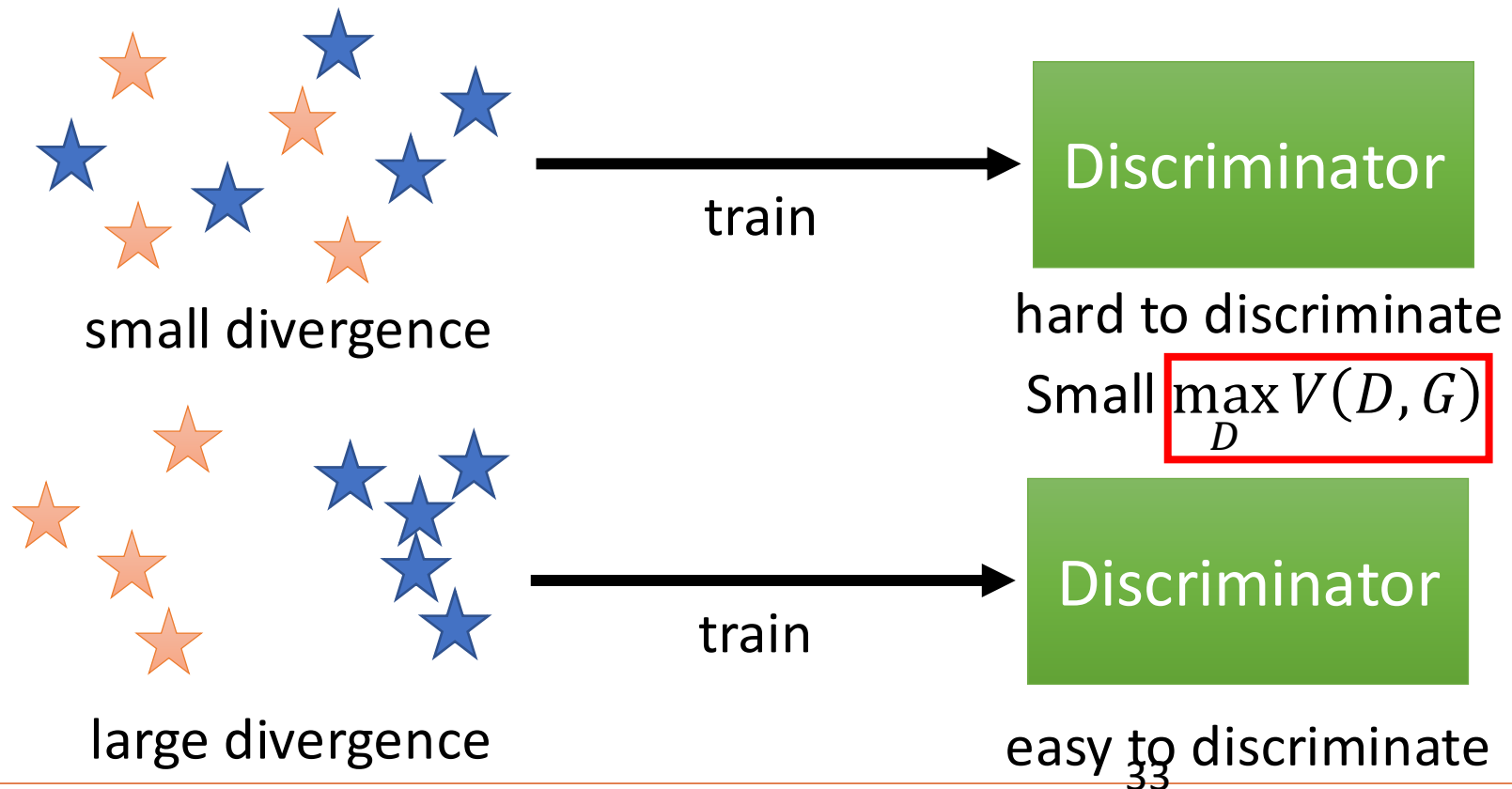
Discriminator

$$G^* = \arg \min_G \text{Div}(P_G, P_{data})$$

- ★ : data sampled from P_{data}
- ★ : data sampled from P_G

Training:

$$D^* = \arg \max_D V(D, G)$$



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

$$D^* = \arg \max_D V(D, G)$$

The maximum objective value is related to JS divergence.

Step 1: Fix generator G , and update discriminator D

Step 2: Fix discriminator D , and update generator G

Can we use other divergence?

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	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} dx - \log(4)$	$u \log u - (u + 1) \log(u + 1)$

Using the divergence
you like 😊

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

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	$-\log(1 - \exp(t))$

GAN is difficult to train



Tips for GAN

JS divergence is not suitable

- In most cases, P_G and P_{data} are not overlapped.

- 1. The nature of data

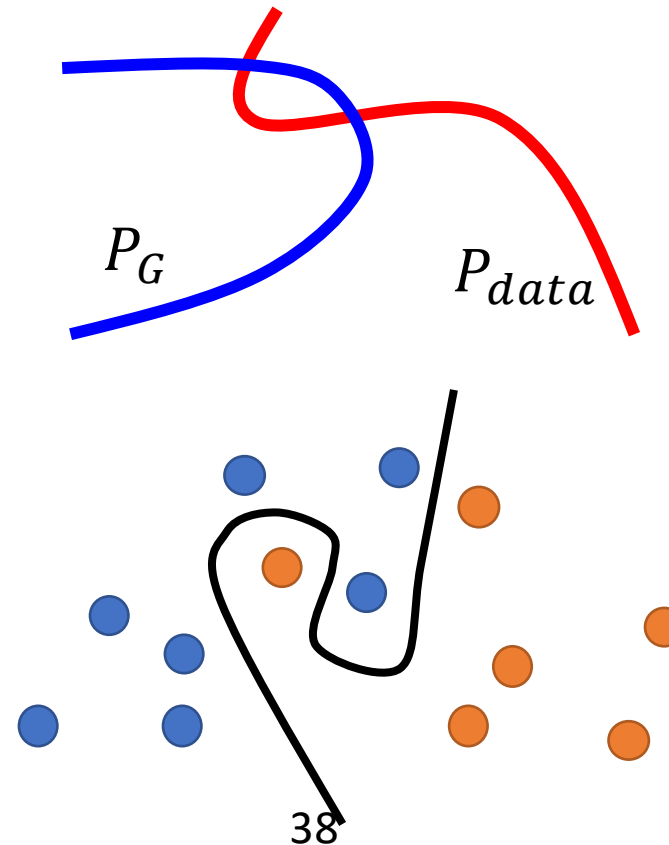
Both P_{data} and P_G are low-dim manifold in high-dim space.

The overlap can be ignored.

- 2. Sampling

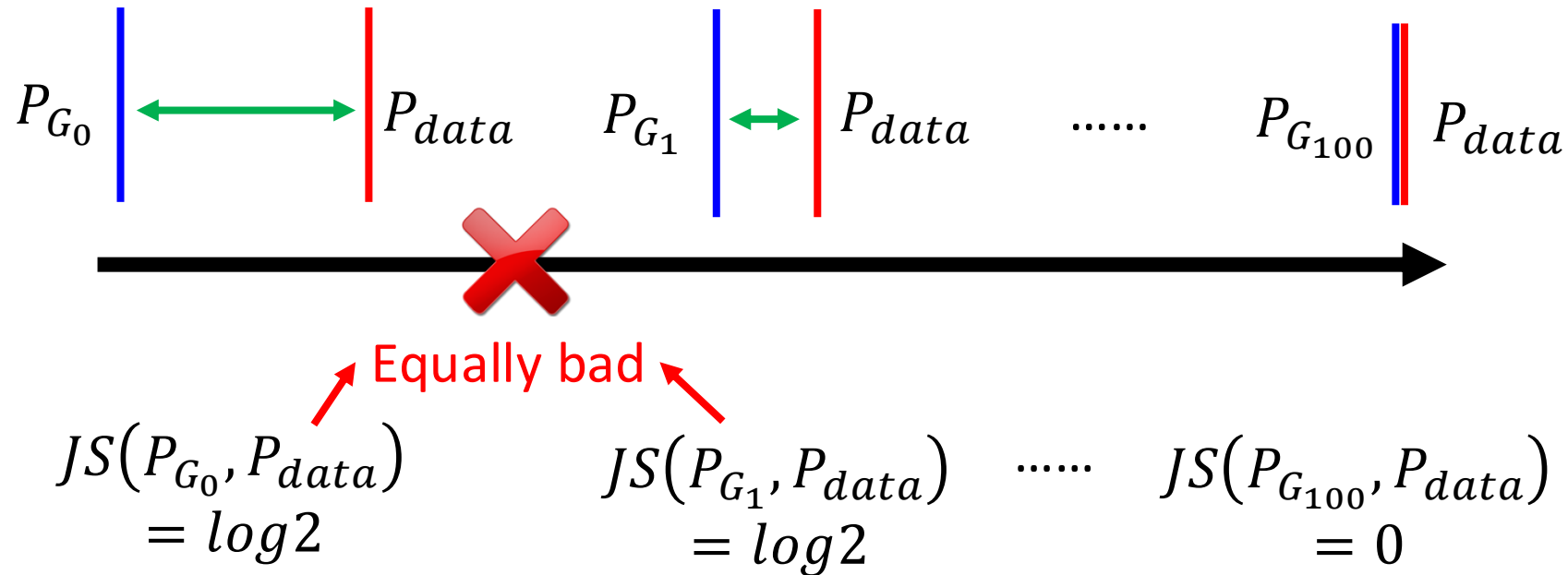
Even though P_{data} and P_G have overlap.

If you do not have enough sampling



What is the problem of JS divergence?

JS divergence is always $\log 2$ if two distributions do not overlap.

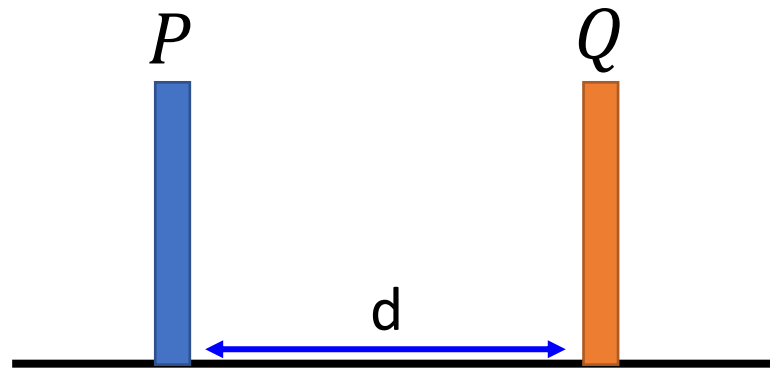


Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy.

The accuracy (or loss) means nothing during GAN training.

Wasserstein distance

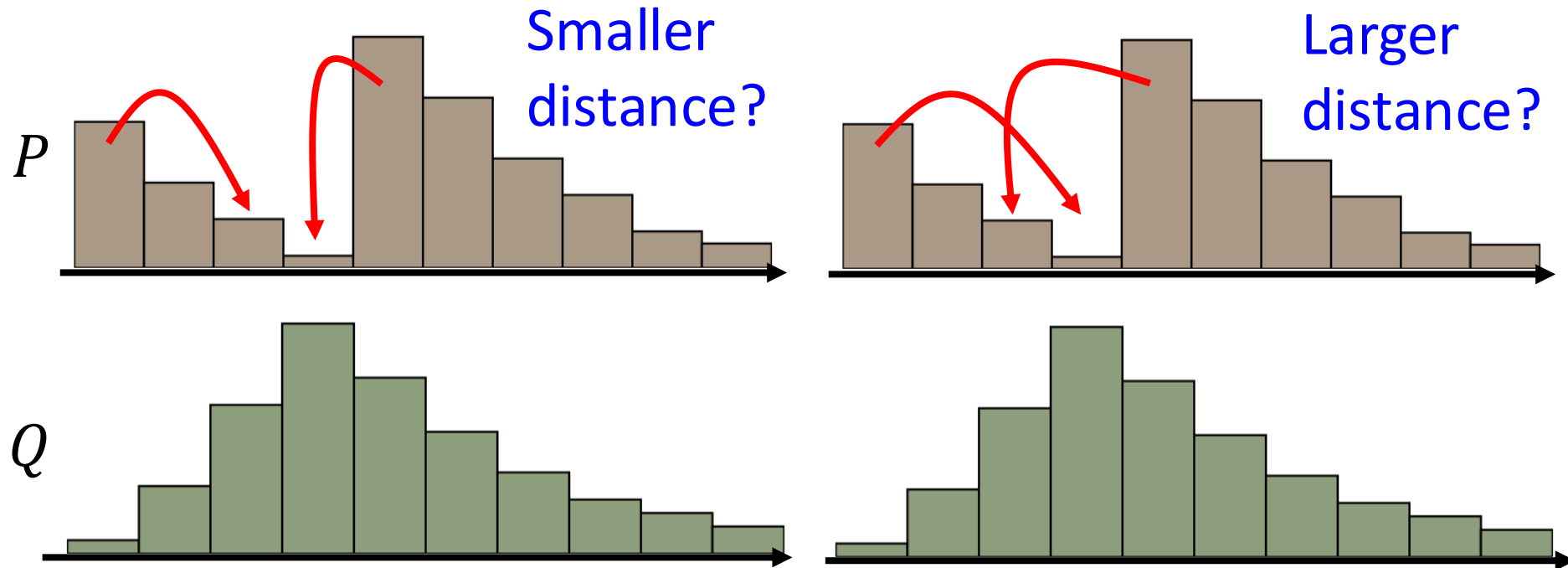
- Considering one distribution P as a pile of earth, and another distribution Q as the target
- The average distance the earth mover has to move the earth.



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



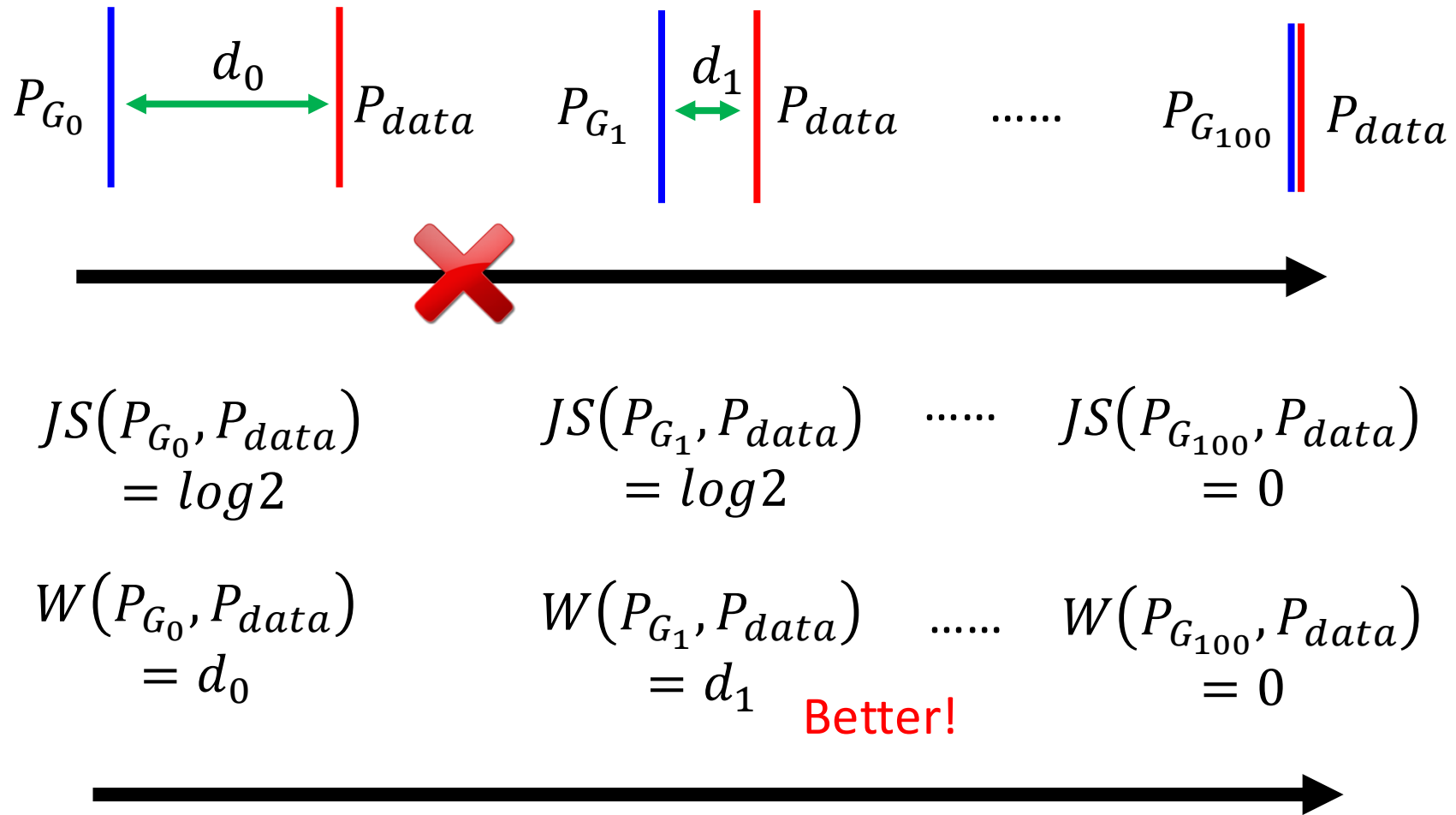
Wasserstein distance



There are many possible “moving plans”.

Using the “moving plan” with the smallest average distance to define the Wasserstein distance.

What is the problem of JS divergence?



Evaluate Wasserstein distance between P_{data} and P_G

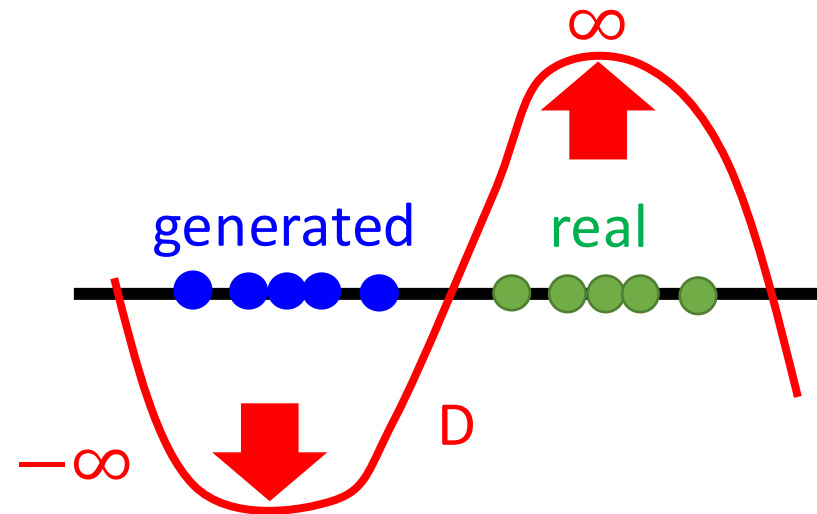
$$\max_{D \in \text{1-Lipschitz}} \{E_{y \sim P_{data}} [D(y)] - E_{y \sim P_G} [D(y)]\}$$

D has to be smooth enough.

How to fulfill this constraint?

Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(y) become ∞ and $-\infty$



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

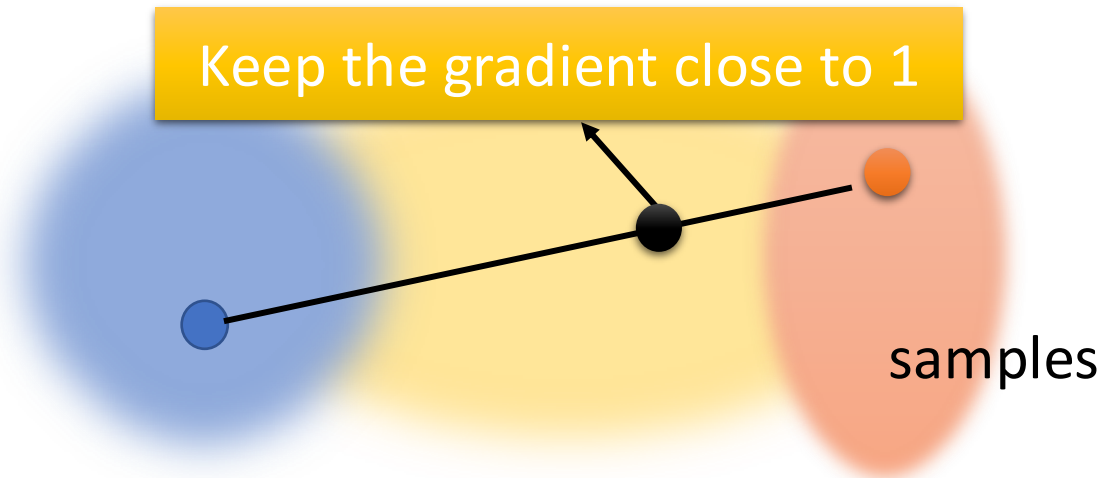
- Original WGAN → Weight

Force the parameters w between c and $-c$

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

- Improved WGAN → Gradient Penalty

Keep the gradient close to 1

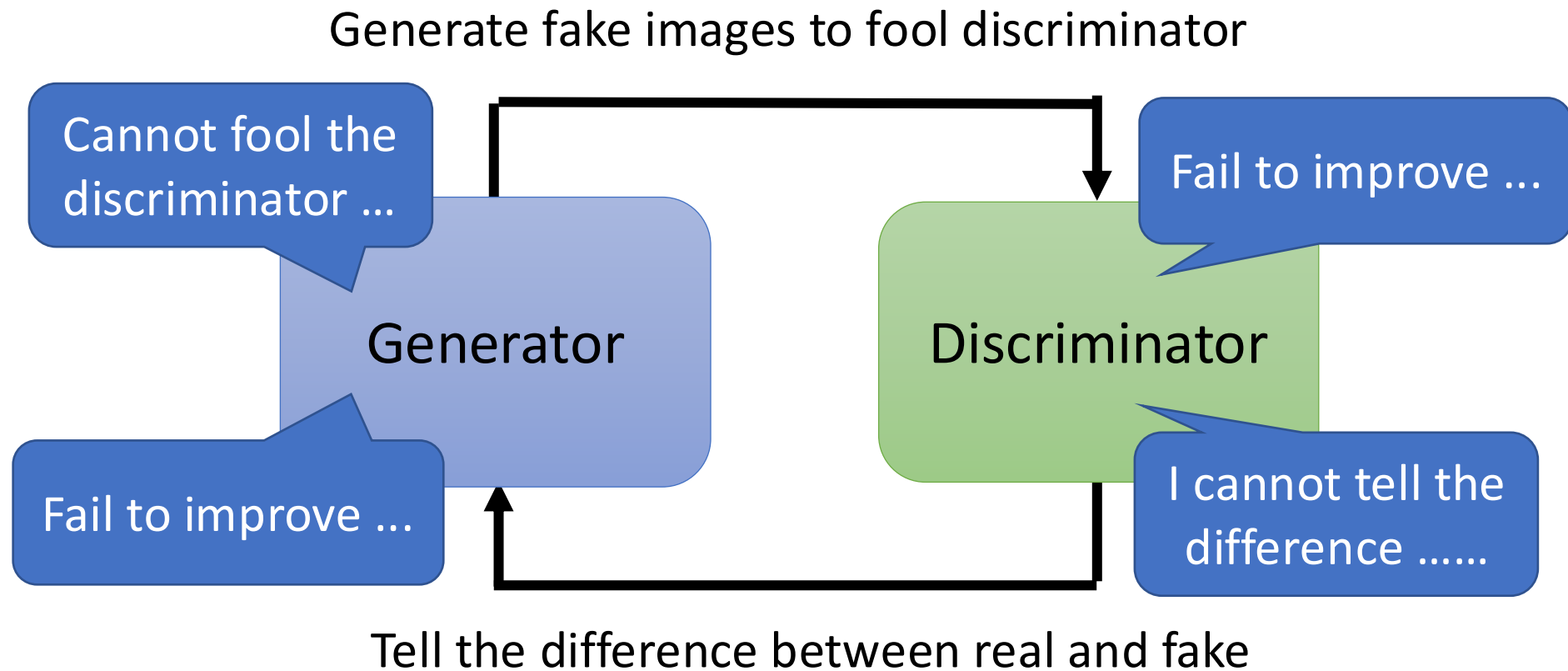


<https://arxiv.org/abs/1802.05957>

- Spectral Normalization → Keep gradient norm smaller than 1 everywhere

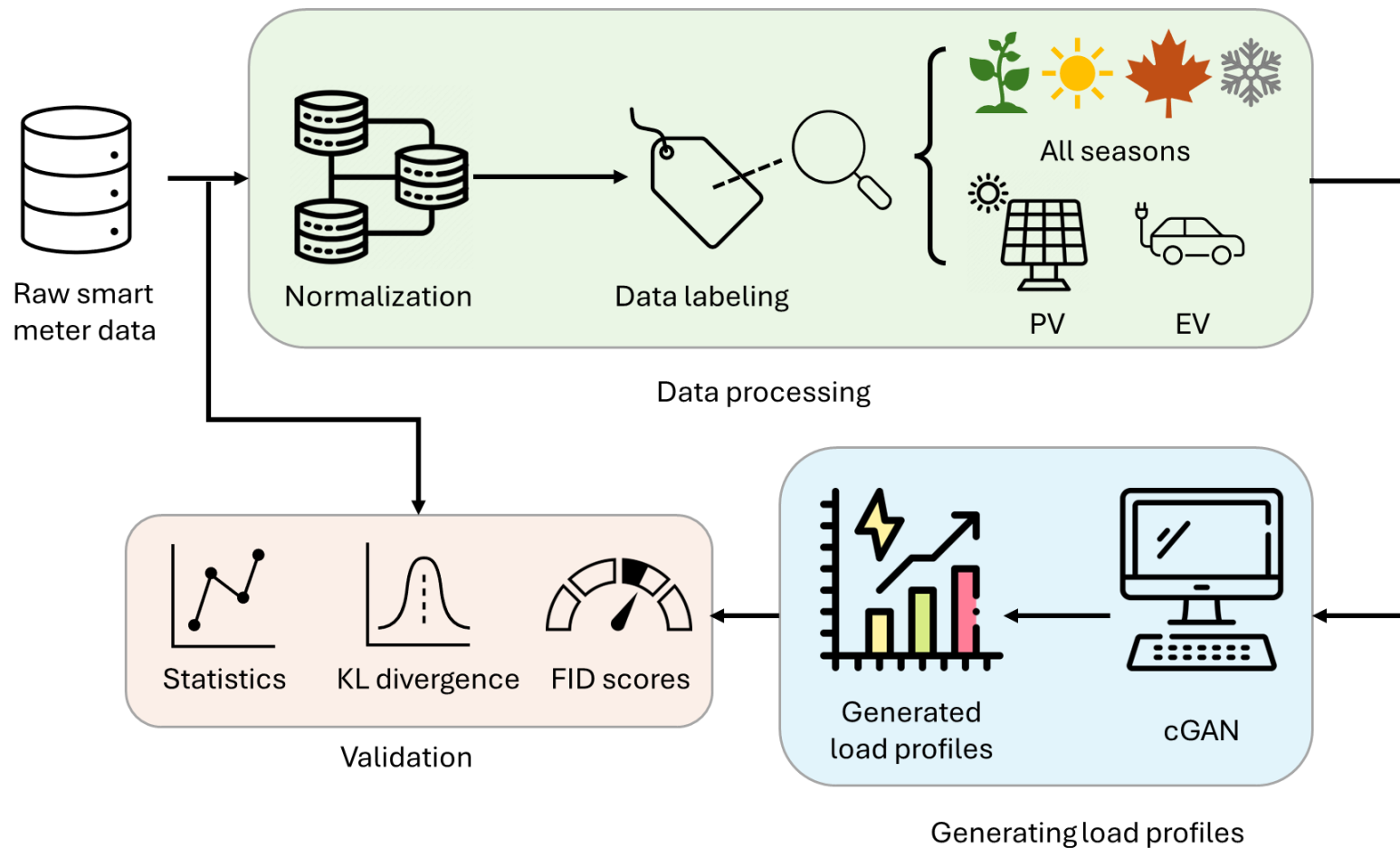
GAN is still challenging ...

- Generator and Discriminator needs to match each other



A Case Study using cGan

Conditional generative adversarial network (cGAN) for generating building load profiles with photovoltaics and electric vehicles



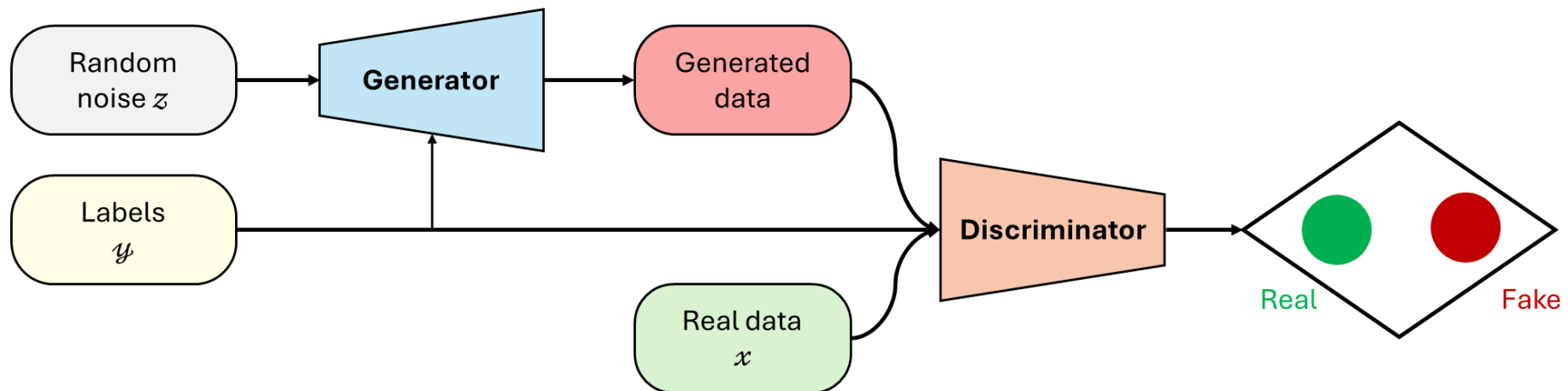
Li, Y., Dong, B., & Qiu, Y. (2025). Conditional generative adversarial network (cGAN) for generating building load profiles with photovoltaics and electric vehicles. *Energy and Buildings*, 335, 115584.

Approach

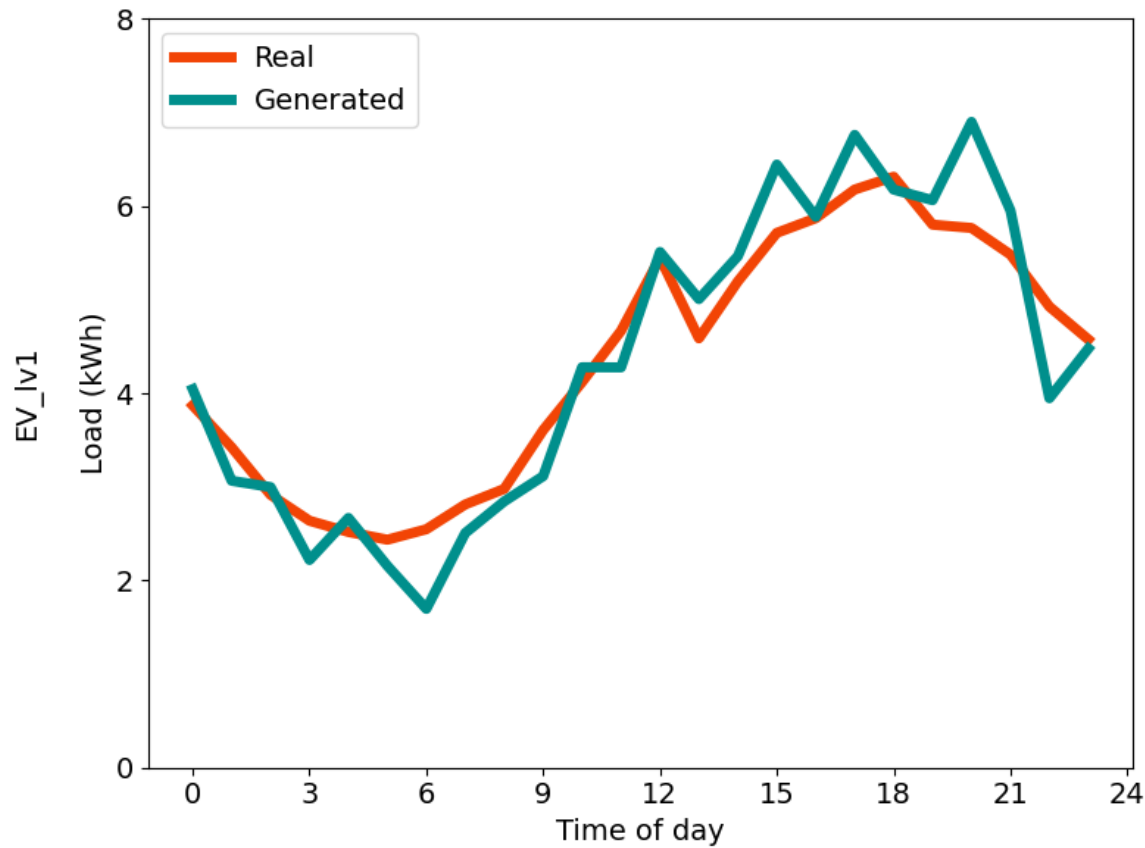
Generating load profiles

Conditional GAN (cGAN) – Expanding traditional GAN including additional conditioned input

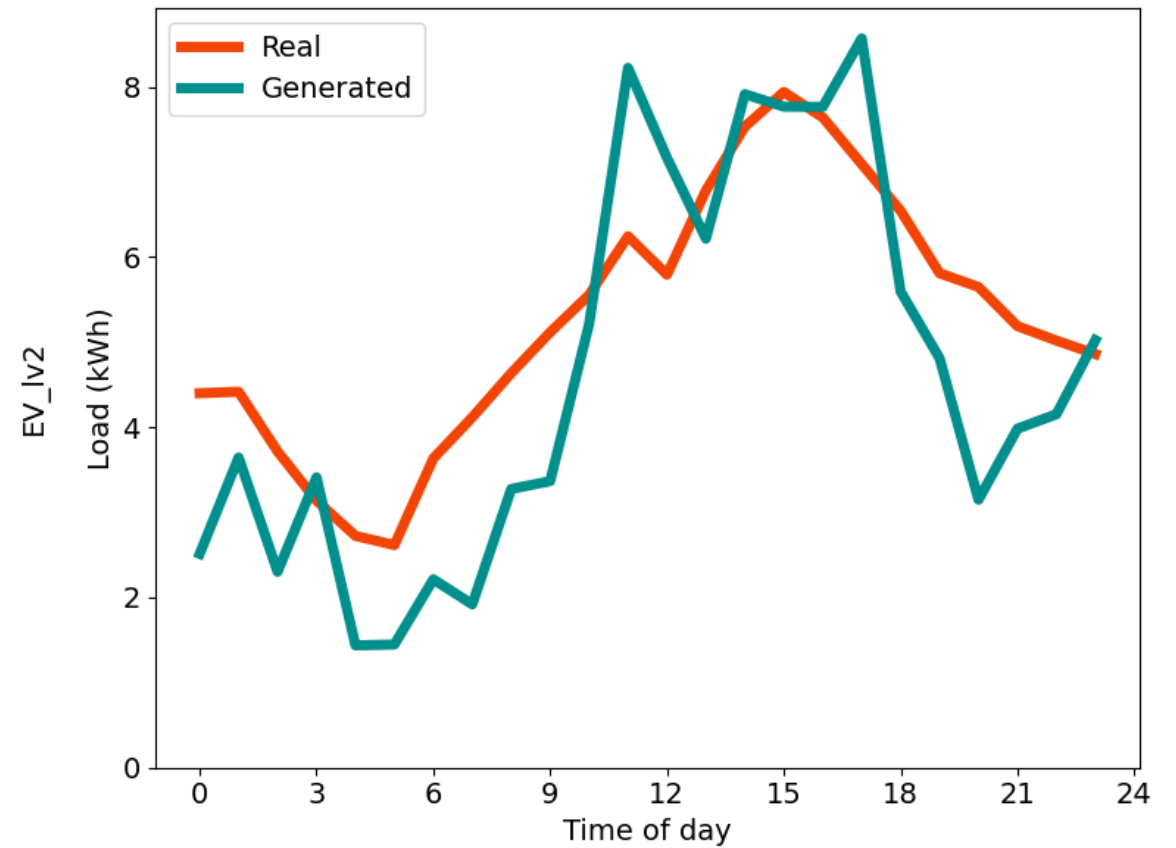
- Generator: improves realistic conditioned load profile generation
- Discriminator: sharpens in spotting subtle differences within the context of the same conditioning label



Results – Generated load profiles



EV Iv1 Summer



EV Iv2 Summer

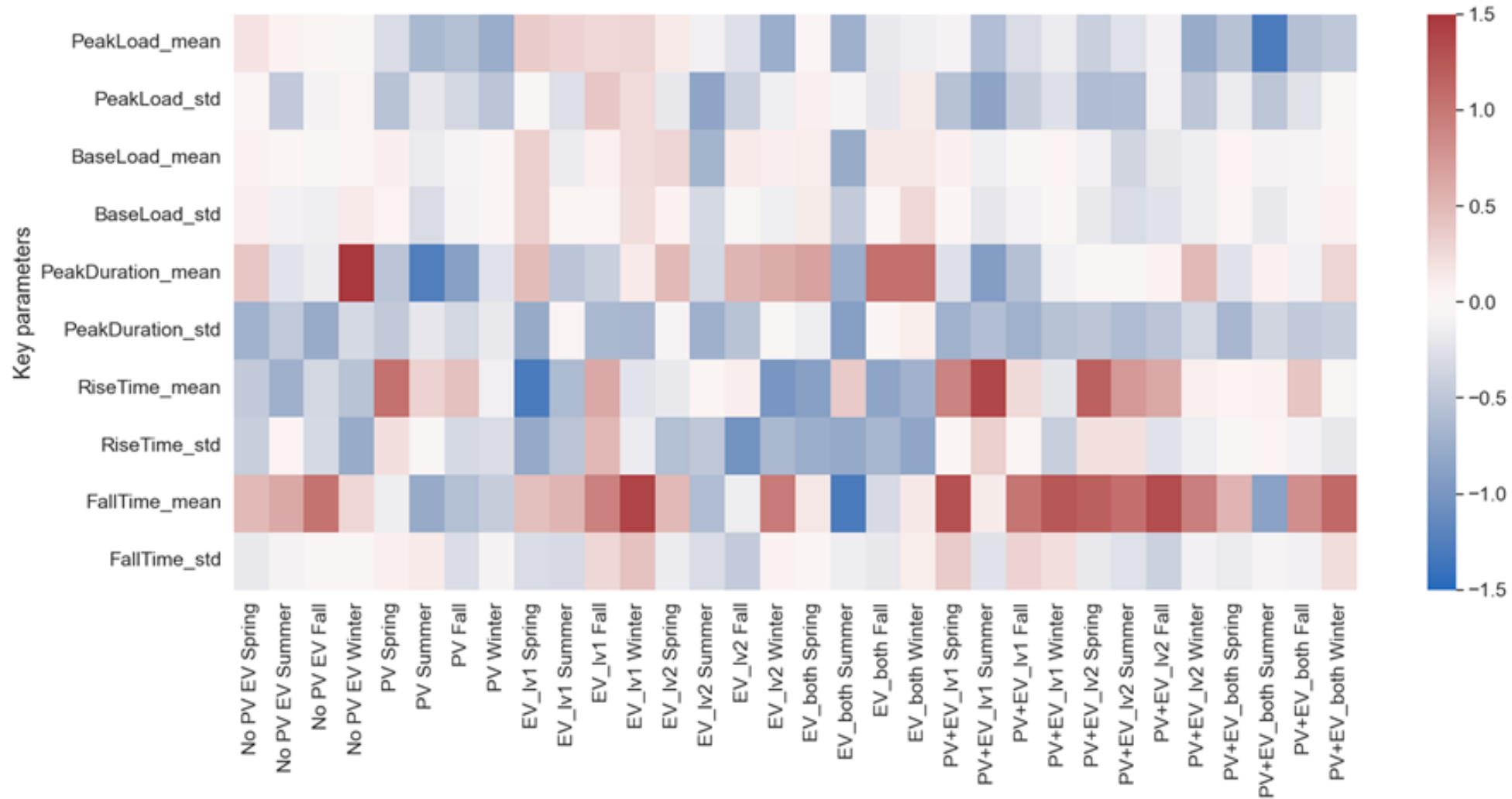
Validation

- **Statistical comparison**
 - Compare the mean and standard deviation of the key parameters.
- **Distribution comparison – 2 metrics**
 - Kullback-Leibler (KL) divergence
 - $D_{KL}(P||Q) = \sum_x P(x) \log(P(x)/Q(x))$
 - $D_{KL} = 0$ indicates two distributions are identical
 - In a similar study, $D_{KL} \approx 0.3$ indicated good quality
 - Fréchet Inception Distance (FID) score
 - $FID = \|\mu_r - \mu_g\|^2 + \text{tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$
 - Evaluate through mean and covariance of the key parameters extracted from generated and real load profiles.
 - Normally used for image generation, but has also been applied to time-series generation task
 - $FID \approx 5$ indicates good quality

Wang, Z., & Hong, T. (2020). Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy and Buildings*, 224, 110299.
Smith, K. E., & Smith, A. O. (2020). Conditional GAN for timeseries generation. arXiv preprint arXiv:2006.16477.

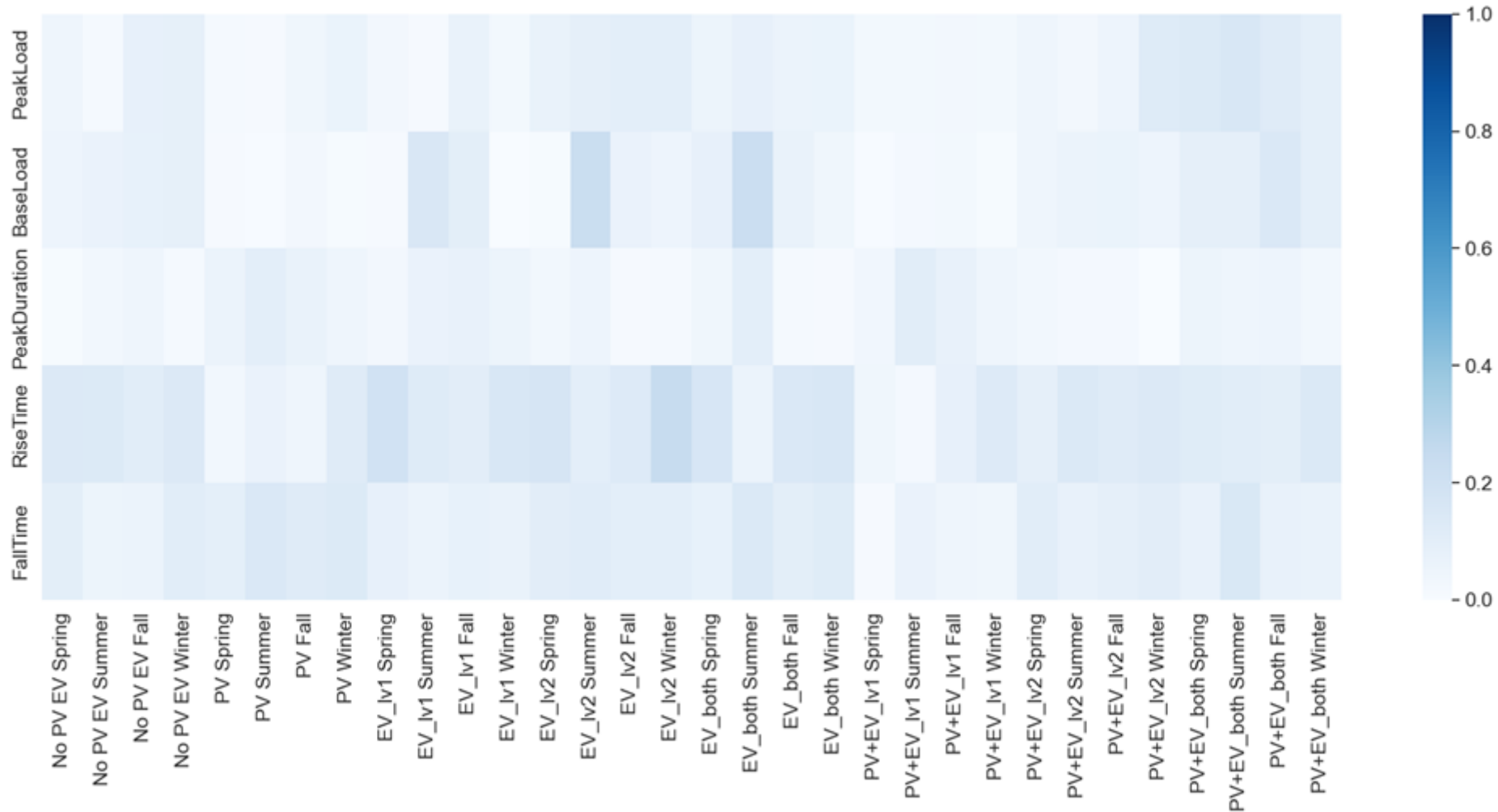
Results – Statistical Comparison

Statistical comparison showed small differences for all key parameters. **➡ Close similarity!**




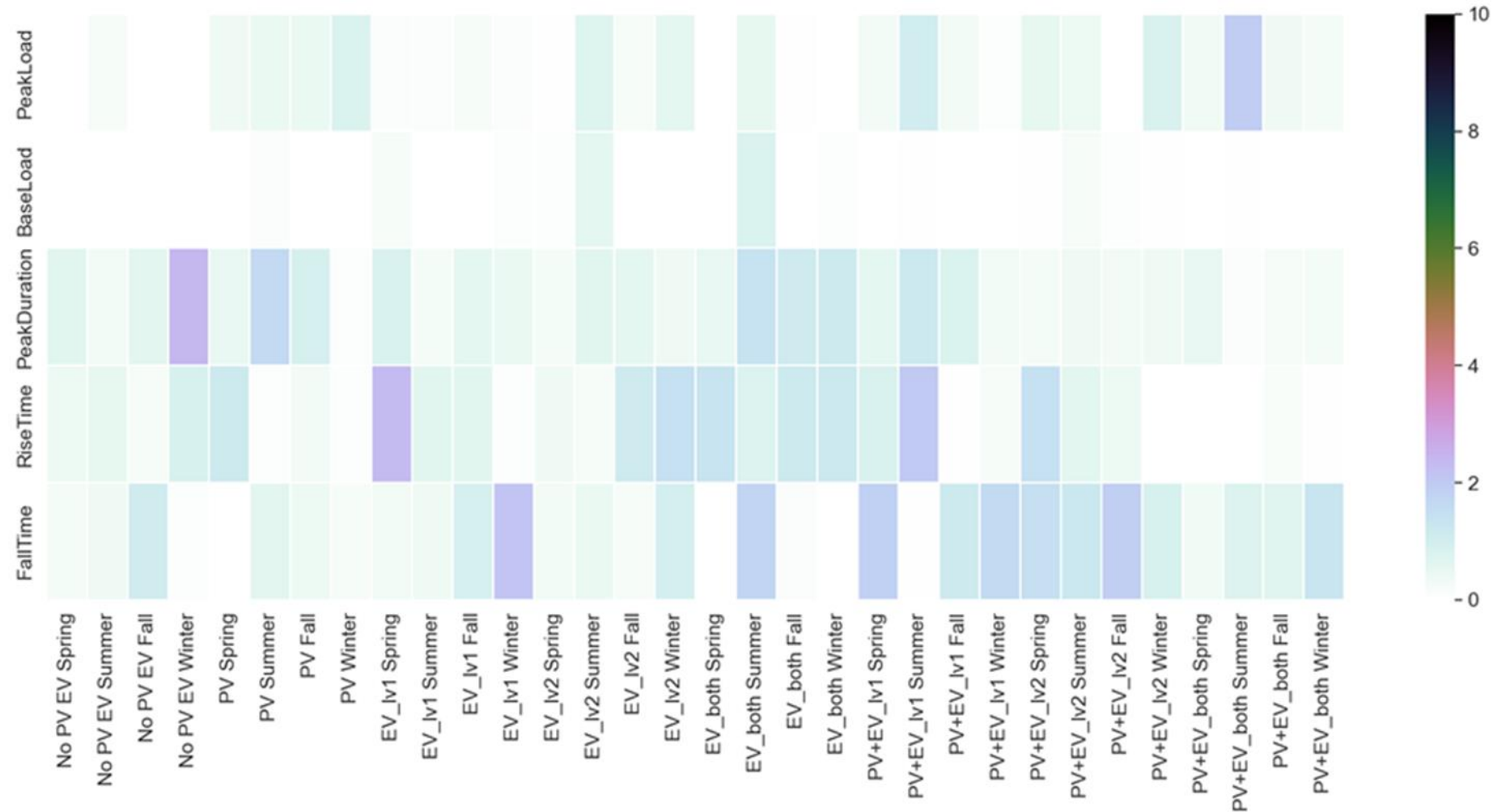
Results – KL divergence

KL Divergences are less than 0.3 for all categories. ➡ Close similarity!



Results – FID score

FID scores are less than 3.  **Close similarity!**



Any Questions?

bidong@syr.edu