Generative models II
Generative adversarial networks

Nikita Kazeev

Where can I use it?

Lots and lots of image manipulation fun

Lots and lots of text manipulation fun

Some (High-Energy) Physics, very preliminary:

- Jets images
- Simulations of calorimeters and RICH
- Cosmological maps

Problem: comparing images

For naive pixel wise metric “cat on the left” is closer to “dog on the left” than to “cat on the right”

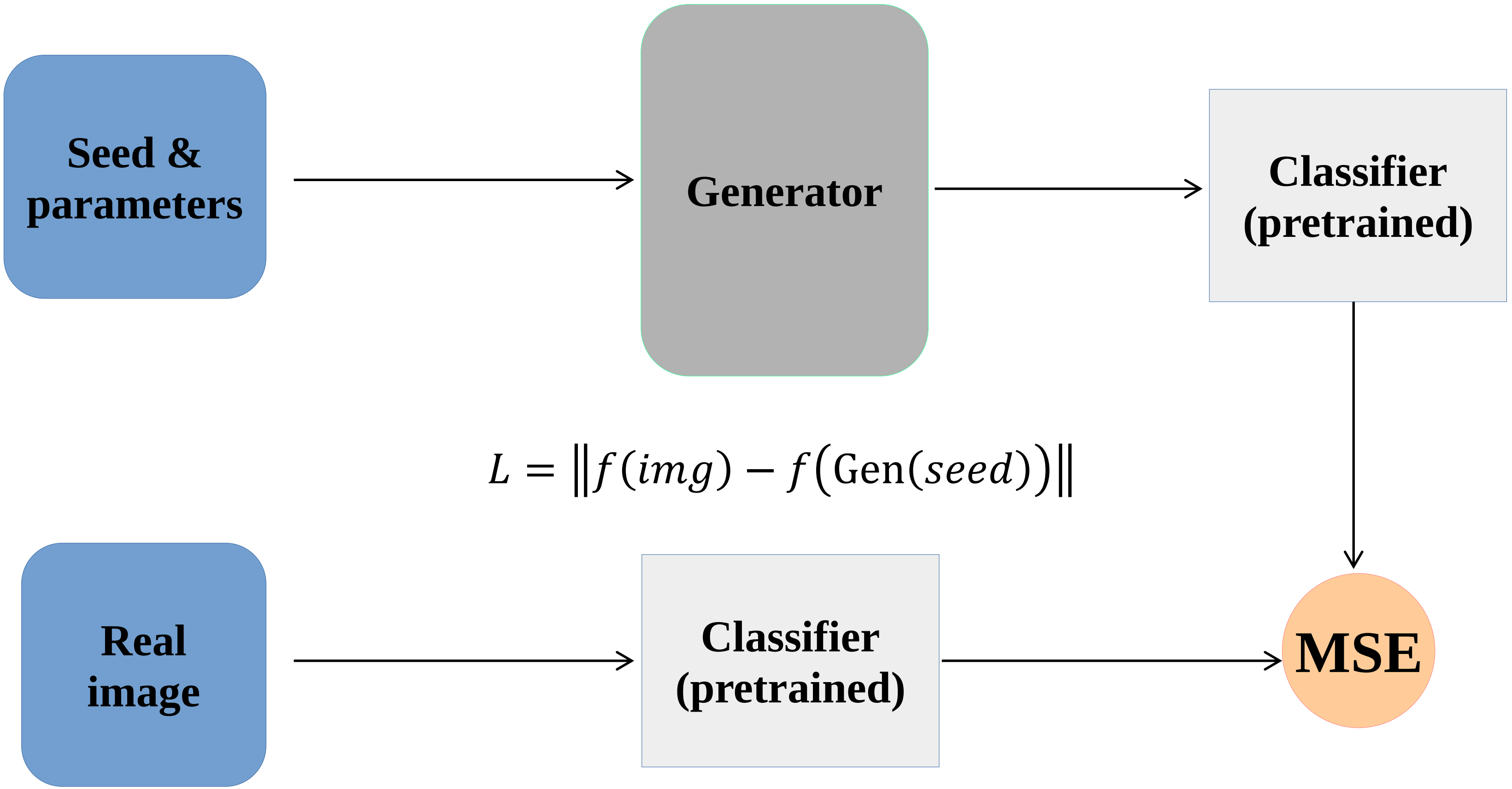
We may want to avoid that effect

Can we obtain image representation that is less sensitive to small shifts?

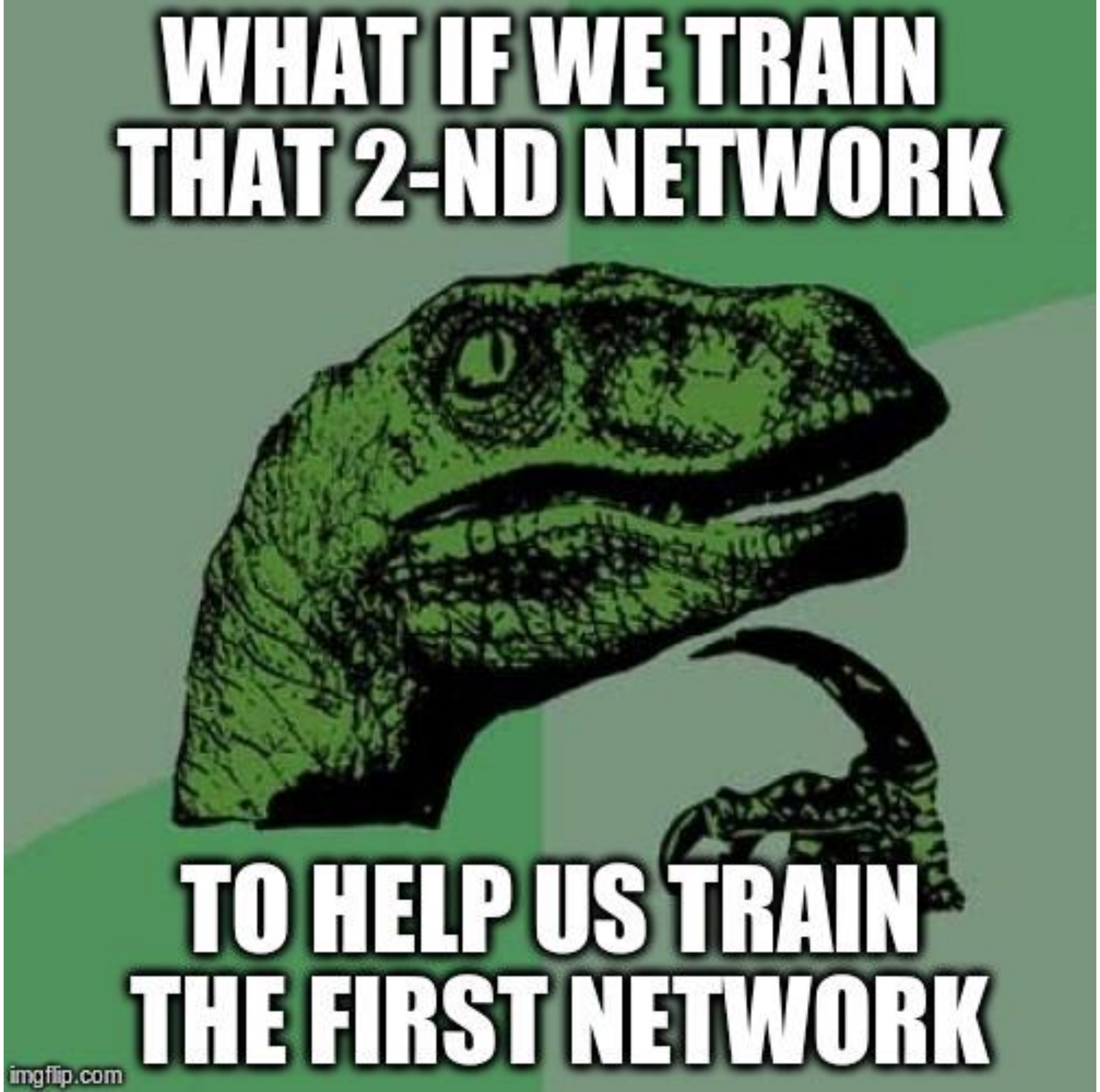
Ideas?

Do we have a representation that focuses on
“semantics”?

Sketch: using pre-trained classifiers



$$L = \left\| f(img) - f\left(\mathrm{Gen}(seed)\right) \right\|$$


WHAT IF WE TRAIN
THAT 2-ND NETWORK
TO HELP US TRAIN
THE FIRST NETWORK
imgflip.com

Generative Adversarial Networks

Generator

Content

Feedback

Discriminator

Generate image
(should be plausible)

Tell if image is plausible
(image) → P(fake)

Generative Adversarial Networks

- Generator

- Discriminator

Training a GAN: Algorithm

for k in 1...K

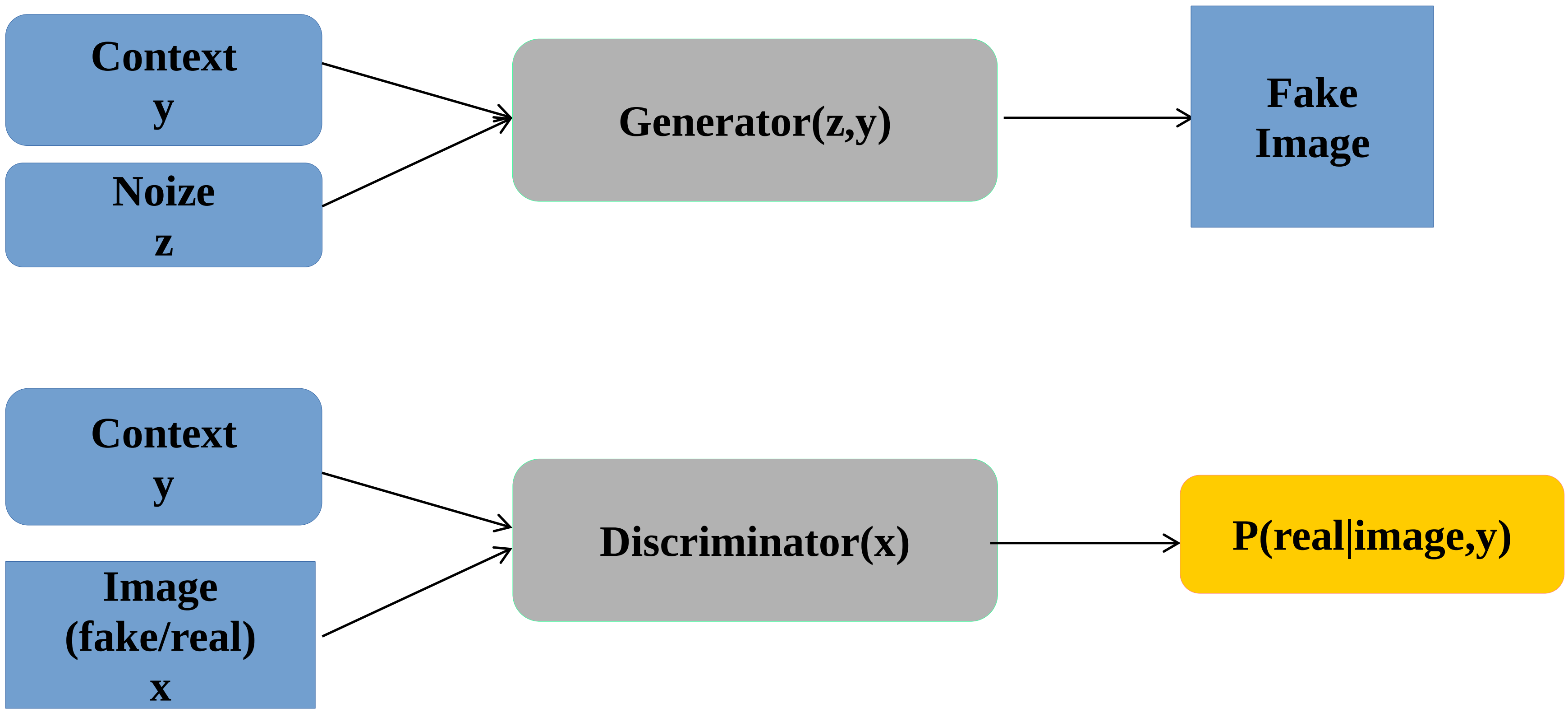
    sample a batch of noise $\mathbf{z}$ and images $\mathbf{x}$

    train step for discriminator

sample a batch of noise $\mathbf{z}$ and images $\mathbf{x}$

train step for generator

Conditional Adversarial Networks



Do we really learn a distribution?

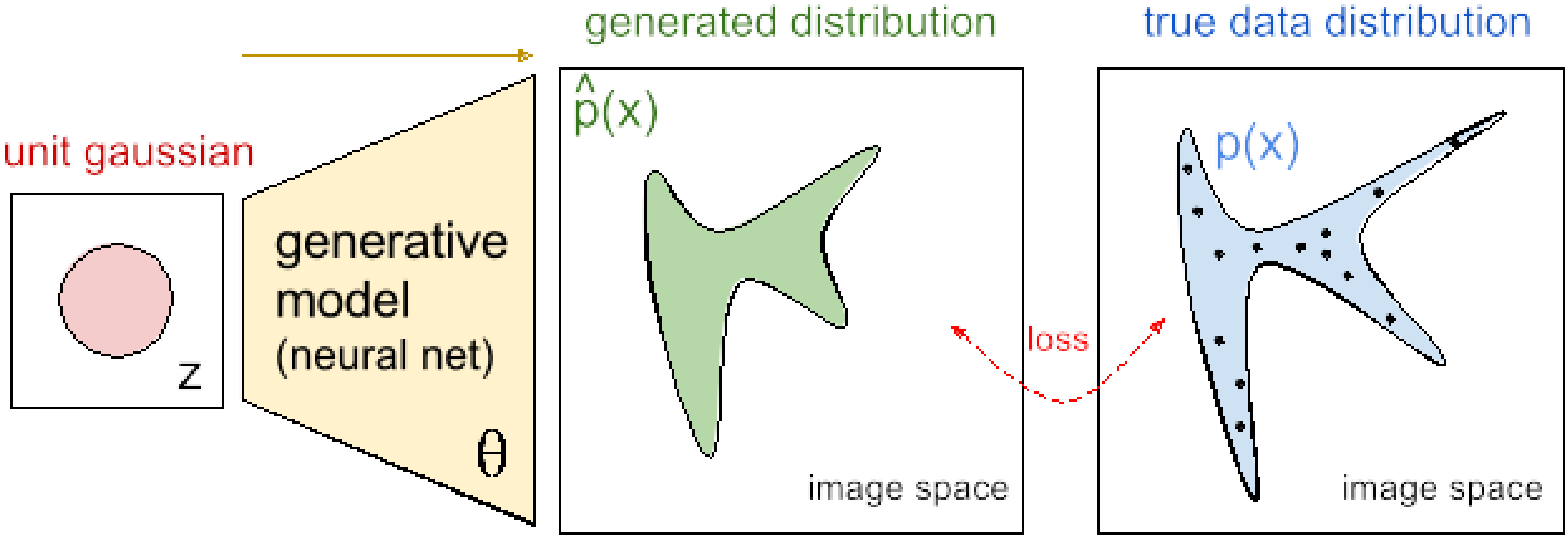


Image: https://blog.openai.com/generative-models/

How does one measure the distance between two distributions?

Kullback–Leibler divergence

$$D_{KL}(P \parallel Q) = \int \log\left(\frac{P(x)}{Q(x)}\right) P(x)dx$$

Asymmetric

Possibly infinite

Has roots in information theory

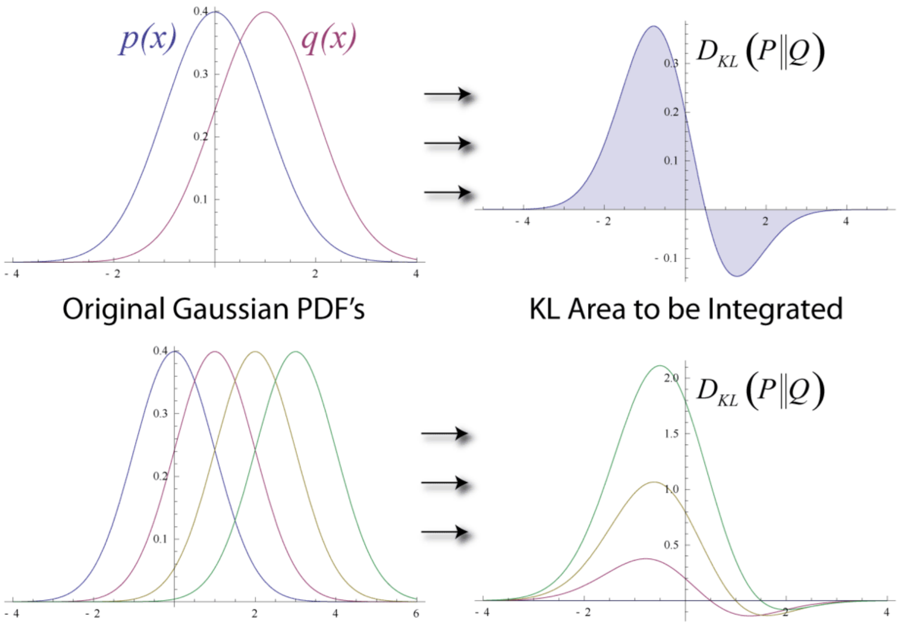


Image: Wikipedia

Jensen-Shannon (JS) divergence

way if we switch P and Q nothing c

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}\left(P \,\middle\|\, \frac{P+Q}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(Q \,\middle\|\, \frac{P+Q}{2}\right)$$

Historically was the first used in GANs (if you hear just “GAN”, it’s likely optimizing JS)

In theory, it’s proven that a GAN converges. In practice however…

Some believe that one reason behind GANs’ success is switching the loss from KL (usually used in maximum-likelihood approach) to JS

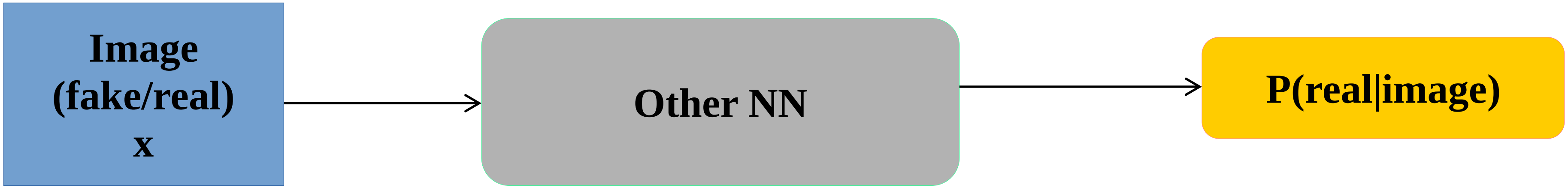
Jensen-Shannon aka “ordinary” GAN

- Generator

$$L_G = -\log \mathrm{Disc}\big(\mathrm{Gen}(z)\big)$$

- Discriminator

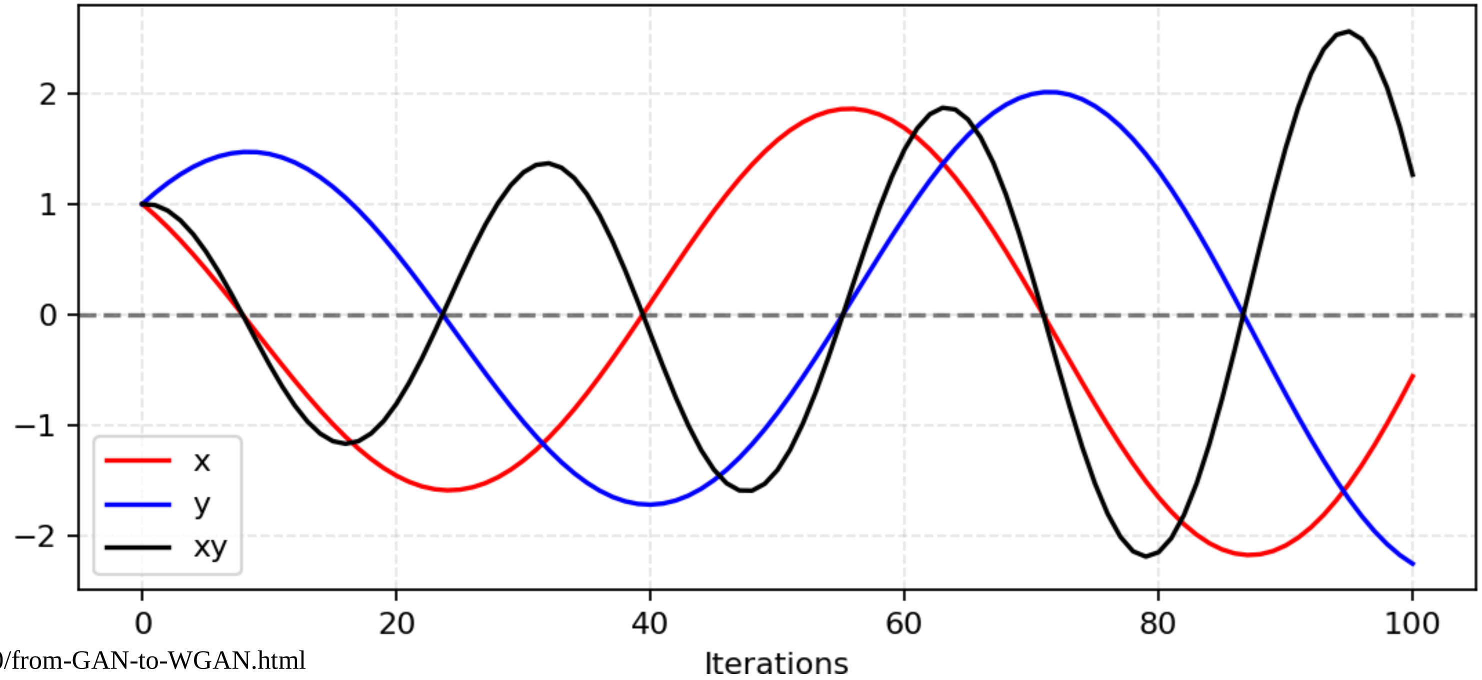
$$L_D = -\log\big[1 - \mathrm{Disc}\big(\mathrm{Gen}(z)\big)\big] - \log \mathrm{Disc}(x)$$



Problem #1: Hard to achieve Nash equilibrium

Player I takes control of x to minimize $f_1(x) = xy$

Player constantly updates y to minimize $f_2(x) = -xy$



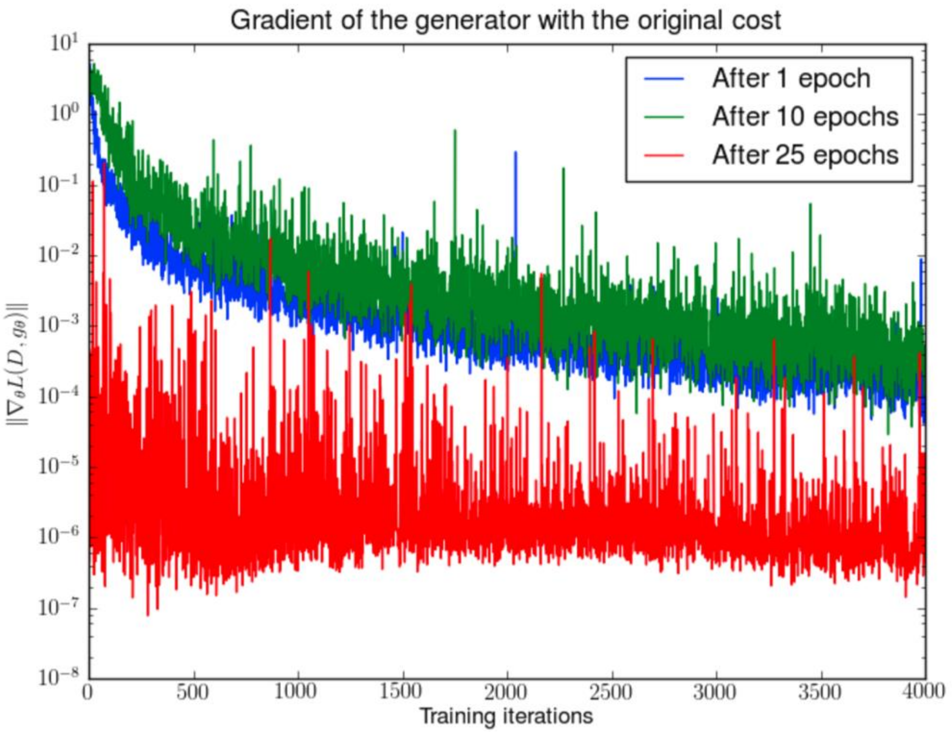
https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Problem #2: Vanishing gradient

If discriminator is perfect,
$D(x) = 1, \forall x \in p_r$ and $D(x) = 0, \forall x \in p_g$

Loss function $L = 0$

No gradient to update the loss during learning iterations

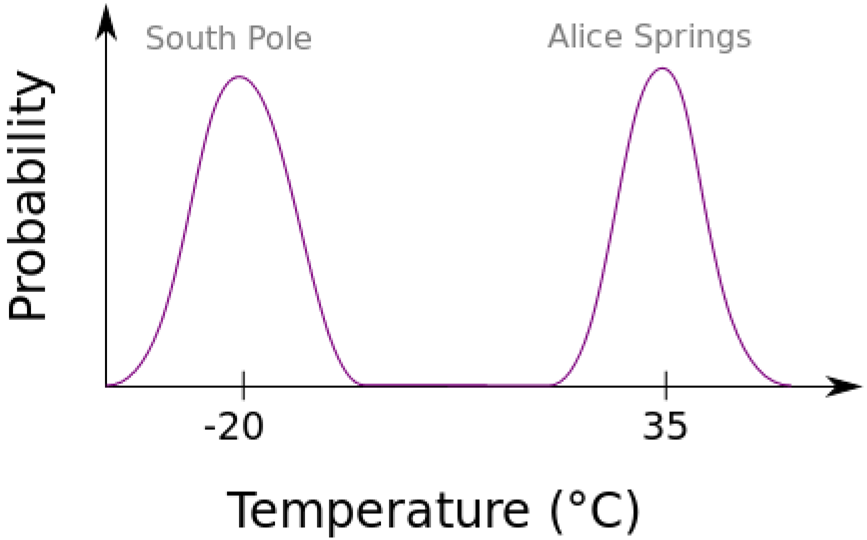


https://arxiv.org/pdf/1701.04862.pdf

Problem #3: Mode collapse

that those are fake and accepts AS, then gen wi

Generator can converge to cover just a part of the phase space



http://aiden.nibali.org/blog/2017-01-18-mode-collapse-gans/

Problem #4 No convergence metric

Theoretically, there is a Nash equilibrium

In practice it is usually not achieved

Hard to devise a stopping criteria

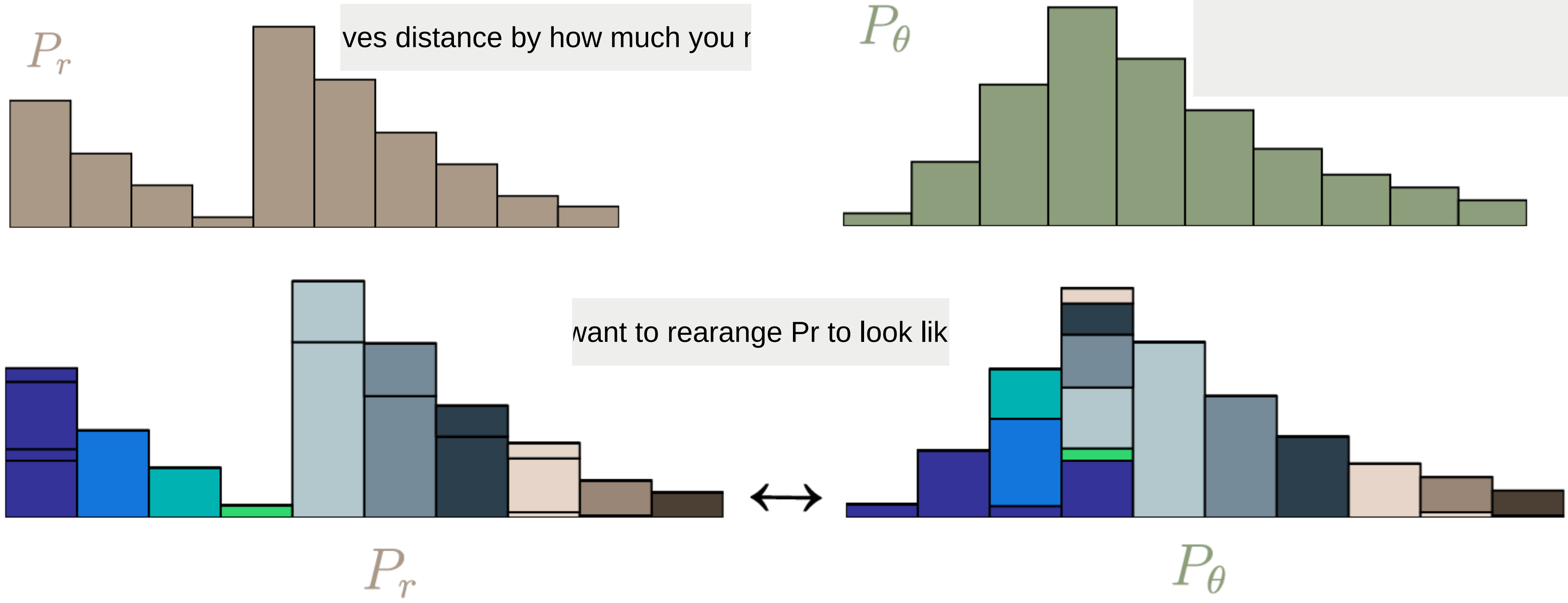
Wasserstein distance aka Earth Mover's distance aka EMD

(Different Similarity Merit)

ves distance by how much you r

ch distribution is viewed as a unit amount o

want to rearange Pr to look lik



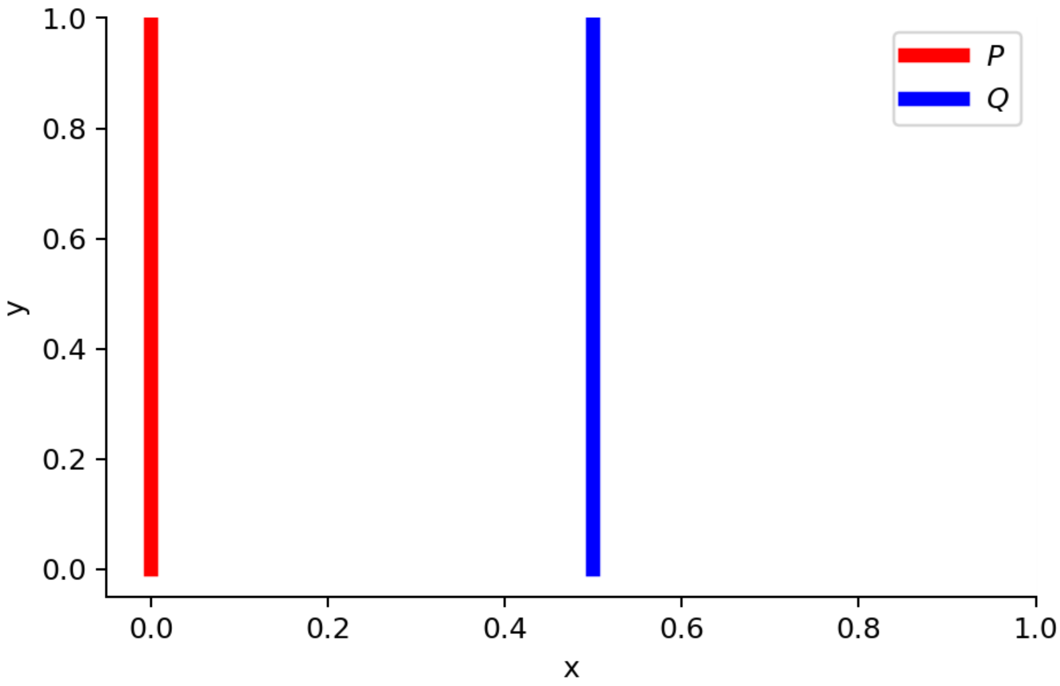
https://vincentherrmann.github.io/blog/wasserstein/

Why Wasserstein is better than JS or KL divergence? #1

Smooth measure, even for disjoint distributions

JS & KL would have given us infinity

(for these distributions)



WGAN formulation

Using the Kantorovich-Rubinstein duality, EMD can be formulated as fol

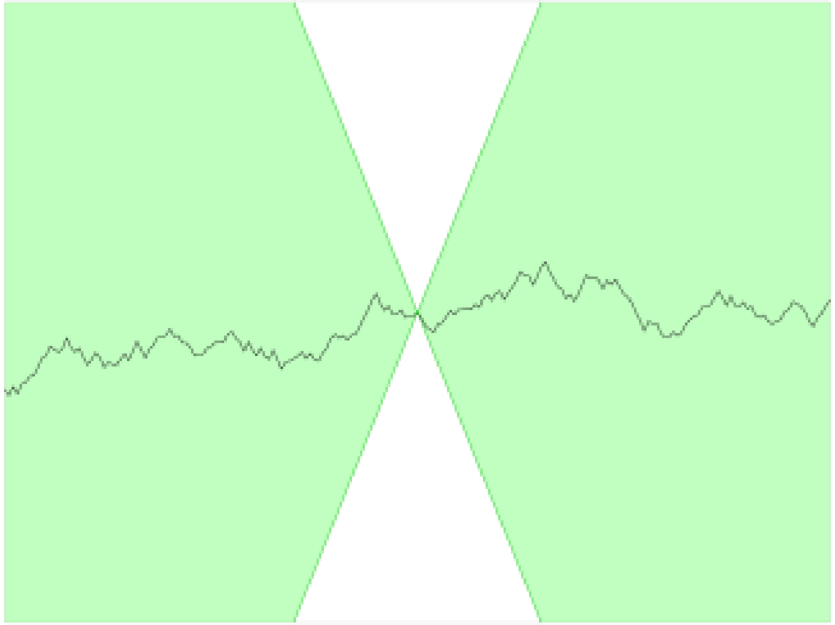
$$W\left(P_r, P_q\right) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_q}[f(x)]$$

The catch? f must be 1-Lipschitz

K-Lipschitz functions

For a Lipschitz continuous function, there is a double cone (shown in white) whose vertex can be translated along the graph, so that the graph always remains entirely outside the cone

$$\forall x_{1,} x_2 \colon |f(x_1) - f(x_2)| \leqslant K|x_1 - x_2|$$



Wikipedia

Lipschitz continuity and NN

Naive way: clip weights. Bad for convergence.

Gradient penalty. Optimal critic has gradient 1 almost everywhere, so add a term to the loss

$$-E_x[(||\nabla_x D(x)||_2 - 1)^2]$$

This is quite a hot area, so expect more

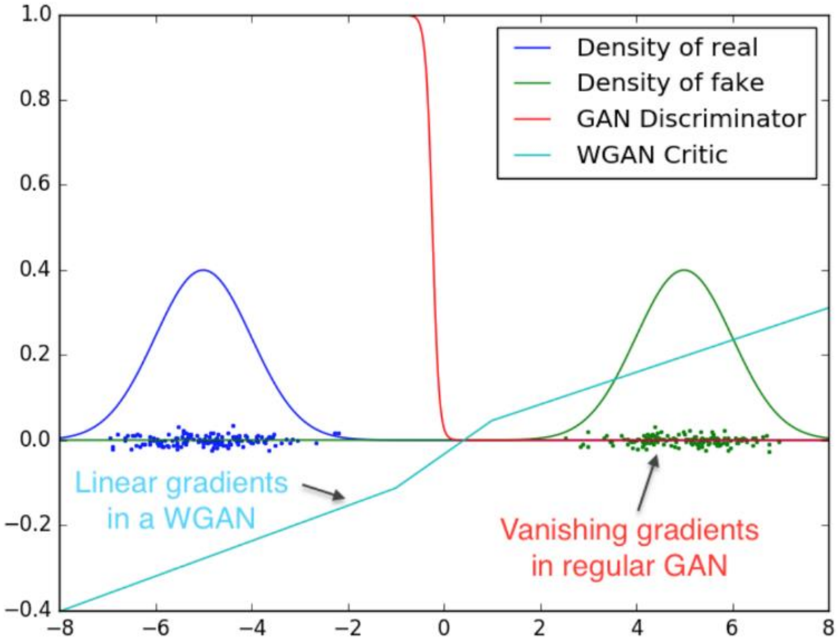
WGAN-GP

Generator $L_G = -\mathbb{E}_z \mathrm{Disc}\big(\mathrm{Gen}(z)\big)$

Discr. $L_D = \mathbb{E}_Z \mathrm{Disc}\big(\mathrm{Gen}(z)\big) - \mathbb{E}_X \mathrm{Disc}(x) - \lambda \mathbb{E}_x [(\|\nabla_x \mathrm{Disc}(x)\|_2 - 1)^2$

Why Wasserstein might be better than JS or KL divergence? #2

No vanishing gradients



https://arxiv.org/pdf/1701.07875.pdf

Practical way to enforce gradient penalty

Enforcing the unit gradient norm constraint everywhere is intractable

The authors propose choosing random points on lines connecting generated and real examples

Why Wasserstein might be better than JS or KL divergence? #3

WGAN critic can be trained until convergence

WGAN critic loss is a meaningful estimate of EMD between the real and generated data

However

WGAN:

Doesn’t go well batch norm
Has slower convergence
Doesn’t work with some (e. g.) ELU activations
Less papers exploring architectures and heuristics

Note: there are other metrics

Name	$D_f(P\|Q)$
Total variation	$\frac{1}{2}\int \lvert p(x) - q(x)\rvert \,\mathrm{d}x$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x$
Reverse Kullback-Leibler	$\int q(x) \log \frac{q(x)}{p(x)} \,\mathrm{d}x$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \,\mathrm{d}x$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)} \,\mathrm{d}x$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \,\mathrm{d}x$
Jeffrey	$\int (p(x) - q(x)) \log \left(\frac{p(x)}{q(x)}\right) \,\mathrm{d}x$
Jensen-Shannon	$\frac{1}{2}\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \,\mathrm{d}x$
Jensen-Shannon-weighted	$\pi \int p(x) \log \frac{p(x)}{\pi p(x)+(1-\pi)q(x)} + (1-\pi)q(x) \log \frac{q(x)}{\pi p(x)+(1-\pi)q(x)} \,\mathrm{d}x$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \,\mathrm{d}x - \log(4)$
α-divergence ($\alpha \notin \{0,1\}$)	$\frac{1}{\alpha(\alpha-1)} \int \left(p(x)\left[\left(\frac{q(x)}{p(x)}\right)^{\alpha} - 1\right] - \alpha(q(x) - p(x))\right) \,\mathrm{d}x$

f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization

Training a GAN

How many iterations of generator and discriminator training?

For JS <u>usually around 1:1</u>

For WGAN, the original paper <u>used</u> 5 discriminator per 1 generator – but you can make it as high as you want

Heuristics for Convolutional GANs (DCGAN)

Replace any pooling layers with strided convolutions (discriminator)
and fractional-strided convolutions (generator)

Use batchnorm in both the generator and the discriminator

Remove fully connected hidden layers for deeper architectures

Use ReLU activation in generator for all layers except for the output,
which uses Tanh

Use LeakyReLU activation in the discriminator for all layers

https://arxiv.org/abs/1511.06434

Intermission: Measuring datasets similarity

Measuring datasets similarity

Sometimes one just has several complex datasets and needs a to measure a meaningful distance between them

- FastFoo generator vs. data
- MC vs data

Measuring datasets similarity: classifier AUC

Pick your favorite classifier (NN, gradient boosting, whatever)

Use cross-validation to check whether it is able to distinguish the datasets

Pro: easy to understand AUC

Con: no real theoretical guarantees

Con: answer depends on the classifier

Measuring datasets similarity: EMD

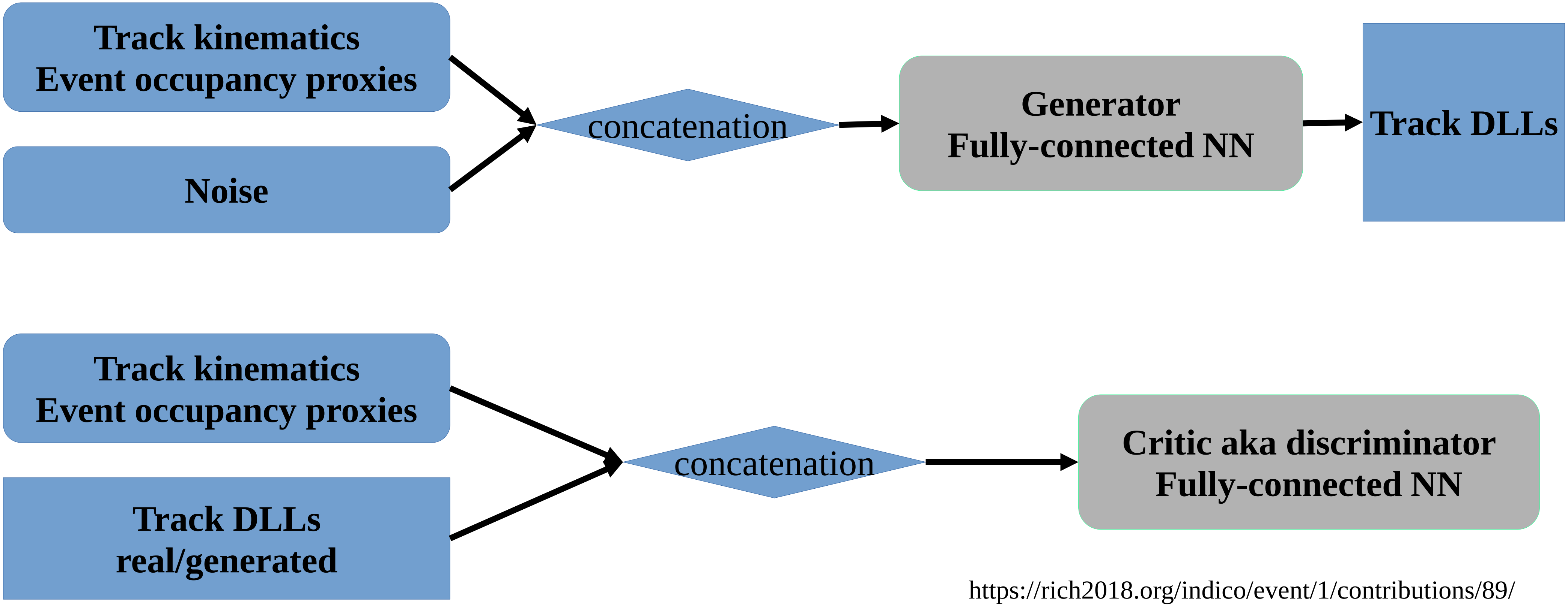
Train a Wasserstein discriminator

Pro: EMD has a well defined meaning

Con: the number is not exactly EMD, there is uncertainty due to the optimization procedure and the way Lipschitz continuity is enforced

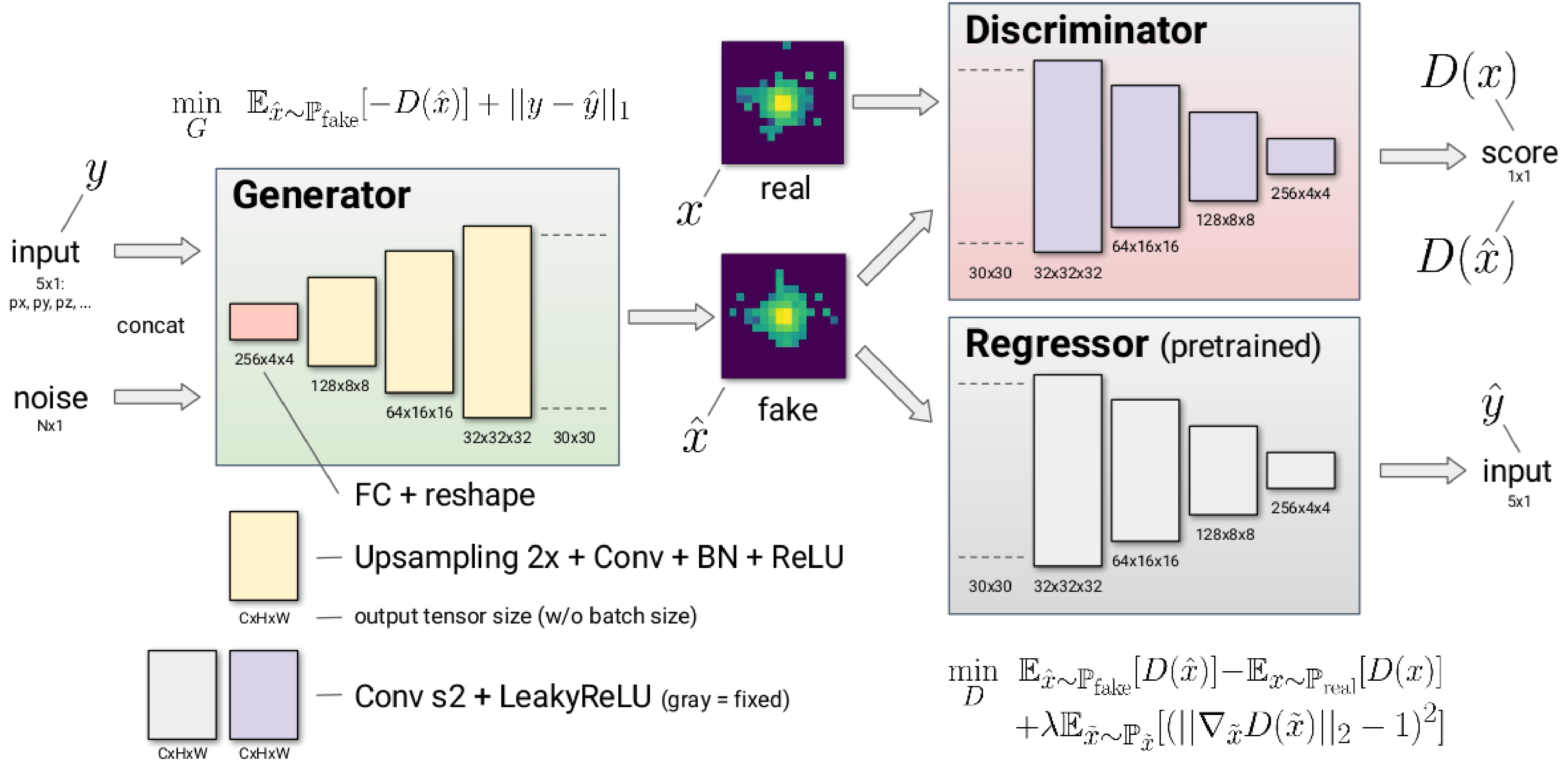
GANs @ HEP

Classic cWGAN for Cherenkov detector high-level observables



https://rich2018.org/indico/event/1/contributions/89/

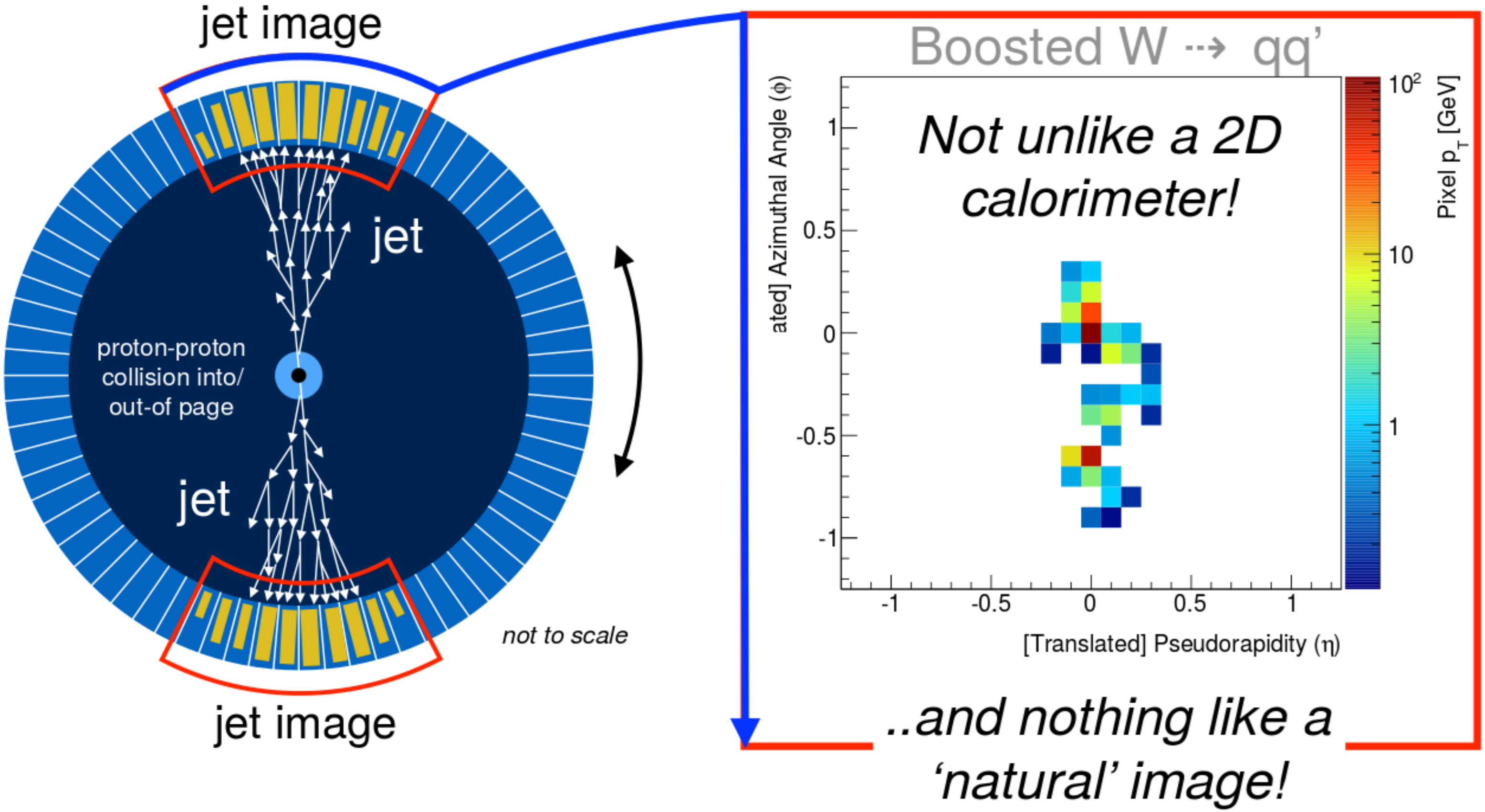
Calorimeter response generation



https://indico.cern.ch/event/587955/contributions/2937612/attachments/1680985/2704964/chep-fast-sim_last3.pdf

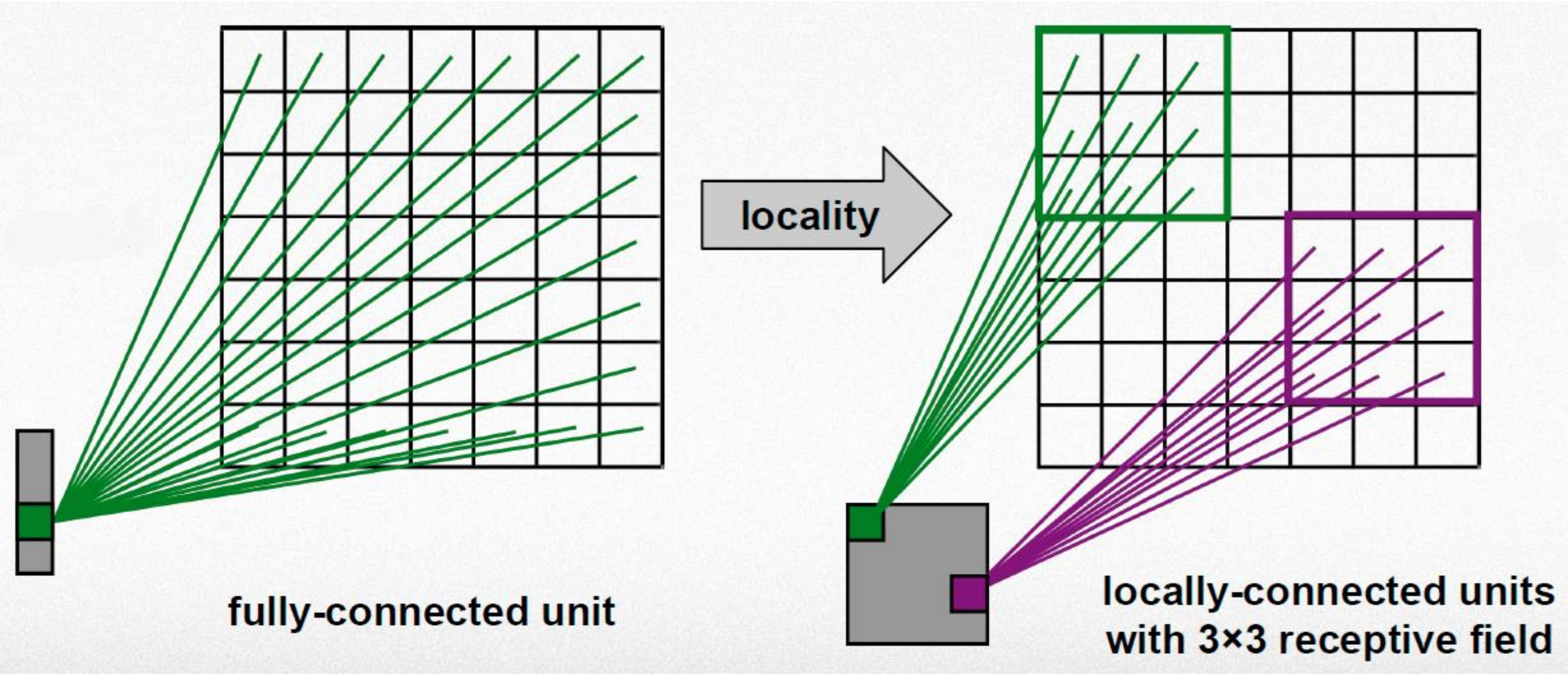
The Jet Image

Jet Image: *A two-dimensional f xed representation of the radiation pattern inside a jet*



axiv, arxiv

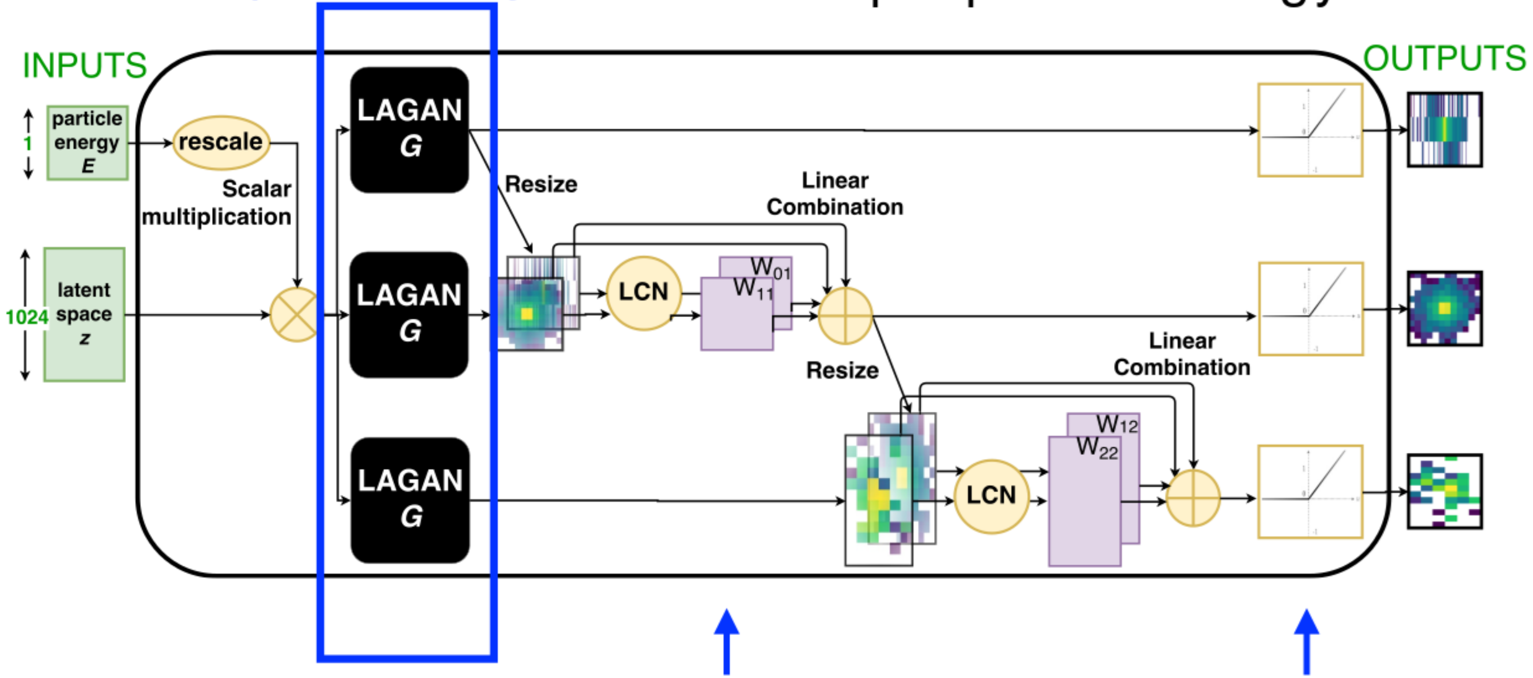
Recap: Locally Connected Layer



Generator Network for CaloGAN

One 'jet image' per calo layer

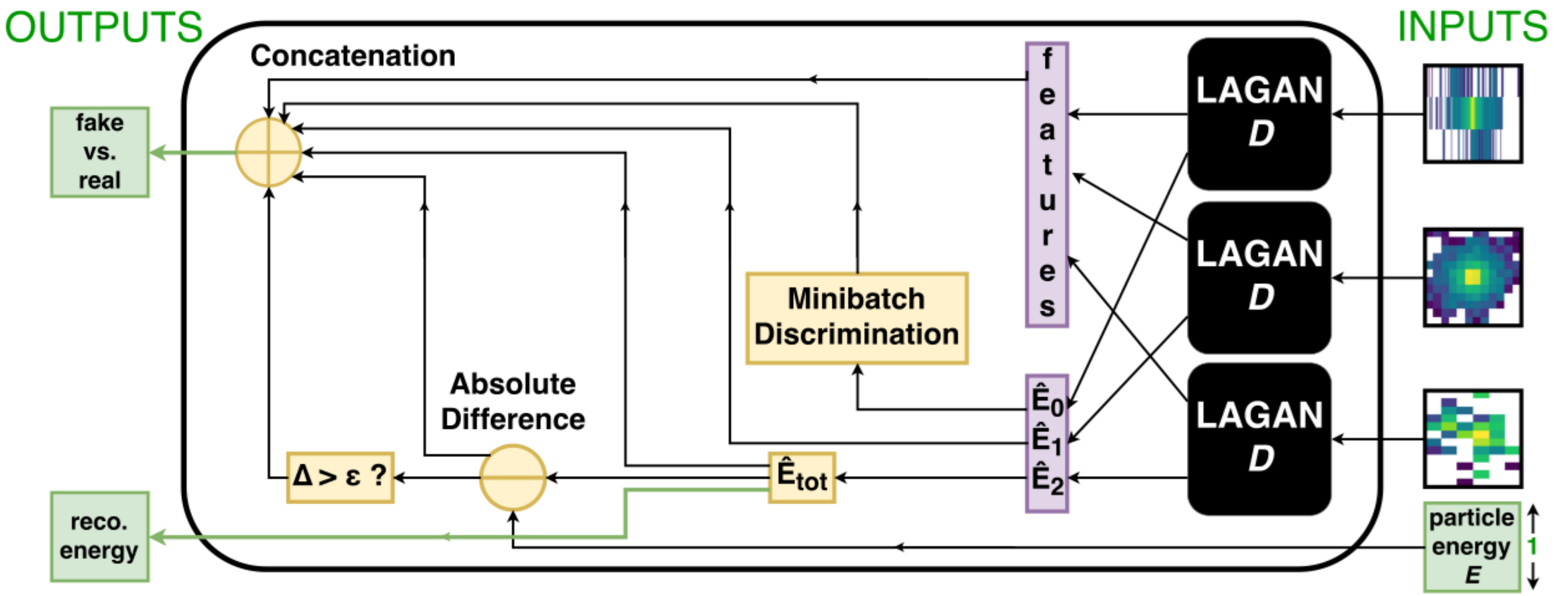
One network per particle type; input particle energy



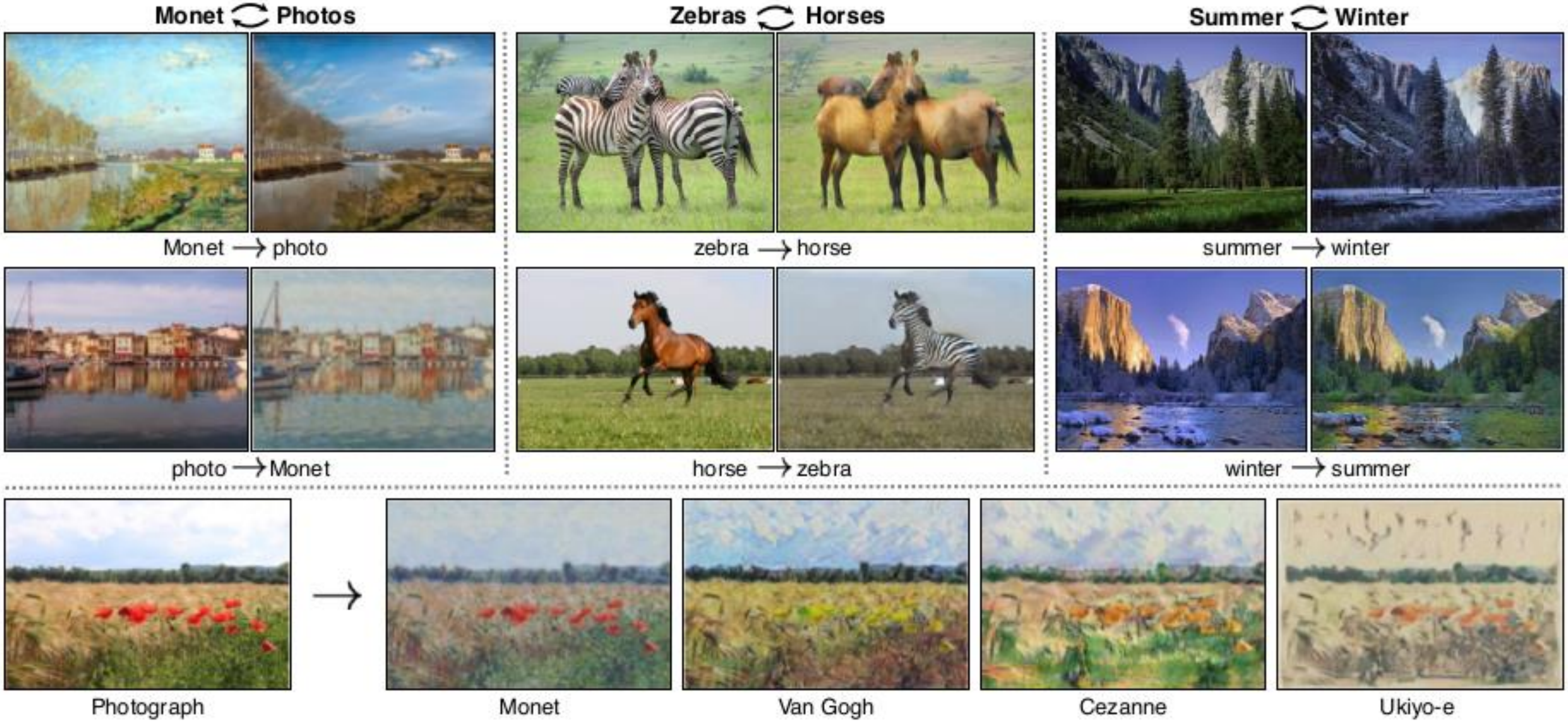
use layer i as input to layer i+1

ReLU to encourage sparsity

Discriminator Network for CaloGAN



Cycle GAN for unpaired images

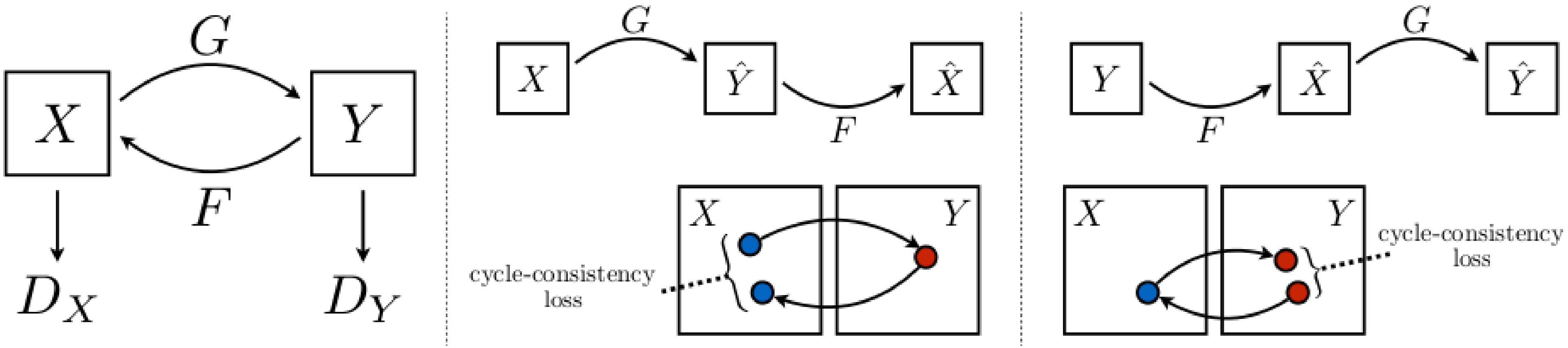


https://arxiv.org/abs/1703.10593

Cycle GAN for unpaired images

Idea: if we don't have image pairs, train two conditional generators $G(z, y) \to x, F(g, x) \to y$

- use non-conditional $D(x), D(y)$
- make sure $\left|F\big(G(x)\big) - x\right| \to \min$



Data $\leftrightarrow$ MC ?

Image: Indiana Jones and the Last Crusade

PartyGAN or Maxim’s quest for the Holy Grail

Feel free to drop a line

Nikita Kazeev
HSE, Rome Sapienza, YSDA, trace amounts of Yandex proper

kazeevn@yandex-team.ru

telegram.me/kazeevn

Bonus

Art style transfer

+

=

Ideas?

Art style transfer

Formulate and optimize texture loss

$$L = \left\|\text{Texture}\left(x_{ref}\right) - \text{Texture}(x_{cand})\right\| + \left\|\text{Content}\left(x_{orig}\right) - \text{Content}(x_{cand})\right\|$$