**Much of the recent progress in unsupervised deep learning has been to invent network architectures that are capable of solving either or both of these related problems directly, without resorting to any auxiliary methods**

**VAE**

**(VARIATIONAL AUTOENCODER)**

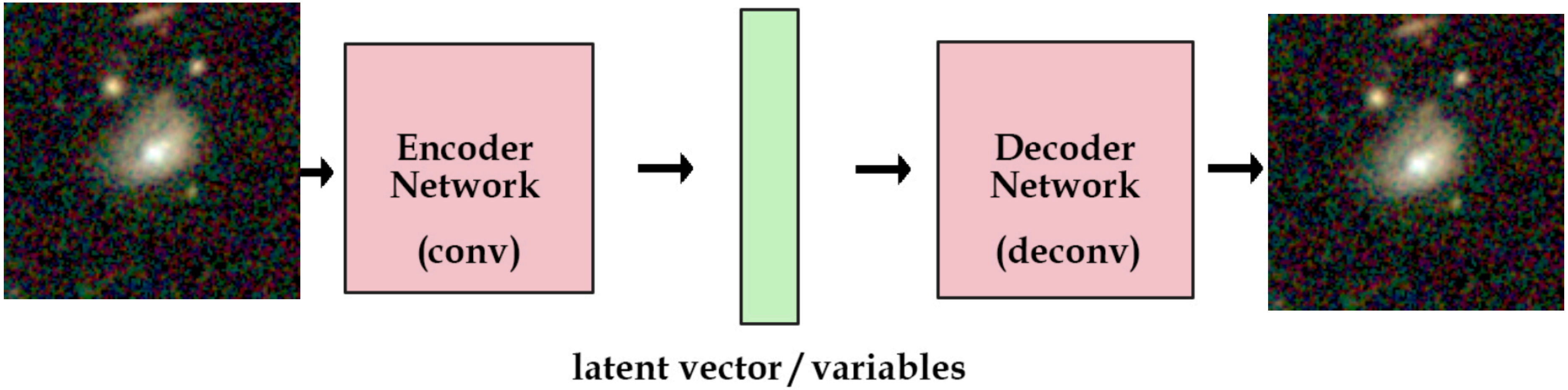
**GAN**

**(GENERATIVE ADVERSARIAL NETOWRK)**

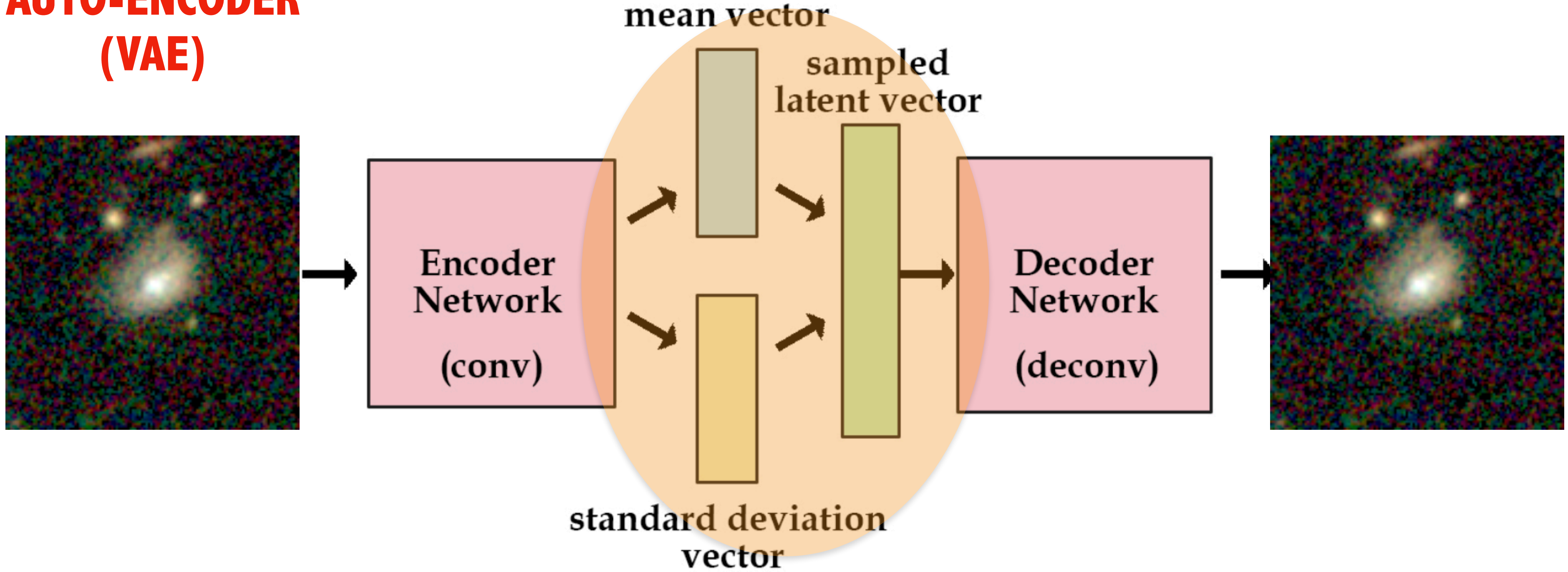
**ARF**

**(AUTOREGRESSIVE FLOWS)**

# AUTO-ENCODER

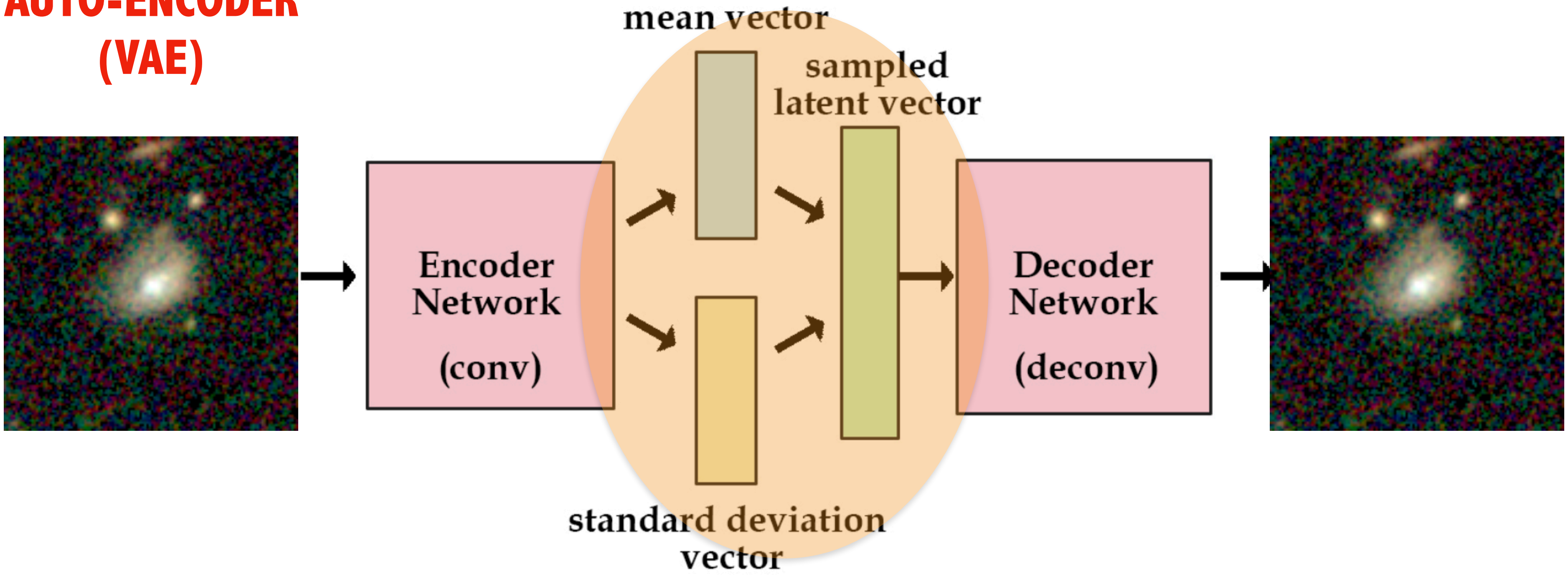



# VARIATIONAL AUTO-ENCODER (VAE)



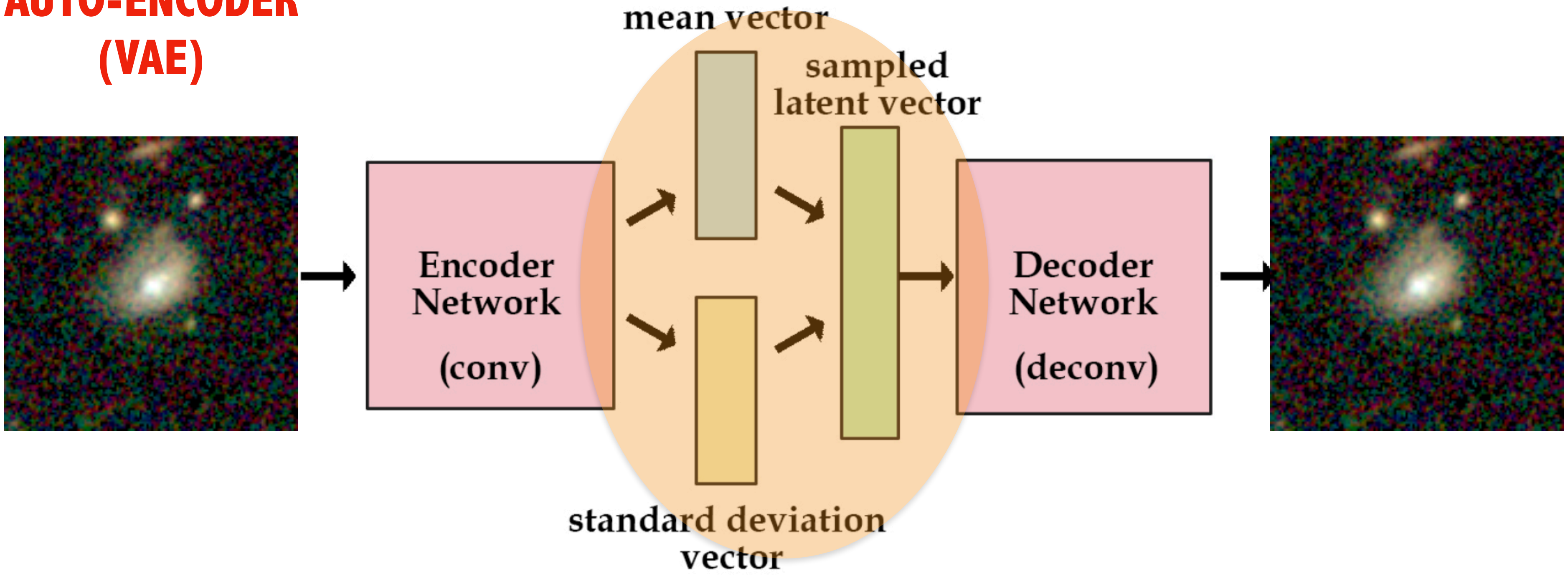

**LET'S MODEL THE LATENT SPACE WITH A MIXUTRE OF GAUSSIANS**

# VARIATIONAL AUTO-ENCODER (VAE)



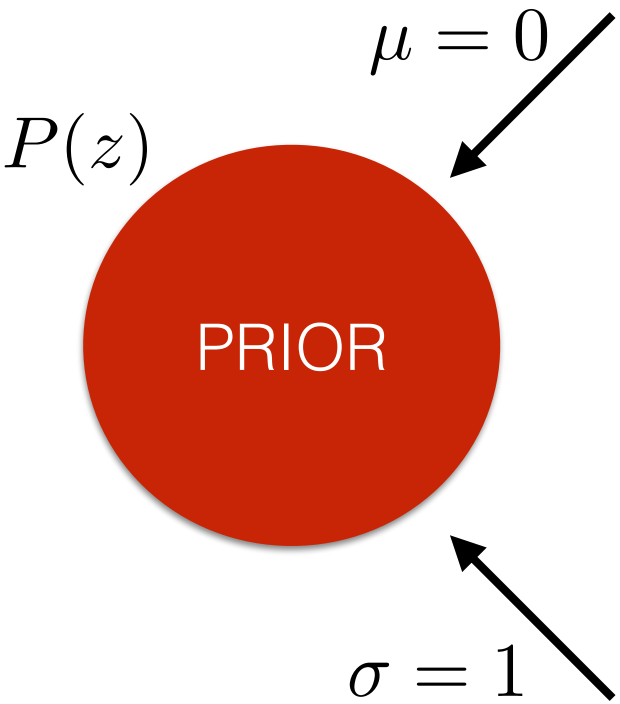

**HOWEVER, NOTHING GUARANTEES US THAT THE LATENT SPACE CAN BE MODELLED BY A MIXTURE OF GAUSSIANS….**

# VARIATIONAL AUTO-ENCODER (VAE)

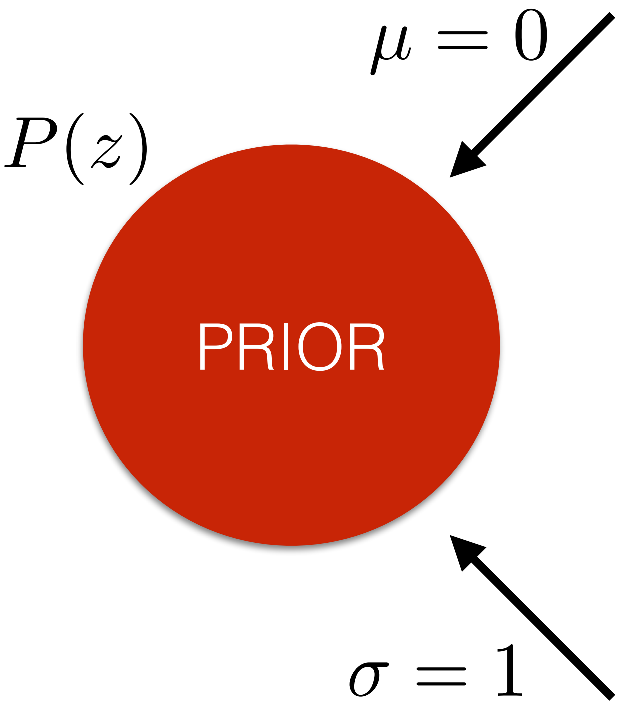



**HOWEVER, NOTHING GUARANTEES US THAT THE LATENT SPACE CAN BE MODELLED BY A MIXTURE OF GAUSSIANS….**

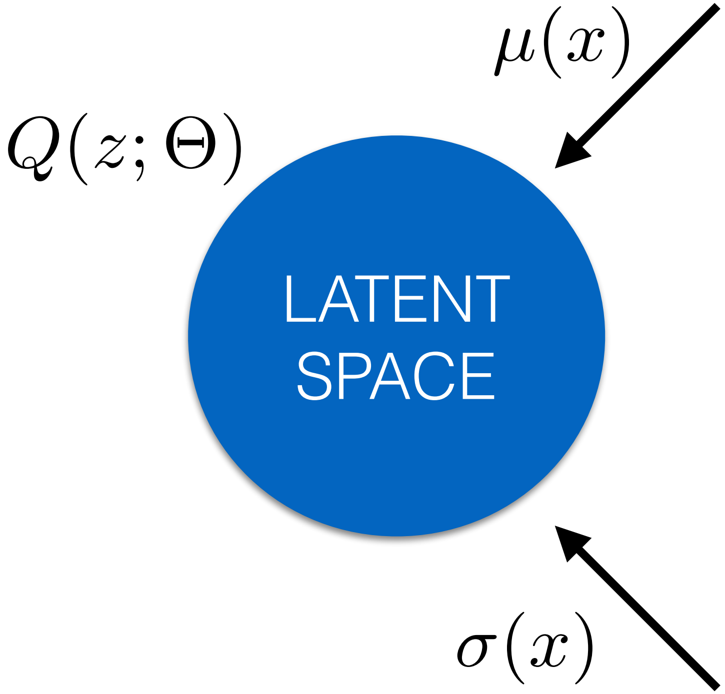
**… LET'S FORCE IT TO BE GAUSSIAN LIKE!**



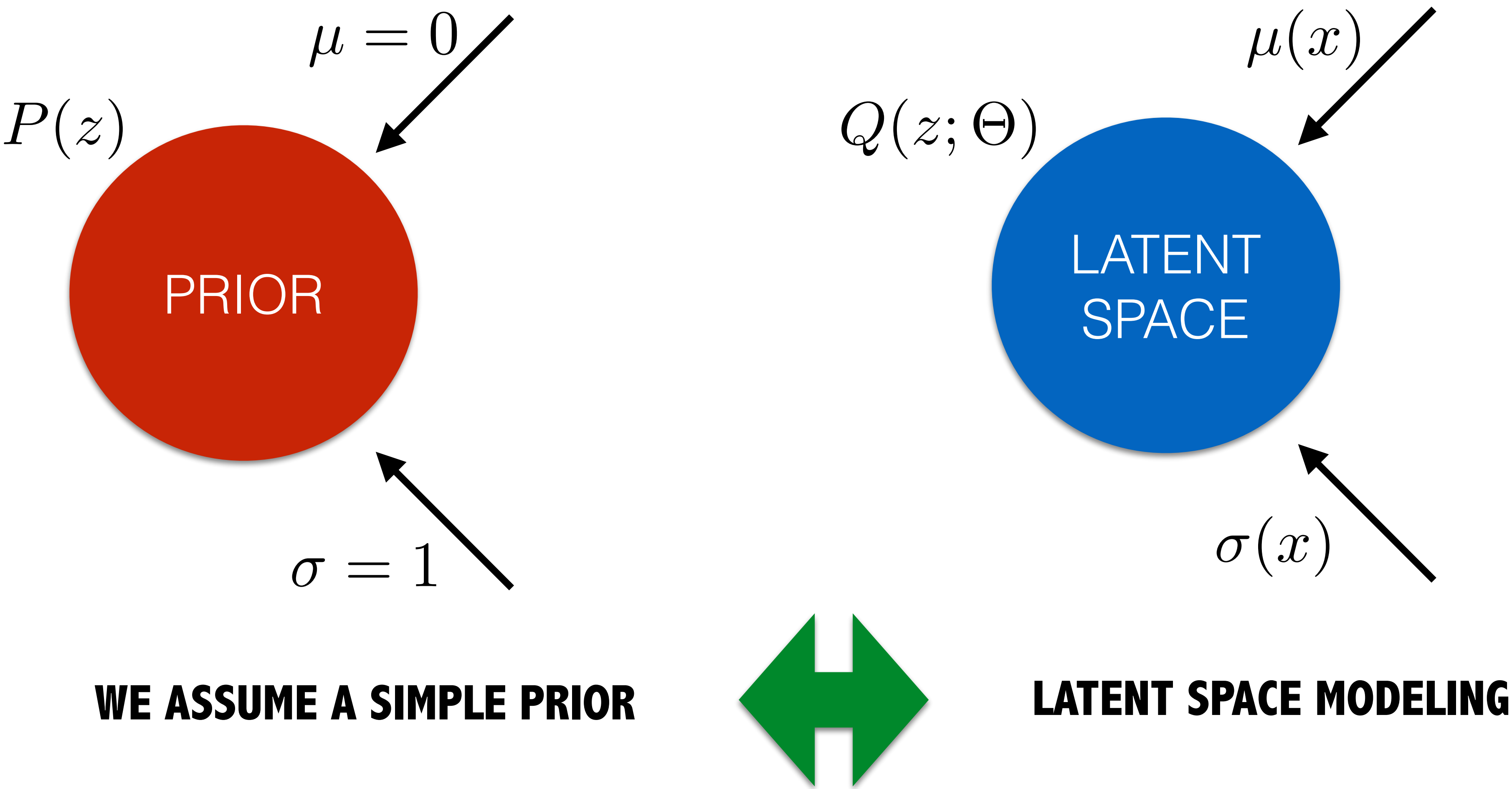

**WE ASSUME A SIMPLE PRIOR**



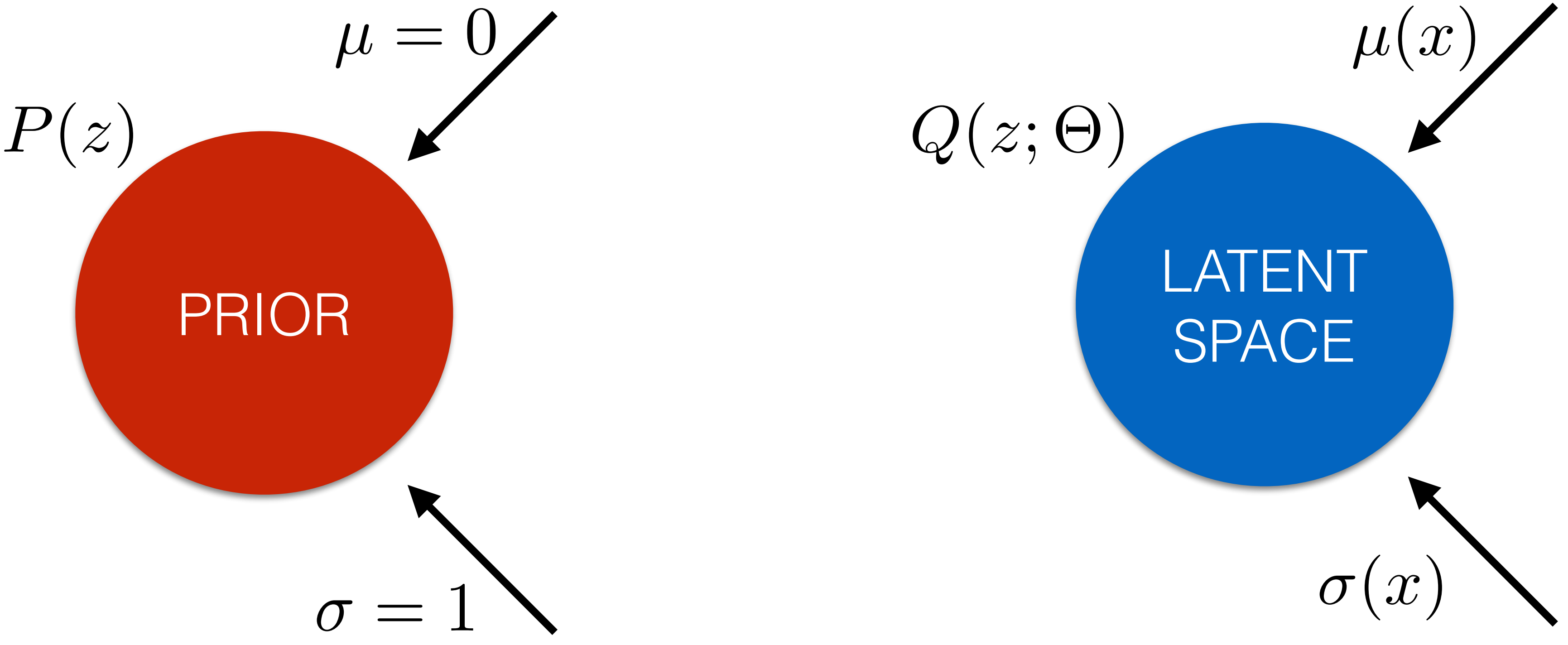

**WE ASSUME A SIMPLE PRIOR**




**LATENT SPACE MODELING**




**WE WANT Q TO BE CLOSE TO THE PRIOR…**




**WE ASSUME A SIMPLE PRIOR**

**LATENT SPACE MODELING**

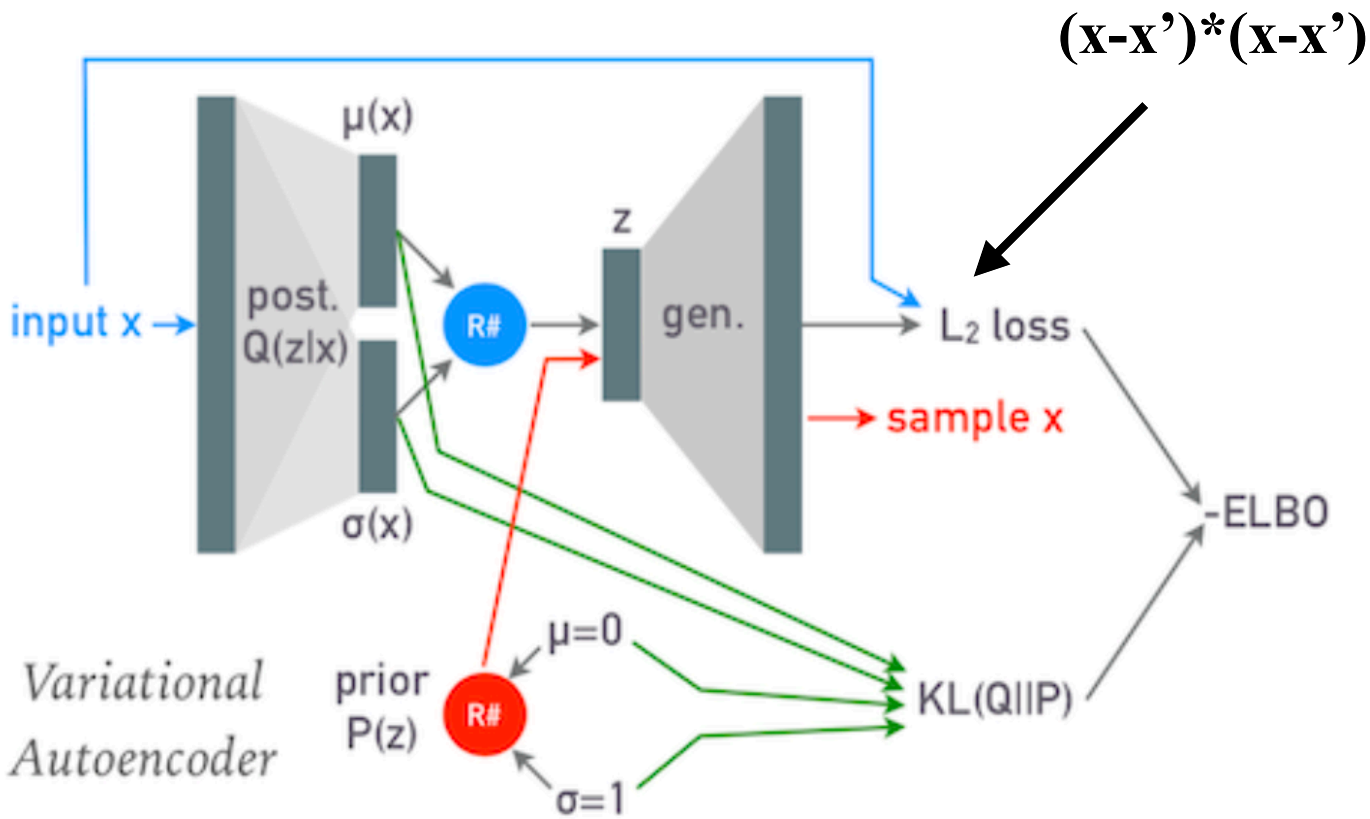
**WE WANT Q TO BE CLOSE TO THE PRIOR…**

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{\mathcal{X}} \log\left(\frac{dP}{dQ}\right) \frac{dP}{dQ}\, dQ,$$

**WE MINIMIZE THE K-L DIVERGENCE BETWEEN P AND Q**

# WHAT WOULD BE THEN THE LOSS FUNCTION OF A VAE?

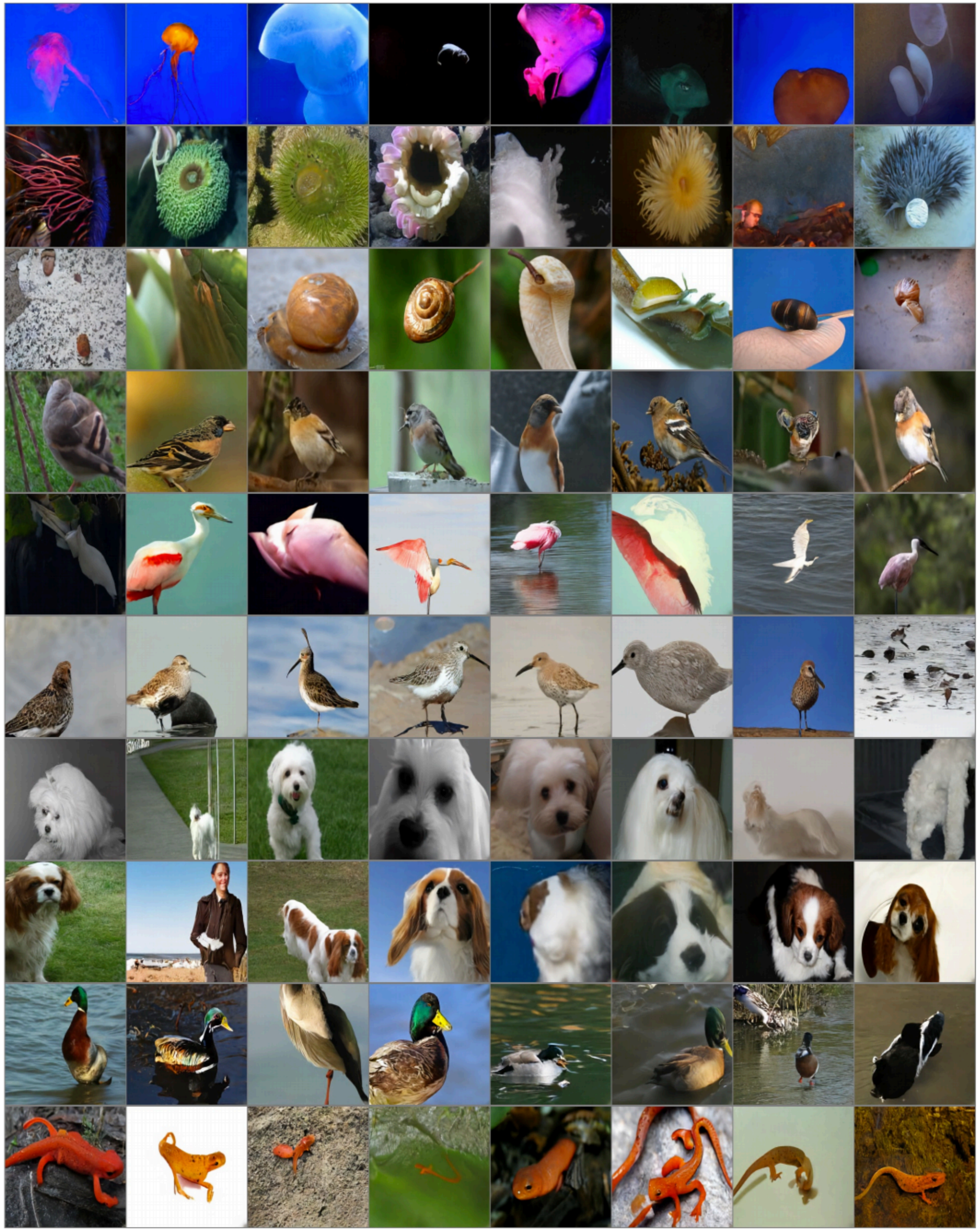



**The key insight of VAE is that we are actually performing <u>variational inference</u> here, which then tells us what the loss function should be…**

$$\boxed{-\text{ELBO} = \langle \log P(\mathbf{x} \mid \mathbf{z}) \rangle_{z\sim Q} + \text{KL}\left(Q(\mathbf{z};\Theta) \parallel P(\mathbf{z})\right) \ ,}$$

L2 LOSS

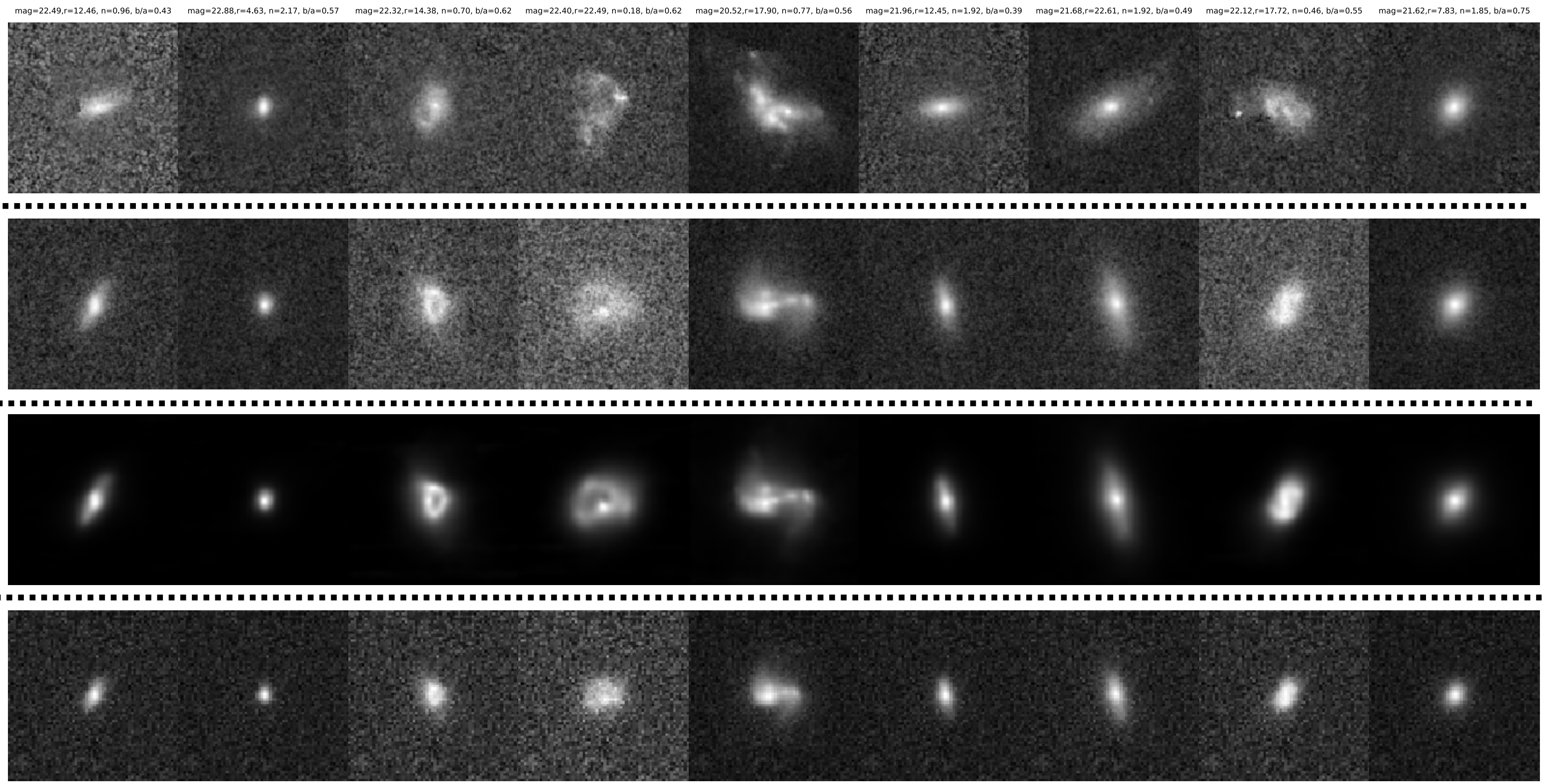
REGULARIZATION TERM



(VQ-VAE)

Razavi+19
(deepmind)

# VAE GENERATED EUCLID REALSITIC GALAXIES




**COSMOS GALAXY**

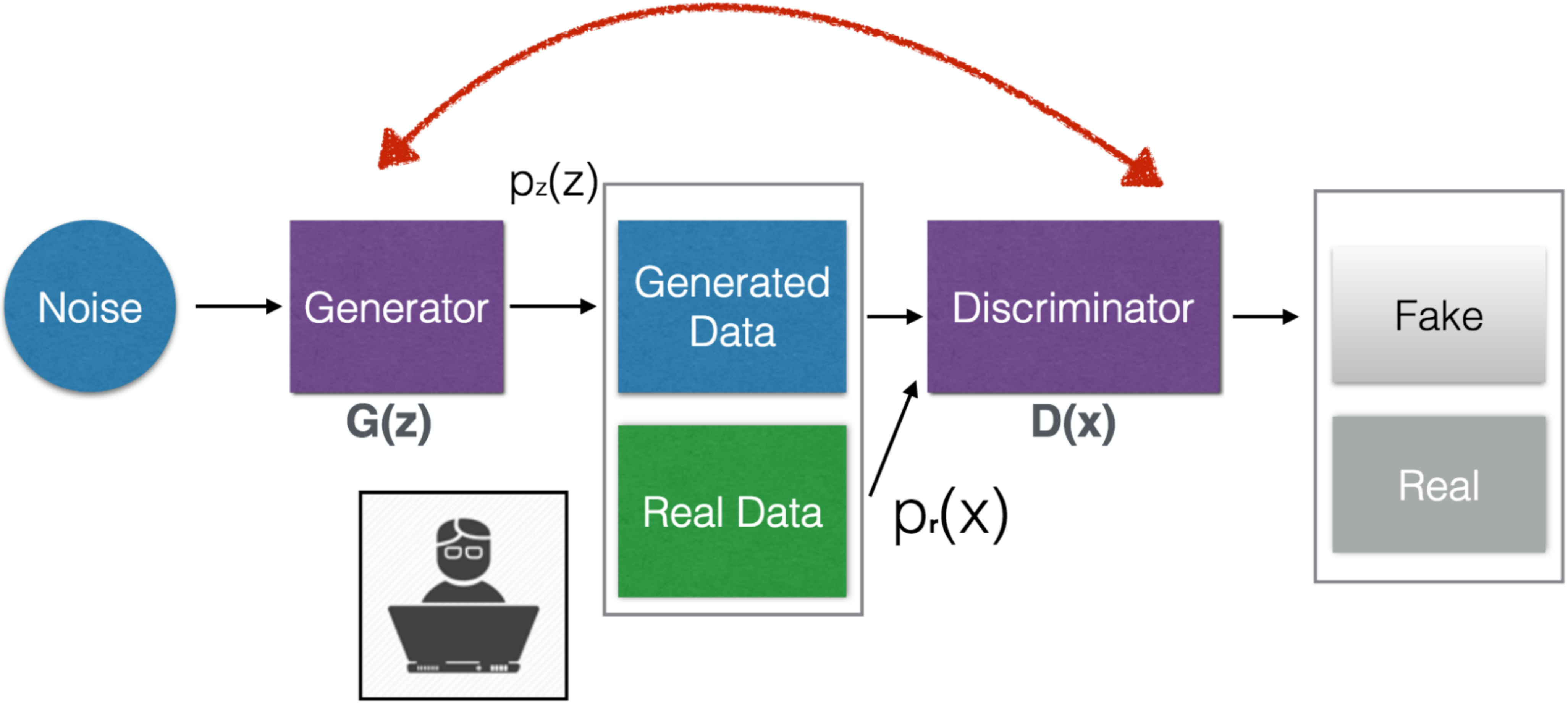
**“FAKE” COSMOS GALAXY**

**“FAKE” COSMOS GALAXY NO NOISE**

**“FAKE” EUCLID GALAXY NO NOISE**

# GENERATIVE ADVERSARIAL NETWORKS

**(Goodfellow+14)**

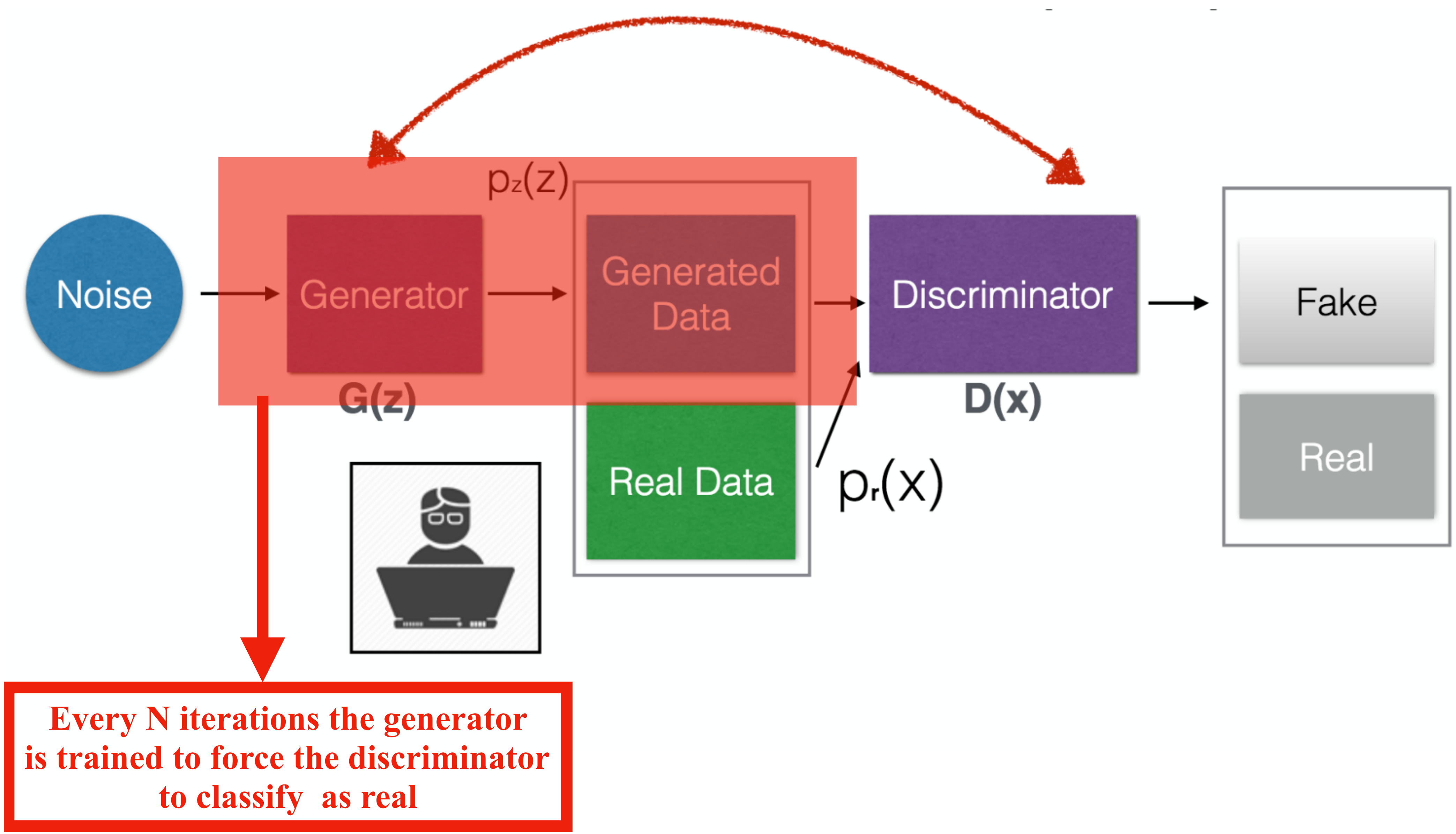



**TWO COMPETING NETWORKS**

# GENERATIVE ADVERSARIAL NETWORKS

**(Goodfellow+)**

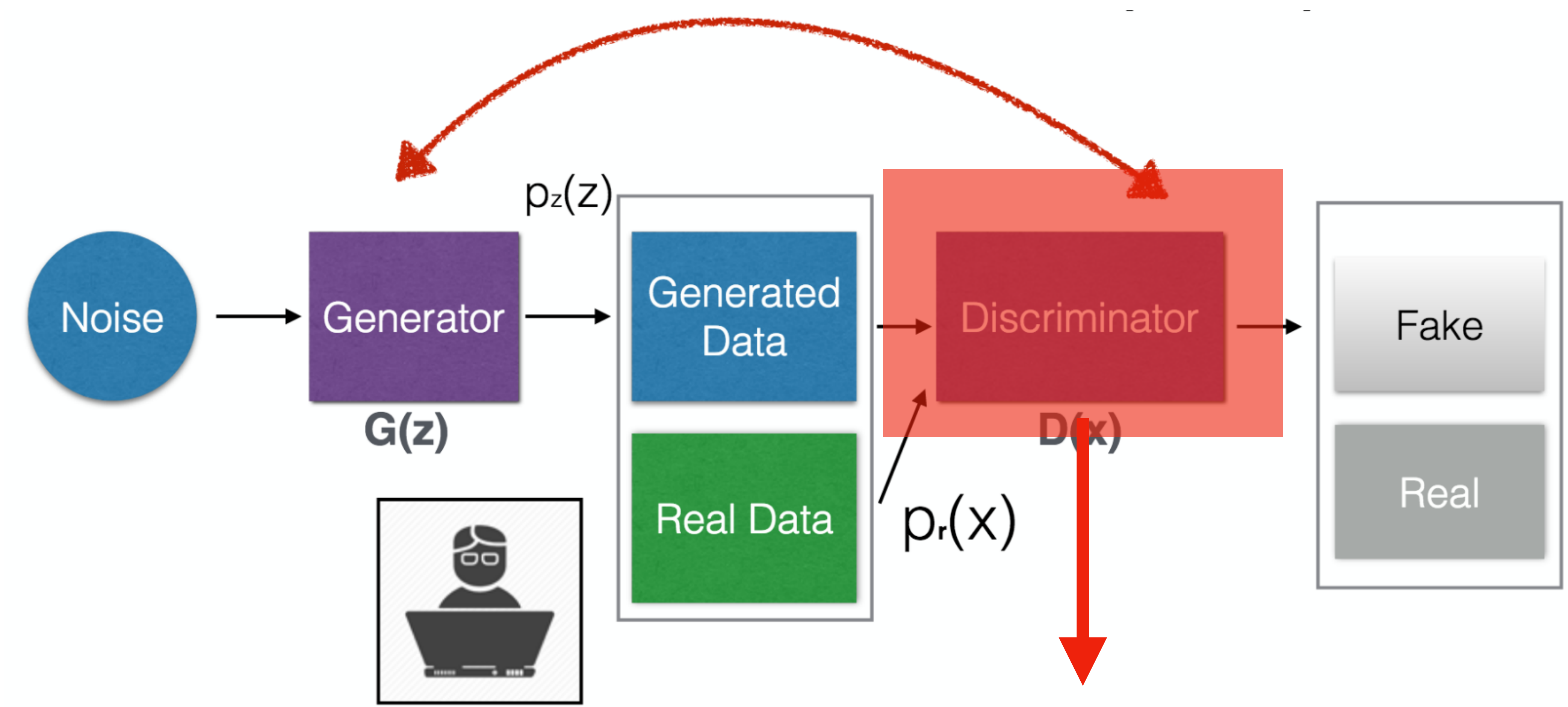
## <u>TWO COMPETING NETWORKS</u>



# GENERATIVE ADVERSARIAL NETWORKS

(Goodfellow+)

## TWO COMPETING NETWORKS

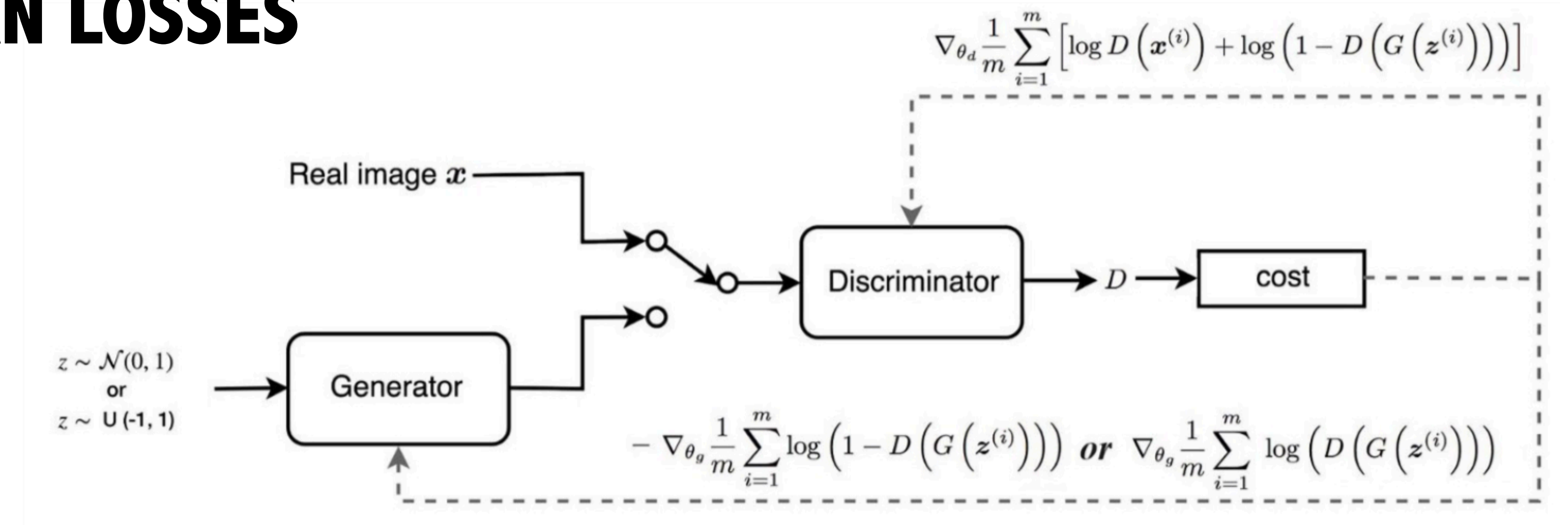



**Every N iterations the discriminator is trained to force to distinguish between real and fake**

# IN PRACTICE

## GAN LOSSES

**DISCRIMINATOR LOSS (CROSS-ENTROPY)**




**GENERATOR LOSS**

# GANs CAN ACHIEVE IMPRESSIVE RESULTS…


Karras+19

## THEY ARE ALSO VERY HARD TO TRAIN

# 1. IT IS HARD TO REACH EQUILIBRIUM AND IT IS ACTUALLY NOT GUARANTEED...

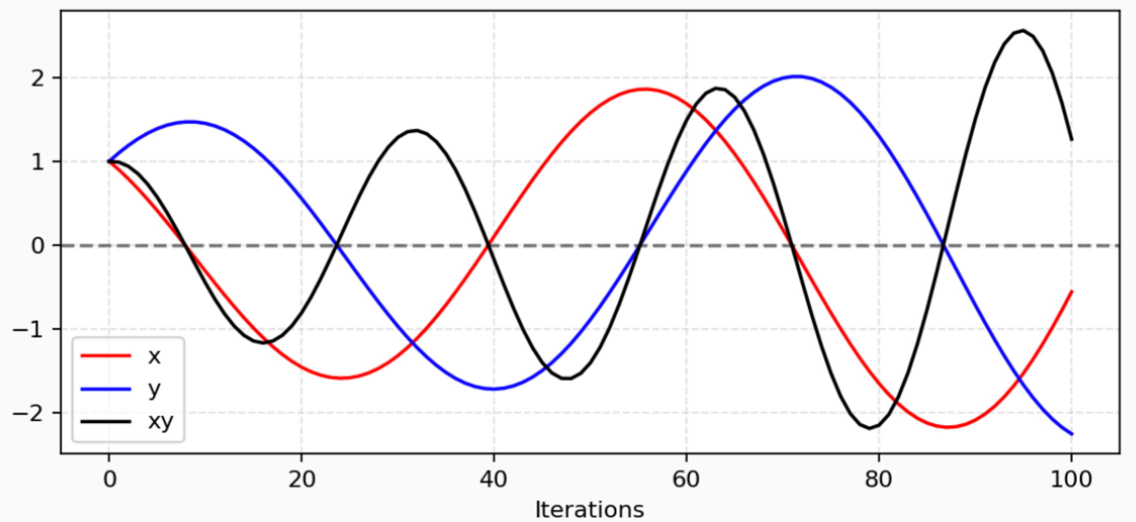



*Fig. 3. A simulation of our example for updating $x$ to minimize $xy$ and updating $y$ to minimize $-xy$. The learning rate $\eta = 0.1$. With more iterations, the oscillation grows more and more unstable.*

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

SALIMANS+16

# THEY ARE ALSO VERY HARD TO TRAIN

## 2. LOW DIMENSIONAL SUPPORTS

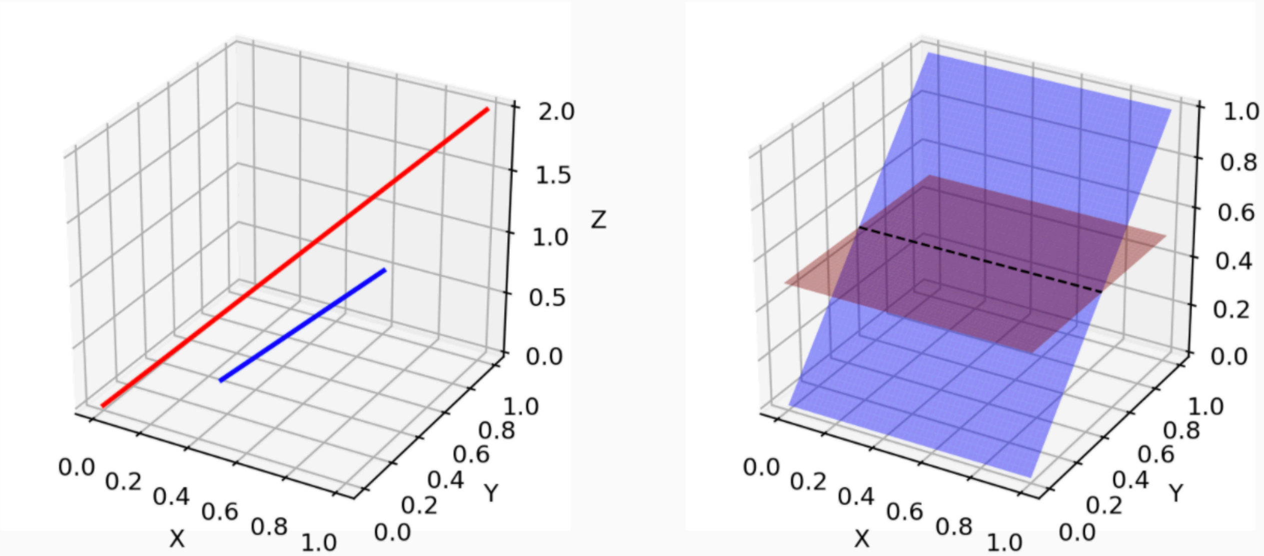



*Fig. 4. Low dimensional manifolds in high dimension space can hardly have overlaps. (Left) Two lines in a three-dimension space. (Right) Two surfaces in a three-dimension space.*

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

ARJOVSKI+17

# THEY ARE ALSO VERY HARD TO TRAIN

## 3. MODE COLLAPSE

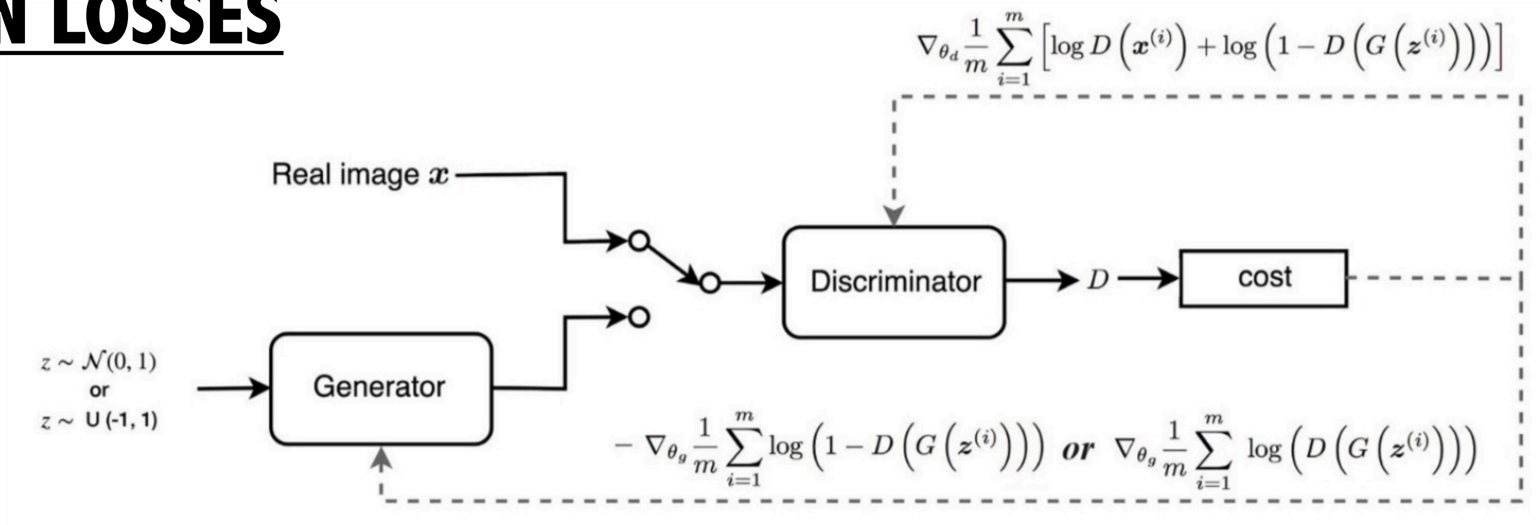
https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

razavi+19

# IN PRACTICE

## GAN LOSSES

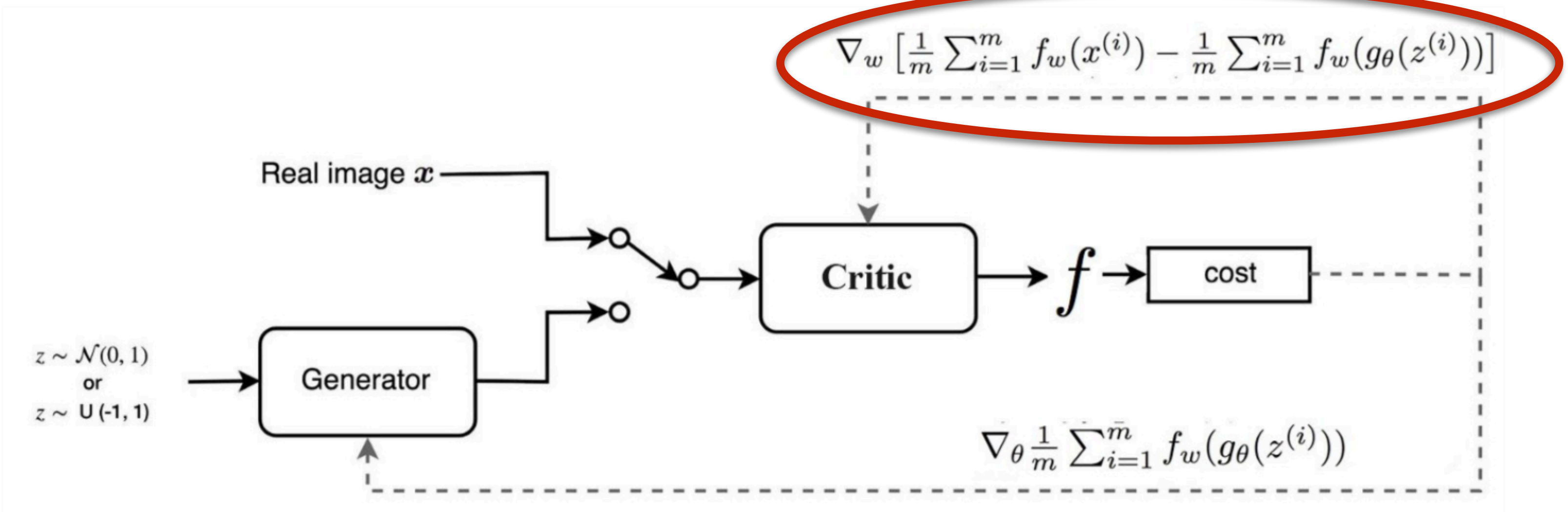
**DISCRIMINATOR LOSS (CROSS-ENTROPY)**




**GENERATOR LOSS**

## WASSERSTEIN GAN LOSSES

K-L DIvergence is replaced by Wasserstein distance

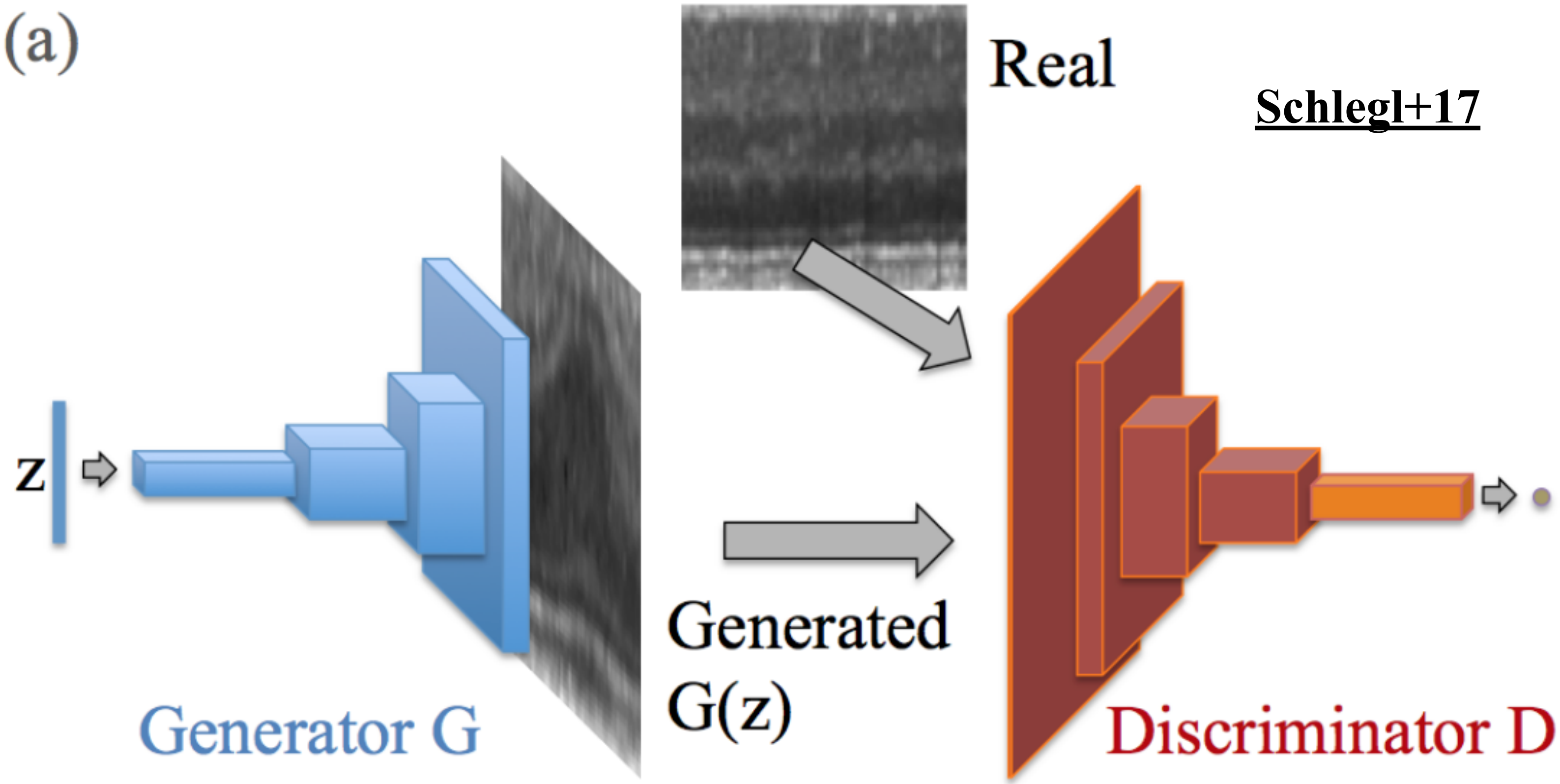


A USEFUL APPLICATION (BESIDES GENERATING NICE IMAGES) IS ESTIMATING THE PROBABILITY DENSITY OF AN ARBITRARY INPUT, RELATIVE TO THE INPUT DISTRIBUTION

# ANOMALY DETECTION IN ASTRONOMY

- FUTURE BIG-DATASETS WILL BE PROCESSED THROUGH AUTOMATED (ML) METHODS - MOST OF THE DATA WILL NEVER BE LOOKED BY HUMANS
- UNKNOWN UNKNOWNS IS WHERE INTERESTING (NEW) SCIENCE WILL BE FOUND
- EFFICIENT ANOMALY DETECTION IS CRUCIAL TO UNLOCK THE DISCOVERY POTENTIAL OF FUTURE SURVEYS

# astroGANomaly: ANOMALY DETECTION WITH GANs




**USE WGANs TO LEARN P(X) [NORMAL DATA]**

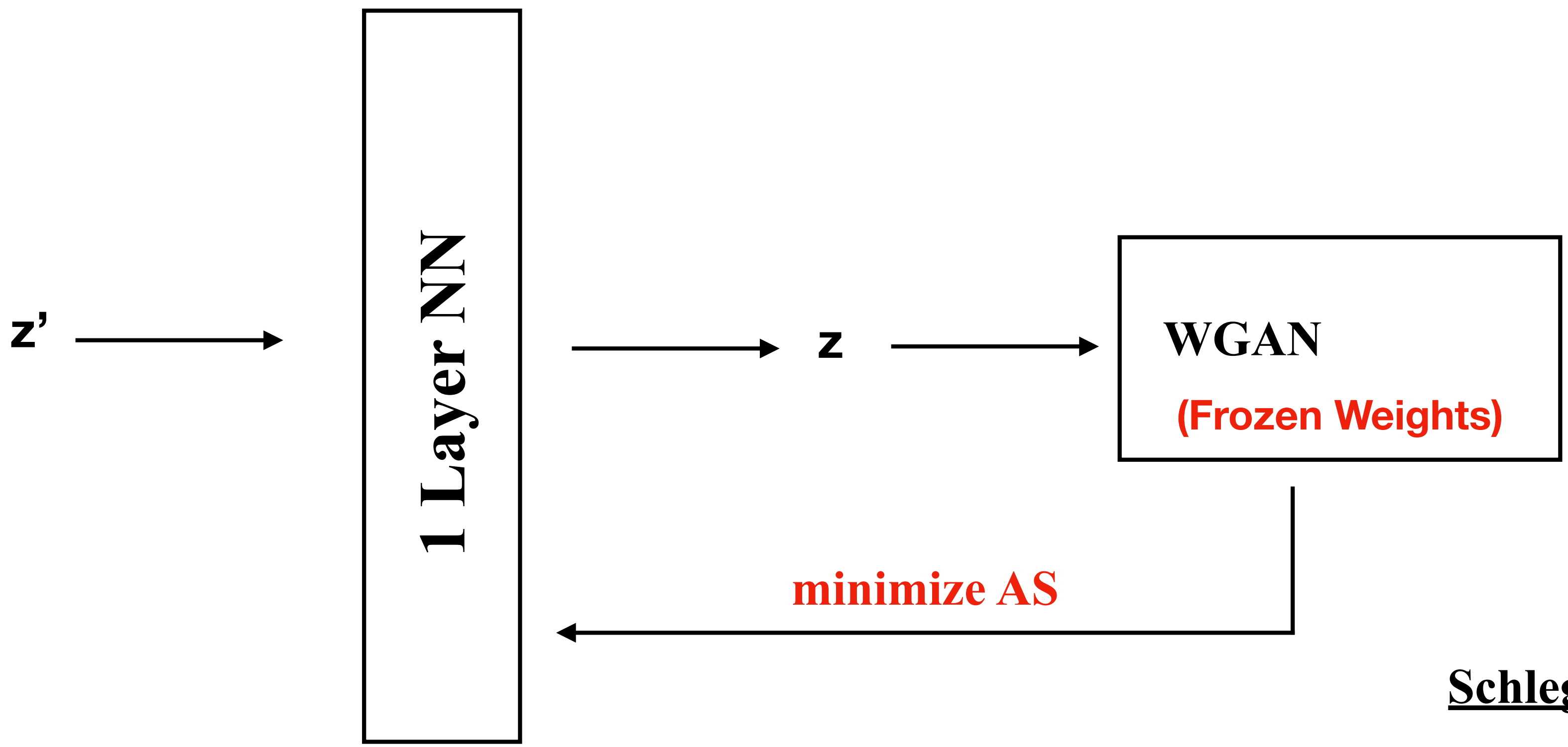
# COMPUTE ANOMALY SCORE BASED ON WGAN RECONSTRUCTION ERROR

**ANOMALY SCORE:** $$AS = \lambda G + (1 - \lambda) C$$

**Hyperparameter**

**RMS btw input and generated**

**RMS btw critic features from input and generated**



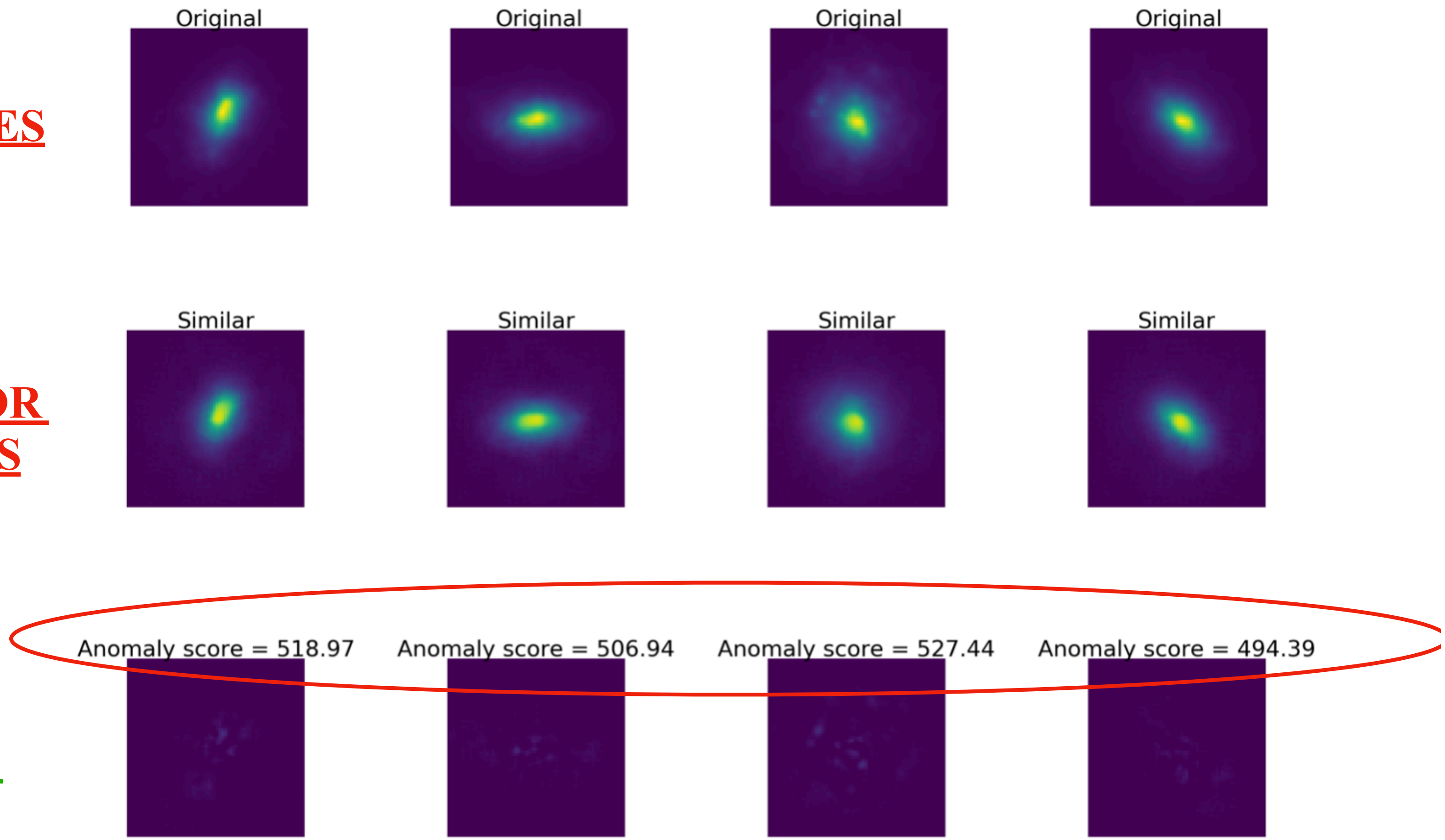

**Schlegl+17**

# EXAMPLE OF RECONSTRUCTION

“NORMAL” GALAXIES

GAN BEST GUESS FOR NORMAL GALAXIES

RESIDUAL MAPS

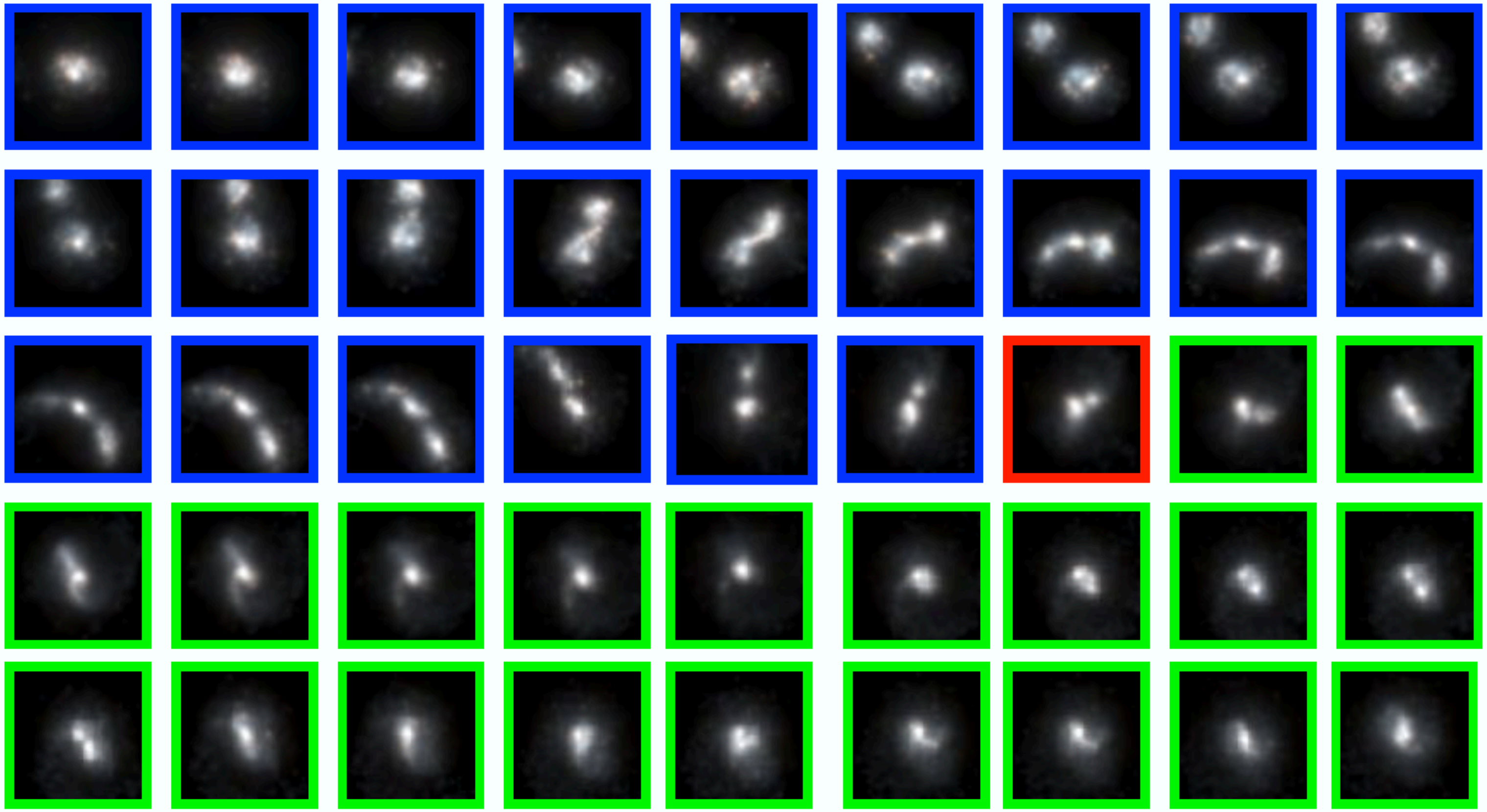



**HORIZON-AGN SIMULATION; DUBOIS+14**

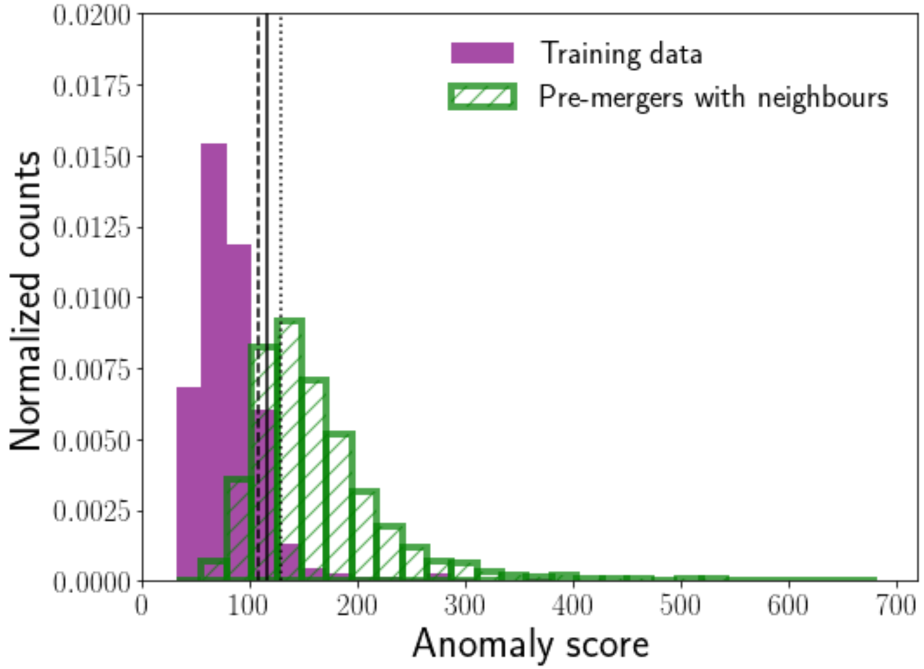
# A TEST CASE OF "KNOWN ANOMALIES": GALAXY MERGERS

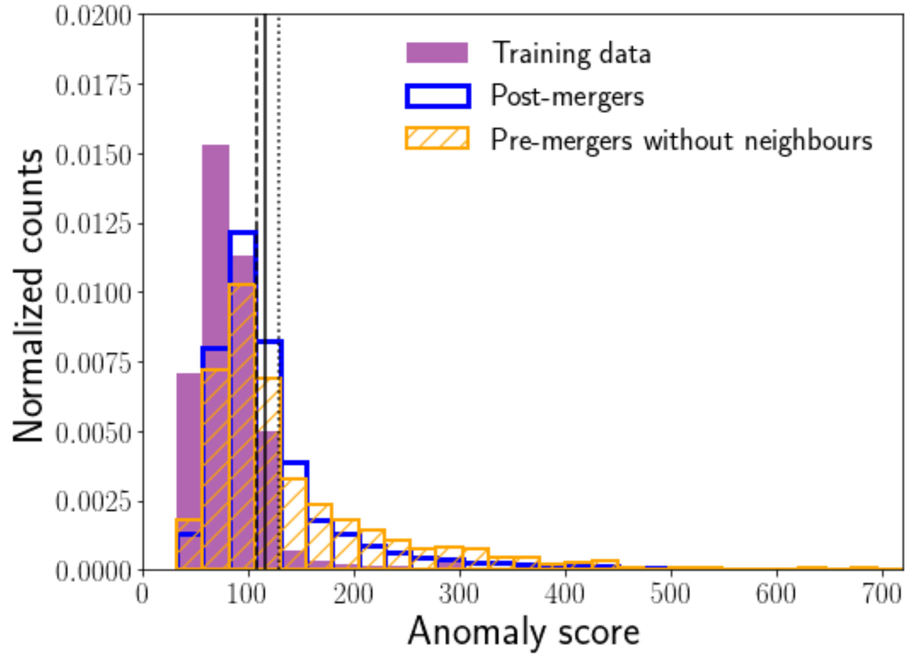
~20.000 major merger sequences

~1 million merger images



PRE-MERGER / MERGER / POST-MERGER



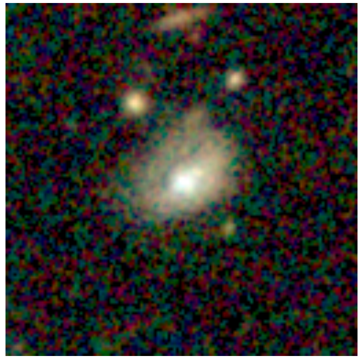




| Percentage of inconsistent images | | |
|---|---|---|
| Sample | WGAN | k-means |
| All | 46% | 8% |
| Pre-mergers | 86% | 74% |
| Post-mergers | 45% | 9% |

# LEARN “NORMAL” DATA WITH GENERATIVE MODELS

x

→ **GENERATIVE MODEL** →

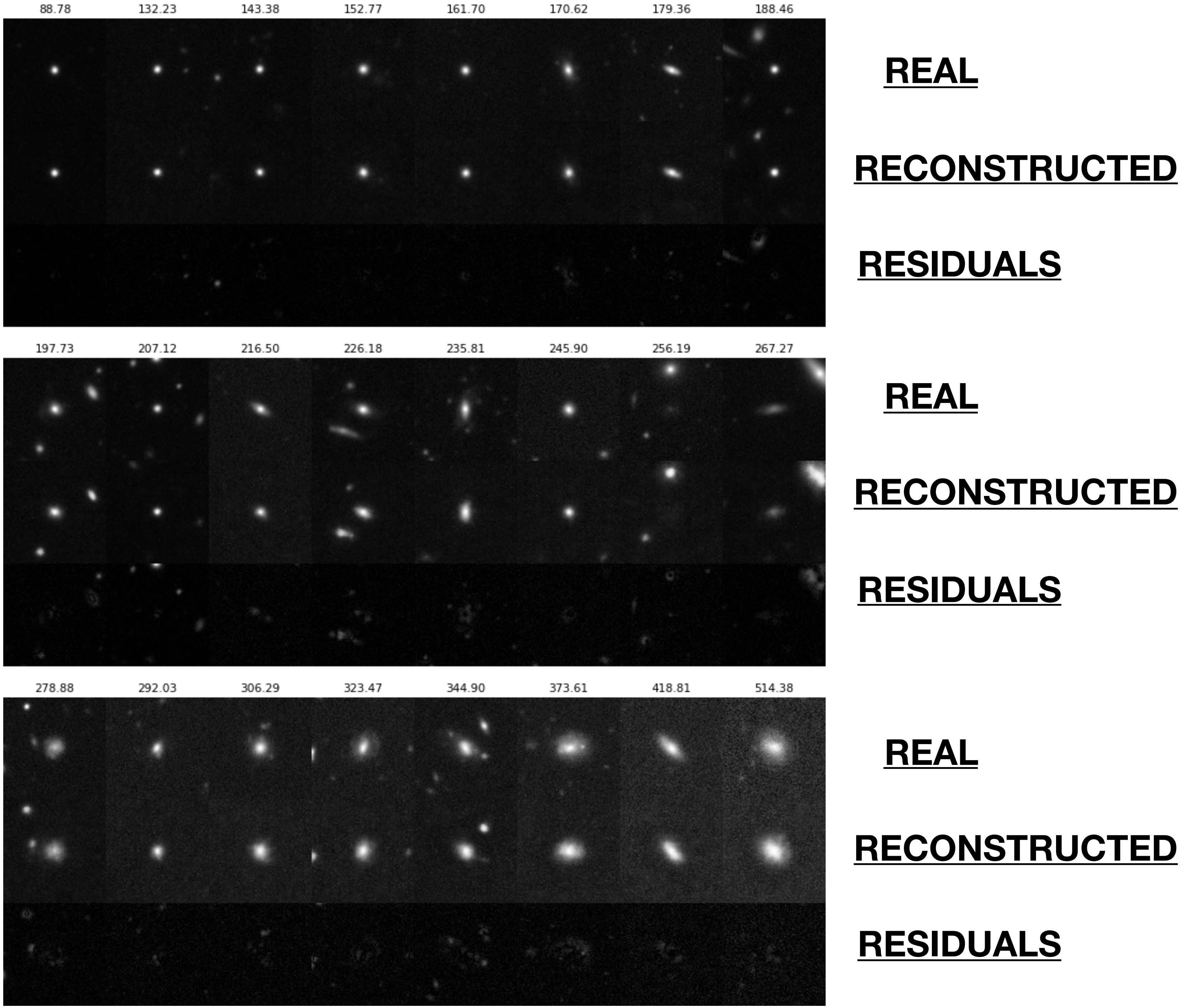


x’

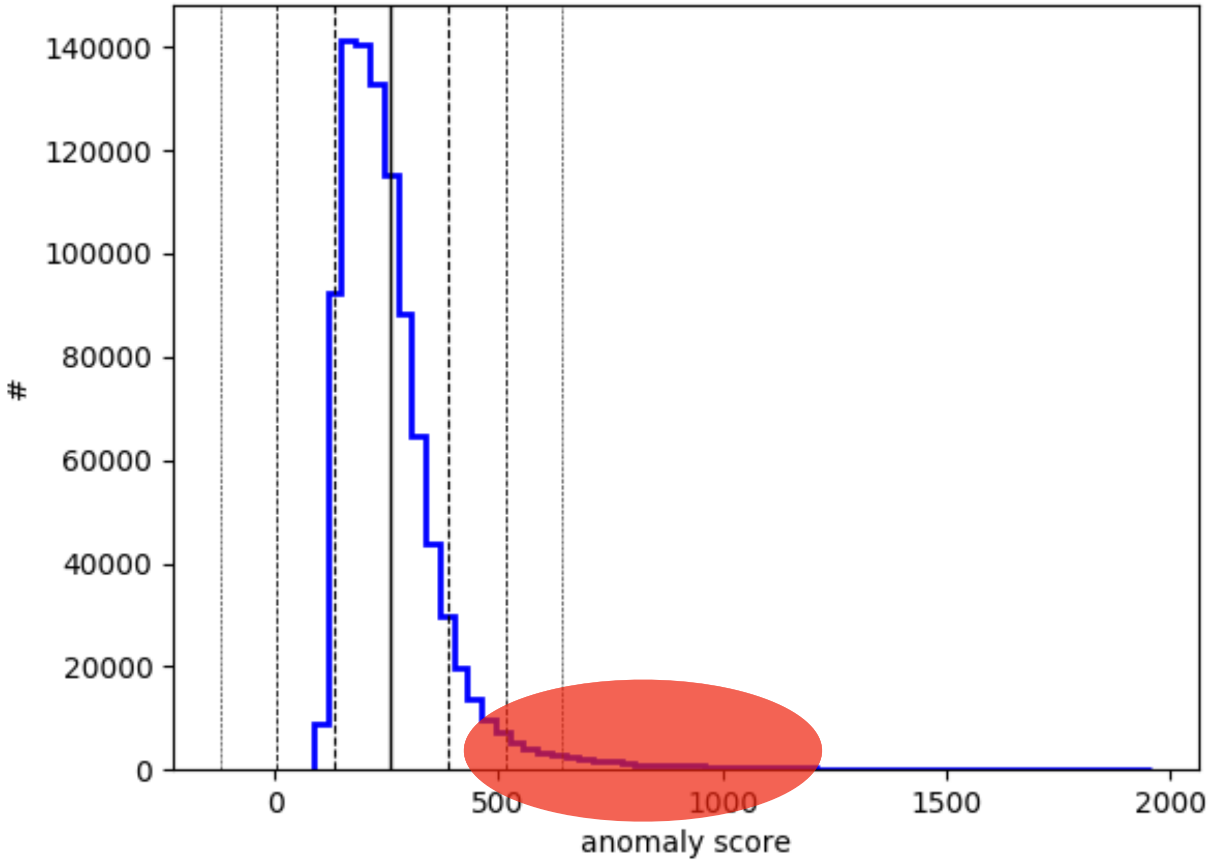
HSC Hyper Suprime-Cam Subaru Strategic Program

SURVEY

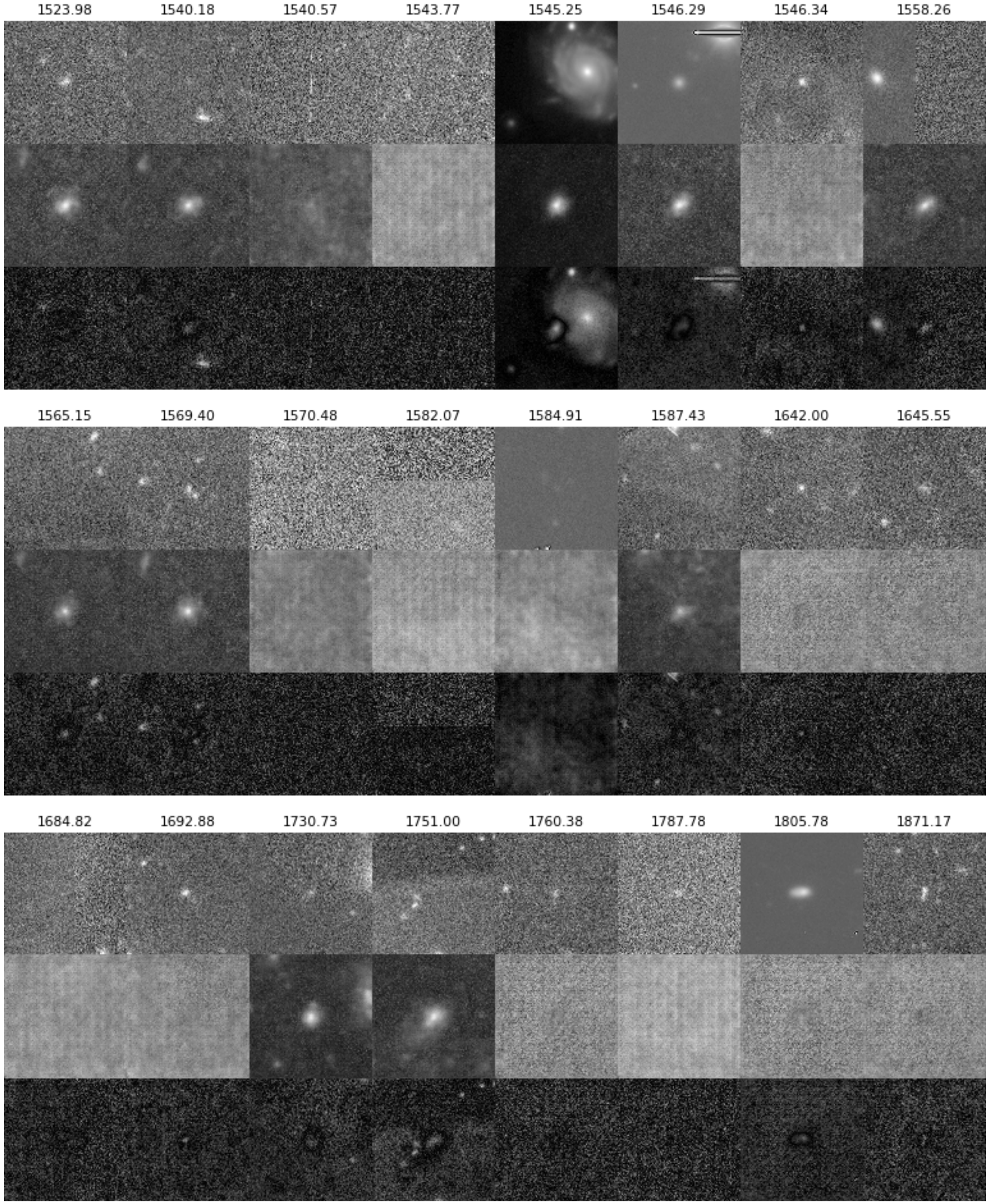
| Layer | Area (deg$^2$) | # of 1.8deg$^2$ HSC fields | Filters & Depth |
|---|---|---|---|
| Wide | 1400 | 916 | *grizy*(*r* ~ 26) |
| Deep | 27 | 15 | *grizy*+4NBs (*r* ~ 27) |
| Ultradeep | 3.5 | 2 | *grizy*+4NBs (*r* ~ 28) |


88.78
132.23
143.38
152.77
161.70
170.62
179.36
188.46
REAL
RECONSTRUCTED
RESIDUALS
197.73
207.12
216.50
226.18
235.81
245.90
256.19
267.27
REAL
RECONSTRUCTED
RESIDUALS
278.88
292.03
306.29
323.47
344.90
373.61
418.81
514.38
REAL
RECONSTRUCTED
RESIDUALS

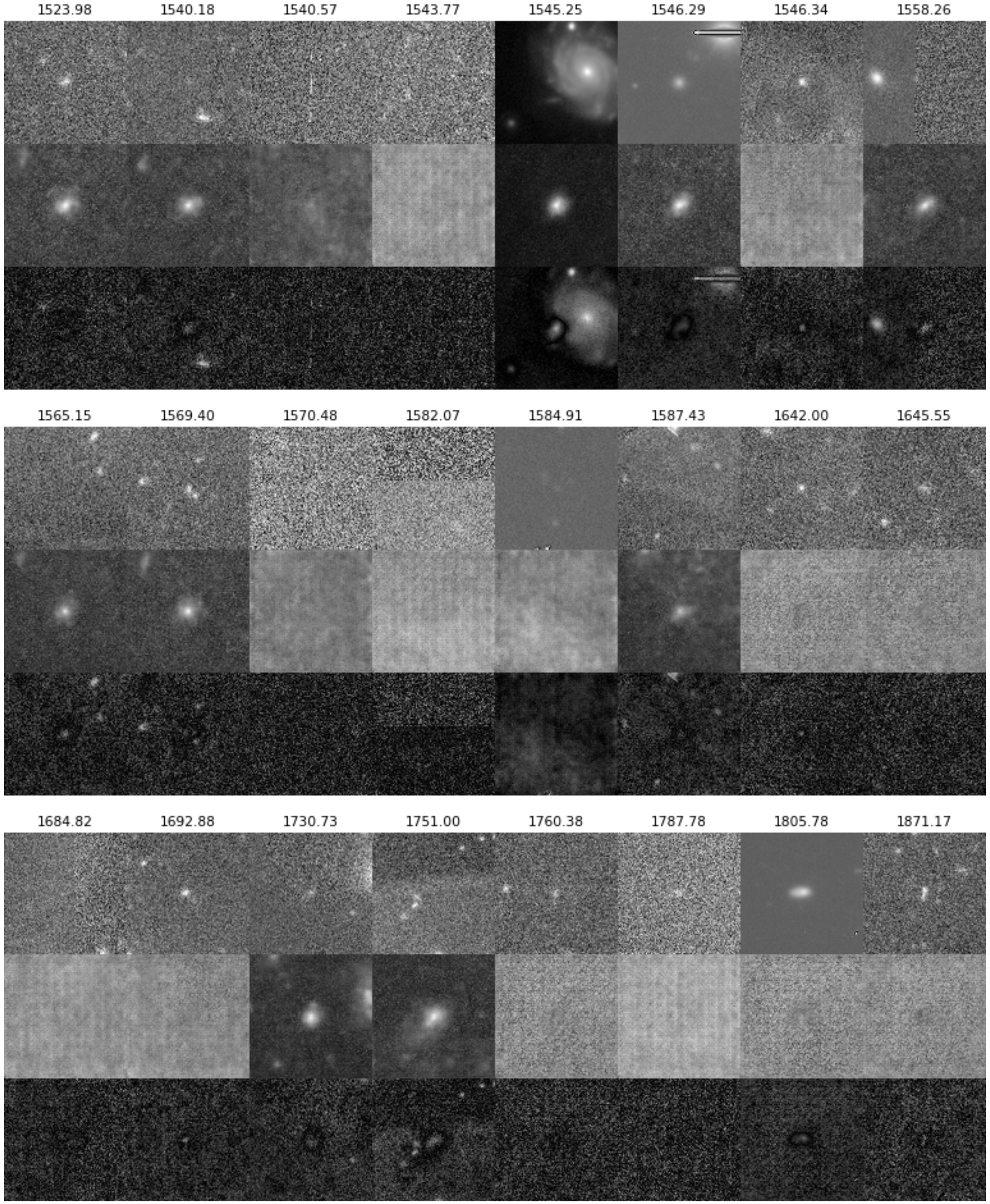
# “ANOMALY SCORE” DISTRIBUTION FOR HSC GALAXIES

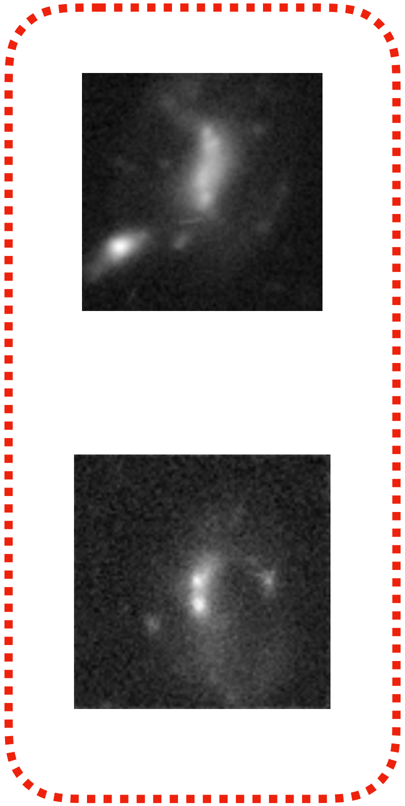


# LARGER ANOMALIES ARE TYPICALLY PIPELINE ERRORS….



# LARGER ANOMALIES ARE TYPICALLY PIPELINE ERRORS….



# PROBABILISTIC MODELS

- Estimate the probability density of an arbitrary input, relative to the input distribution

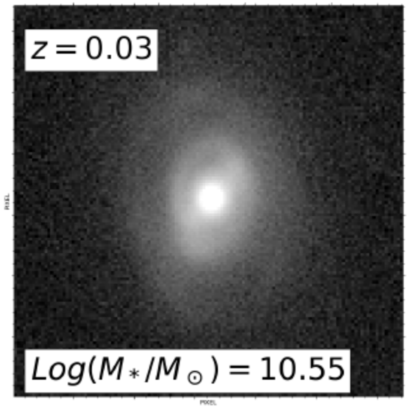
# GANs AND VAEs ARE VERY POWERFUL BUT DO NOT PROVIDE AN EXPLICIT LIKELIHOOD

| | Method | Train on data | One-pass Sampling | Exact log-likelihood | Free-form Jacobian |
|---|---|---|---|---|---|
| | Variational Autoencoders | ✓ | ✓ | ✗ | ✓ |
| | Generative Adversarial Nets | ✓ | ✓ | ✗ | ✓ |
| | Likelihood-based Autoregressive | ✓ | ✗ | ✓ | ✗ |
| Change of Variables | Normalizing Flows | ✗ | ✓ | ✓ | ✗ |
| | Reverse-NF, MAF, TAN | ✓ | ✗ | ✓ | ✗ |
| | NICE, Real NVP, Glow, Planar CNF | ✓ | ✓ | ✓ | ✗ |
| | **FFJORD** | ✓ | ✓ | ✓ | ✓ |

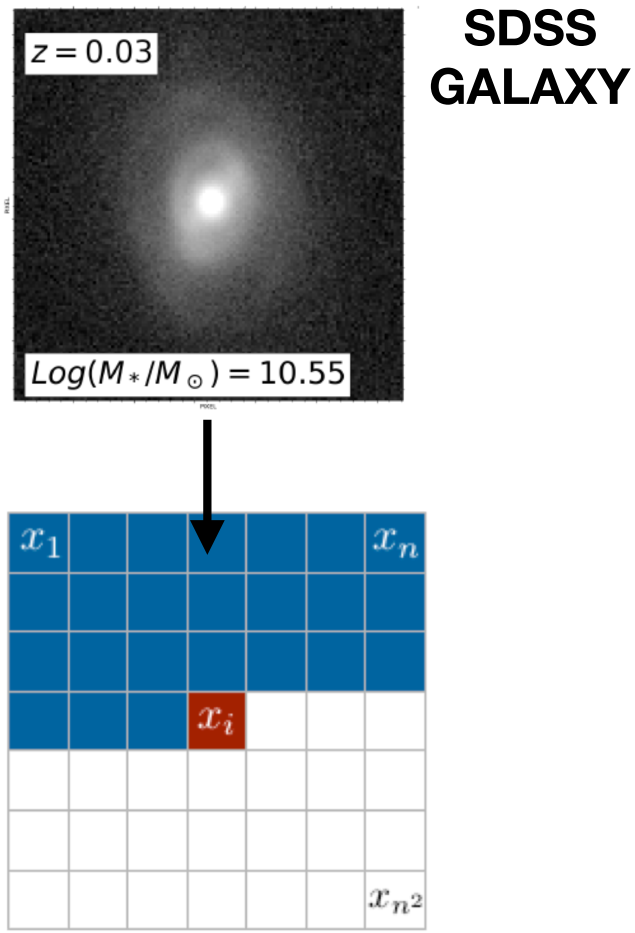
Table 1: A comparison of the abilities of generative modeling approaches.

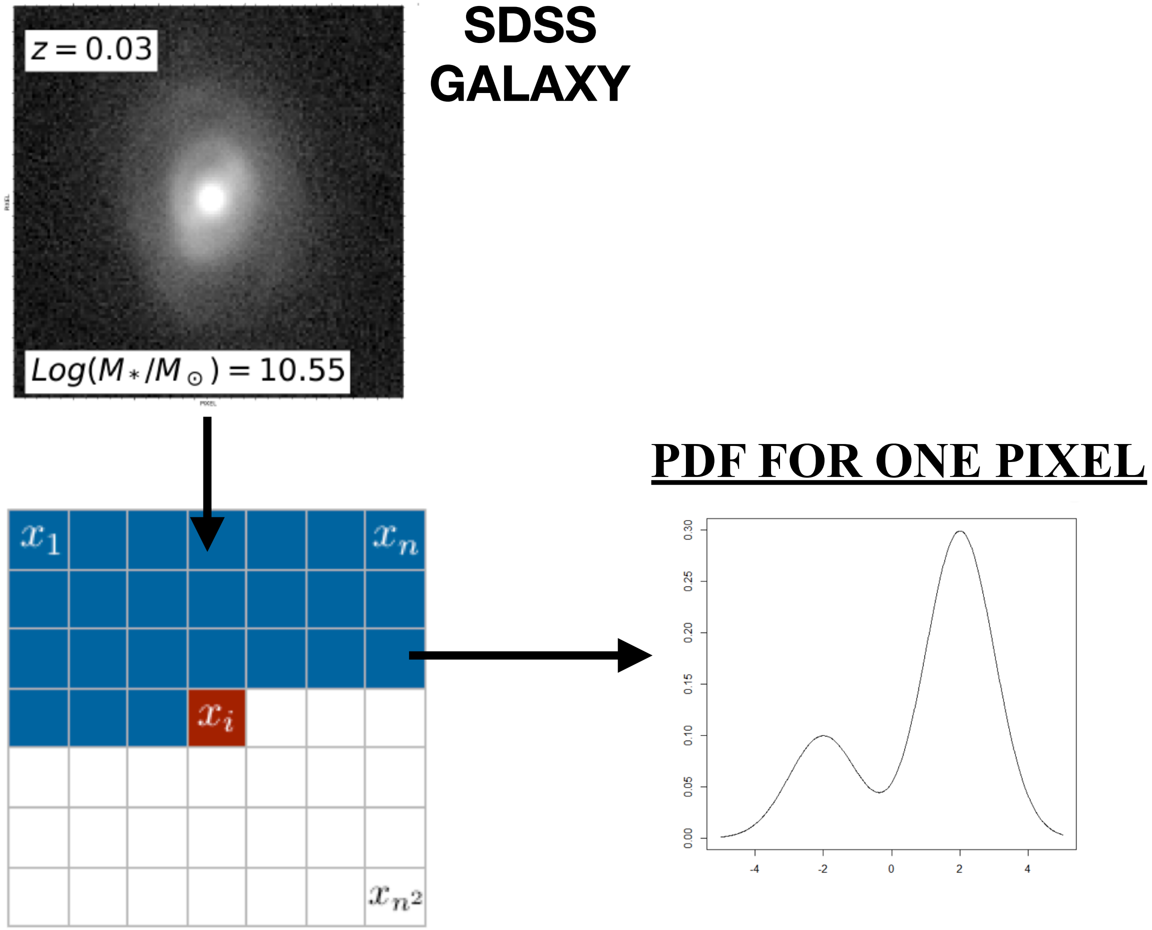
**Grathwohl+18**

**MAIN CAVEAT: INFERENCE IS VERY SLOW!**

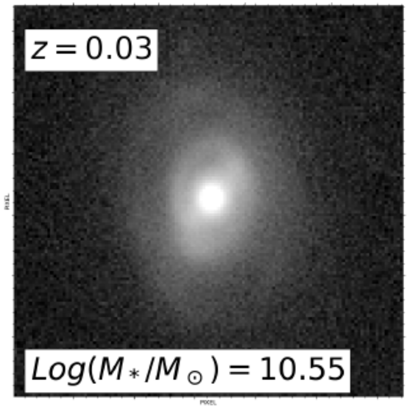


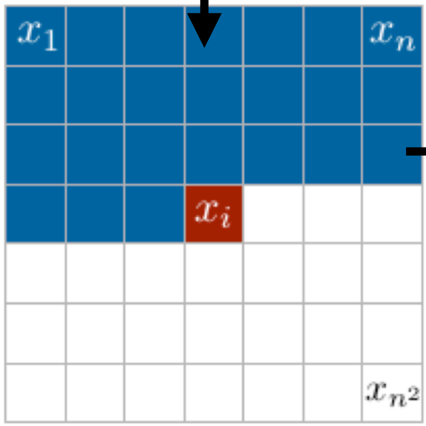

**SDSS GALAXY**


SDSS
GALAXY
$z = 0.03$
$Log(M_*/M_\odot) = 10.55$
$x_1$
$x_n$
$x_i$
$x_{n^2}$

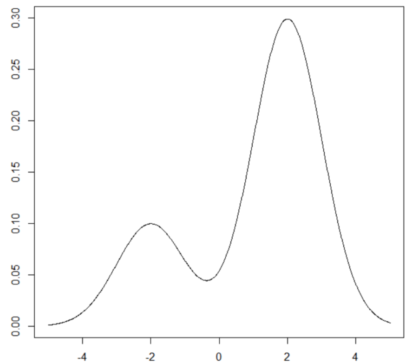

SDSS
GALAXY
z = 0.03
Log(M*/M⊙) = 10.55
x1
xn
xi
xn2
PDF FOR ONE PIXEL
0.00
0.05
0.10
0.15
0.20
0.25
0.30
-4
-2
0
2
4

**SDSS GALAXY**

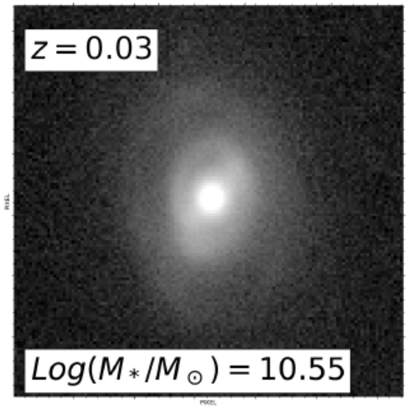


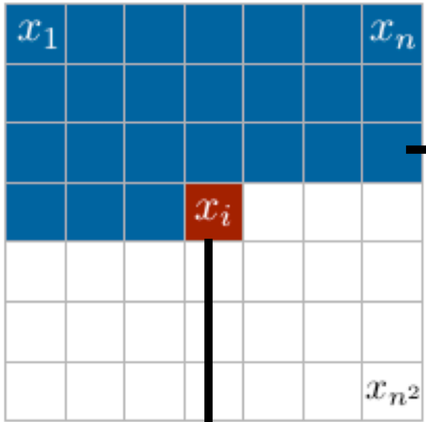

**PDF FOR ONE PIXEL**



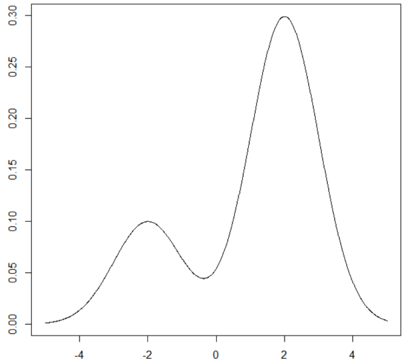

$$\boxed{p(x) = p(x_0, x_1, ..., x_{n^2} | \theta_{SDSS})}$$




**SDSS GALAXY**



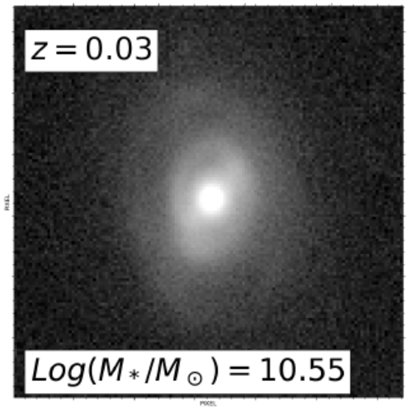

**PDF FOR ONE PIXEL**



$$p(x) = p(x_0, x_1, ..., x_{n^2} | \theta_{SDSS})$$

$$p(x_i | x_1, ..., x_{i-1})$$

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

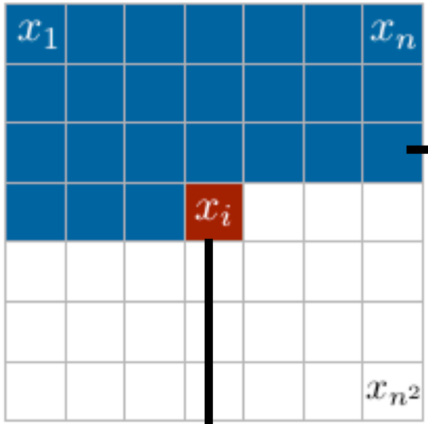


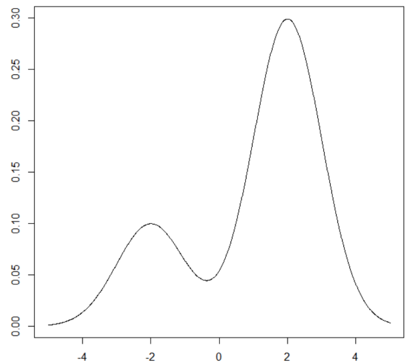

**SDSS GALAXY**

# AUTOREGRESSIVE IMAGE GENERATION: pixelCNN

[van der Oord+16, Salimans+17]

## PDF FOR ONE PIXEL



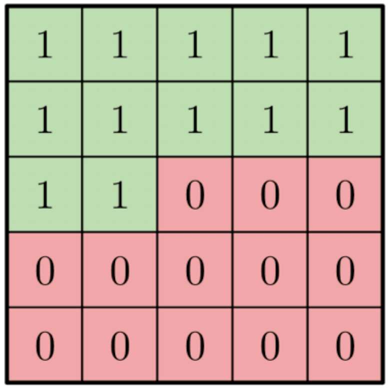

$$p(x) = p(x_0, x_1, \dots, x_{n^2} | \theta_{SDSS})$$

$$p(x_i | x_1, \dots, x_{i-1})$$

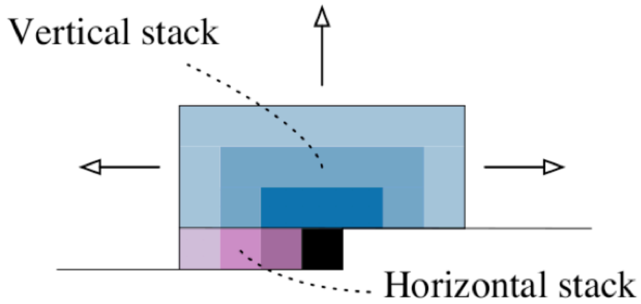
$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, \dots, x_{i-1})$$

# IN PRACTICE….

## Masked convolutions





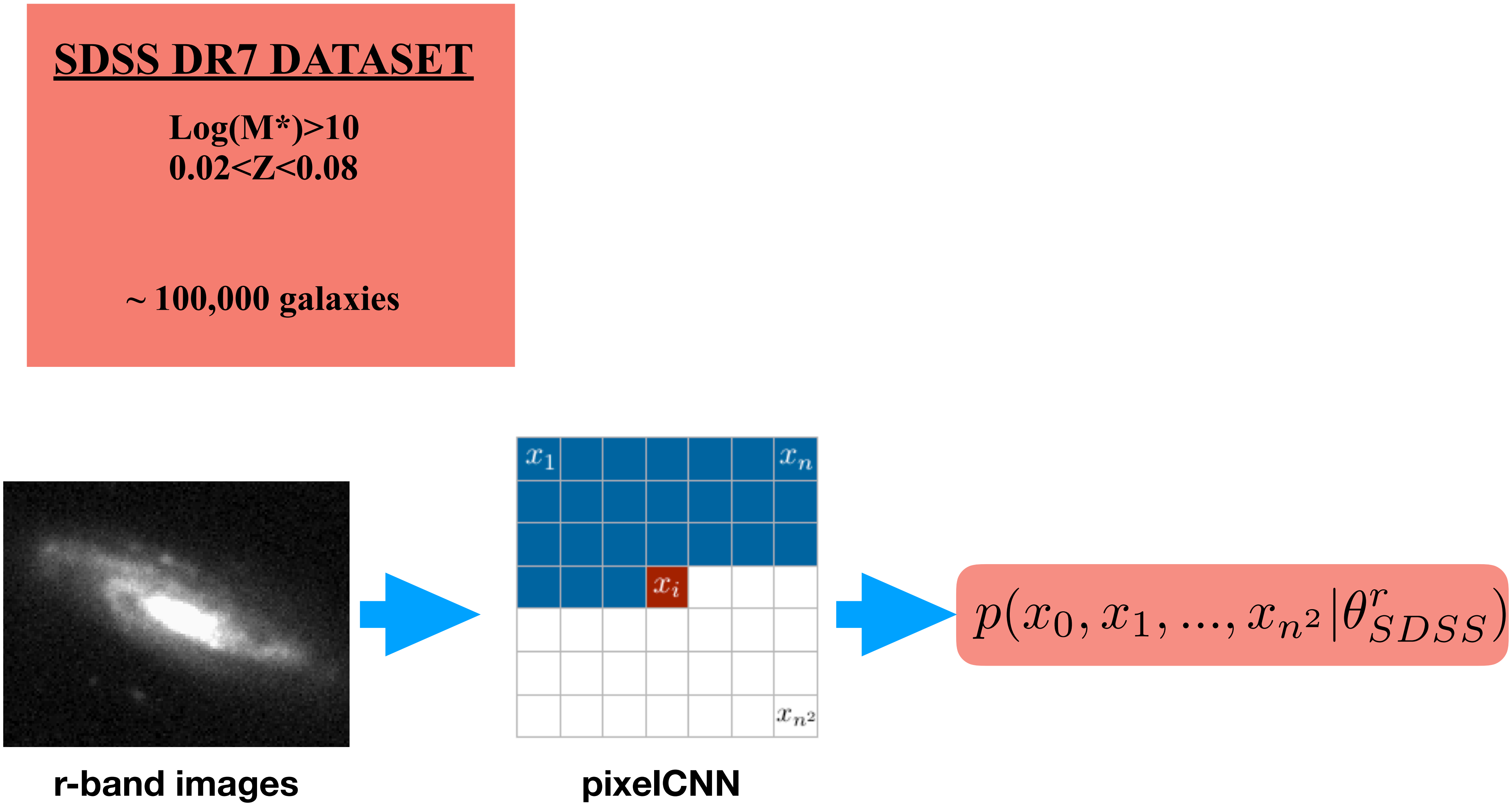

Makes learning conditioned only on previous pixels

This implementation gives a “blind spot”

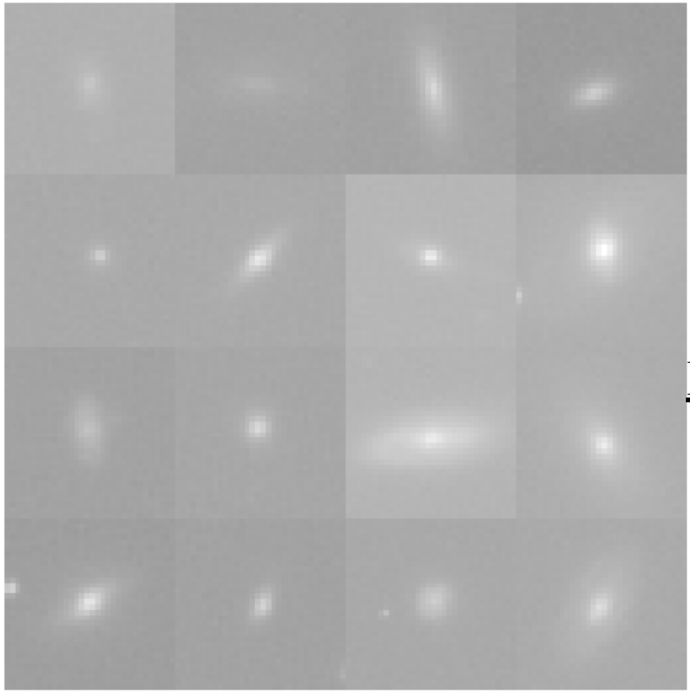
Solution: sum two rectangular convolutions

# COMPARING SIMULATIONS OF GALAXY FORMATION WITH OBSERVATIONS WITH REGRESSIVE FLOWS

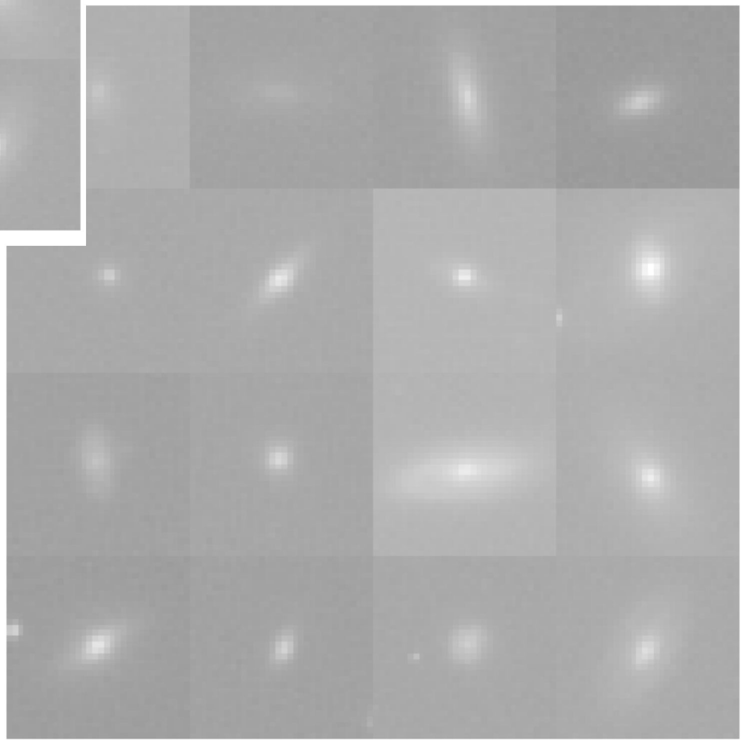
***DOES A NEURAL NETWORK KNOW ABOUT HORSES IF IT HAS ONLY SEEN CATS AND DOGS?***


SDSS DR7 DATASET
Log(M*)>10
0.02<Z<0.08
~ 100,000 galaxies
$x_1$
$x_n$
$x_i$
$x_{n^2}$
r-band images
pixelCNN
$p(x_0, x_1, ..., x_{n^2} | \theta^r_{SDSS})$

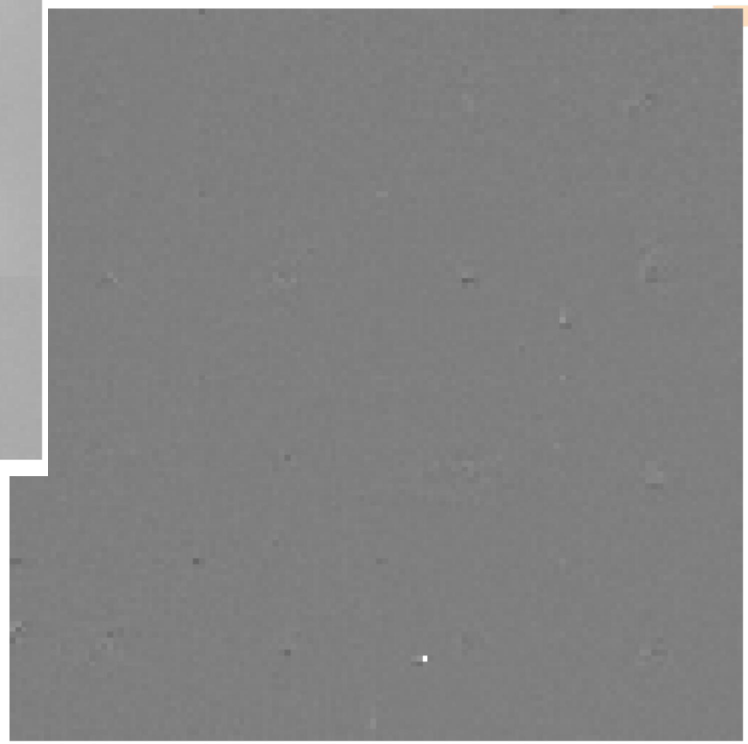
**ORIGINAL SDSS**



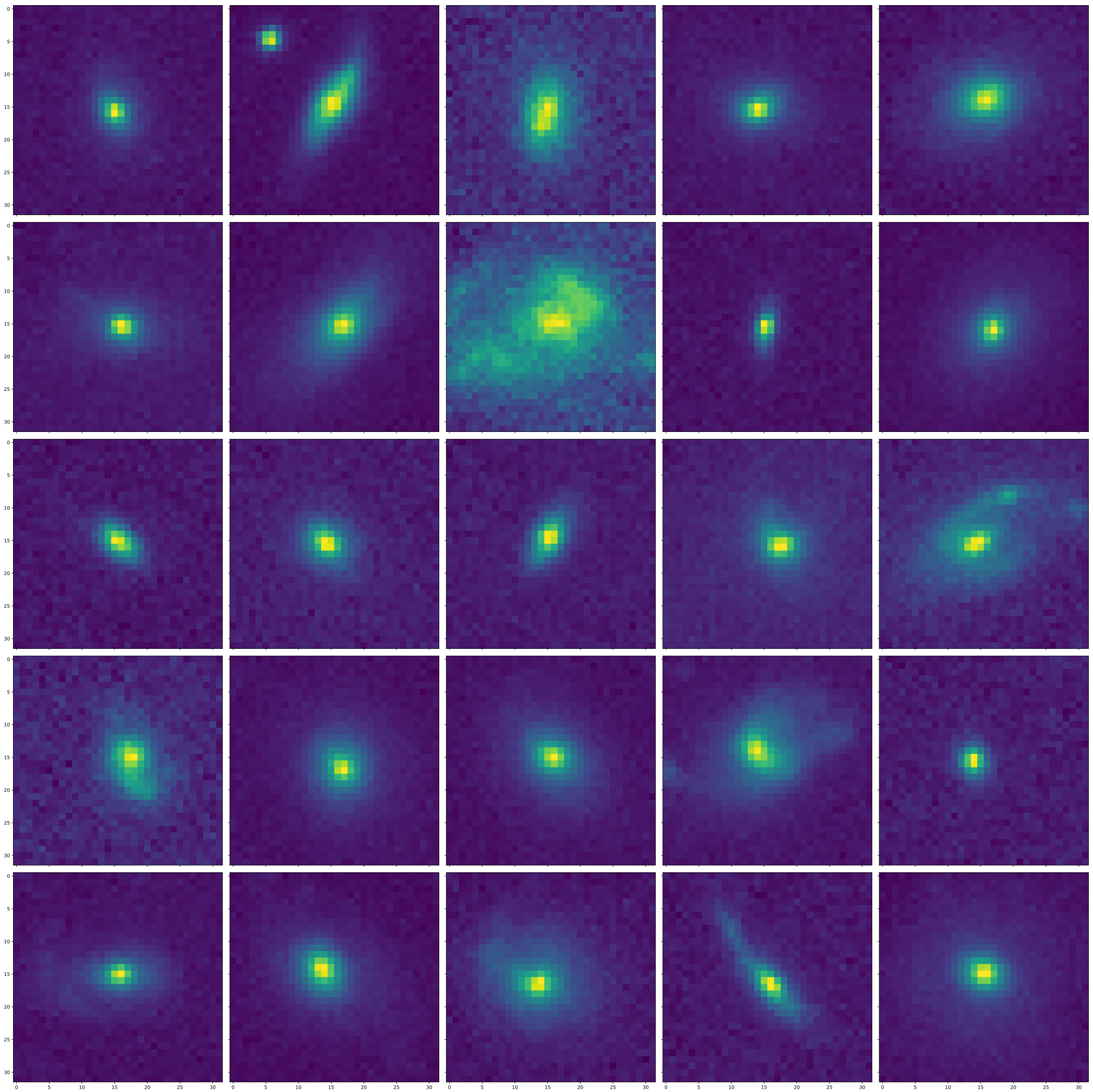
**RECONSTRUCTED WITH PIXELCNN**



**RESIDUALS**

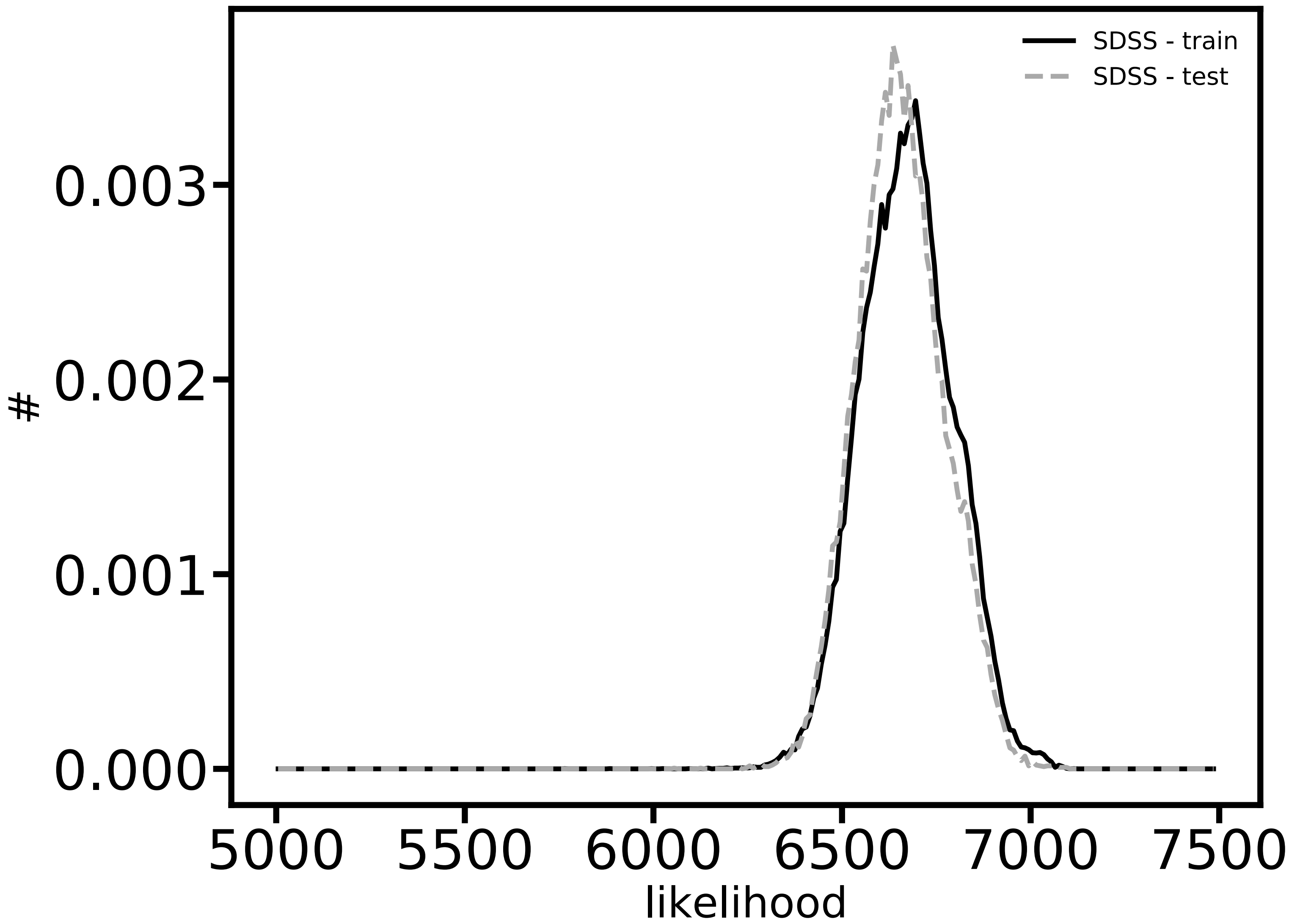


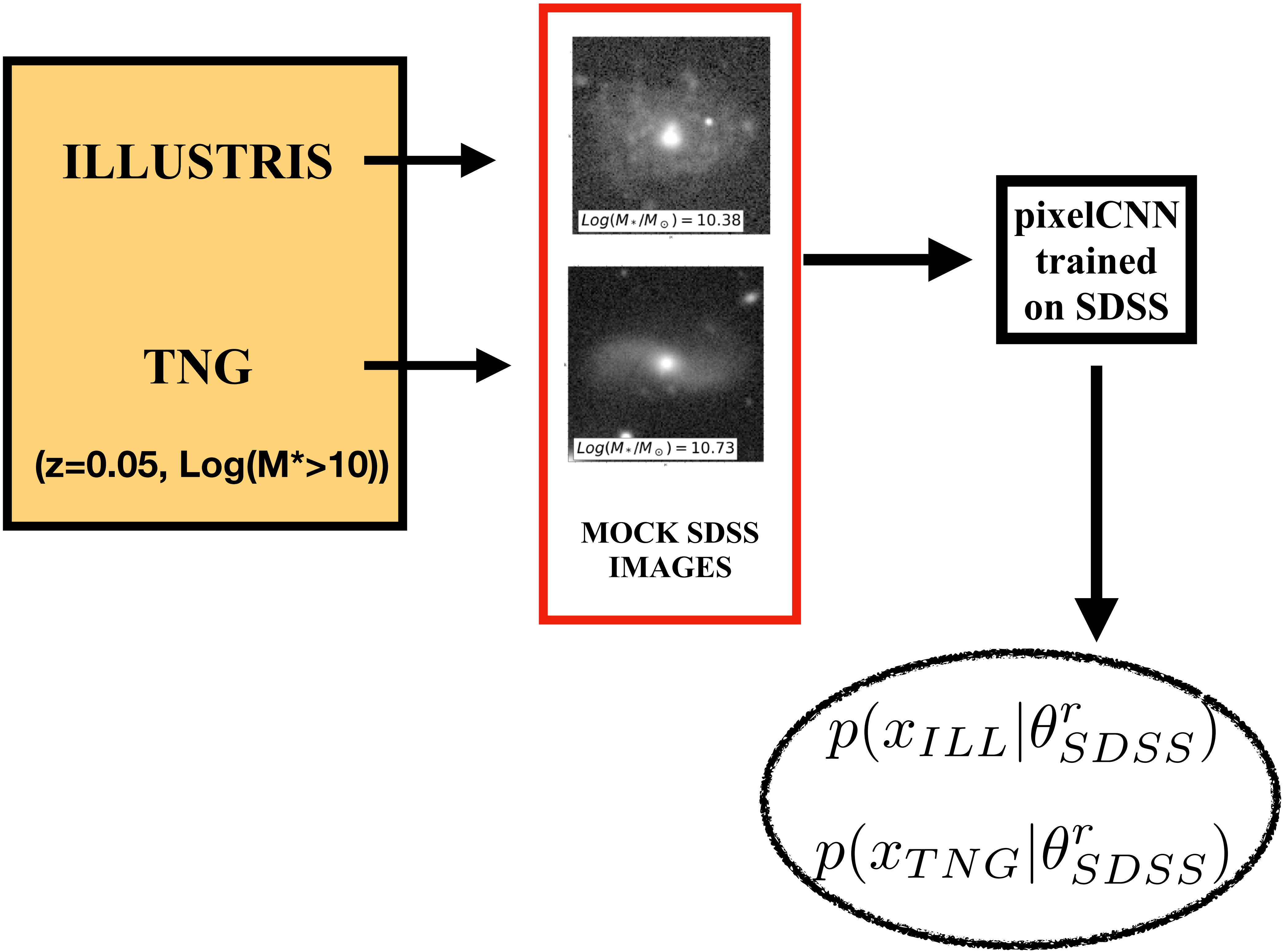
# <u>RANDOM SAMPLES FROM THE MODEL</u> $\theta^r_{SDSS}$

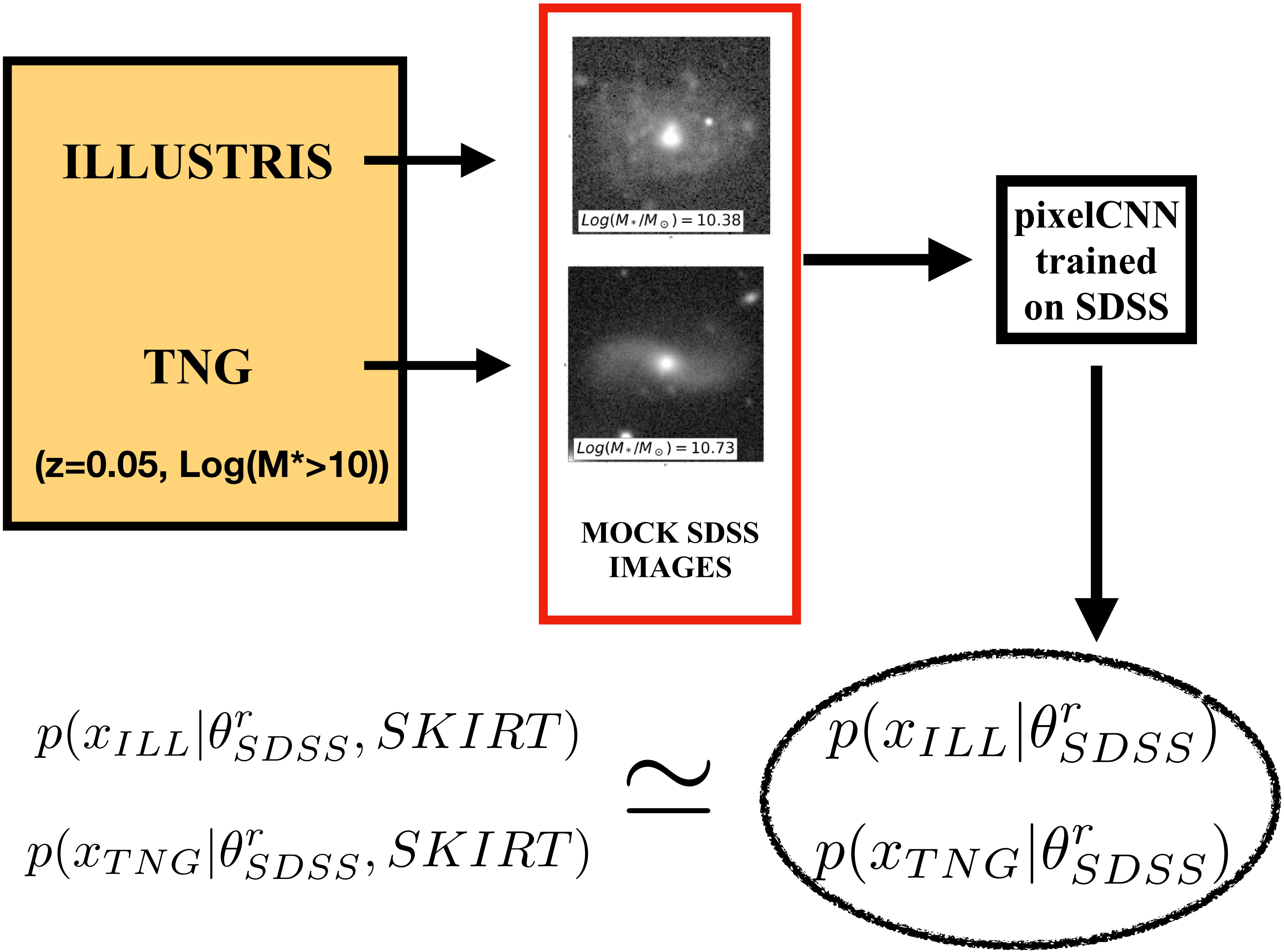


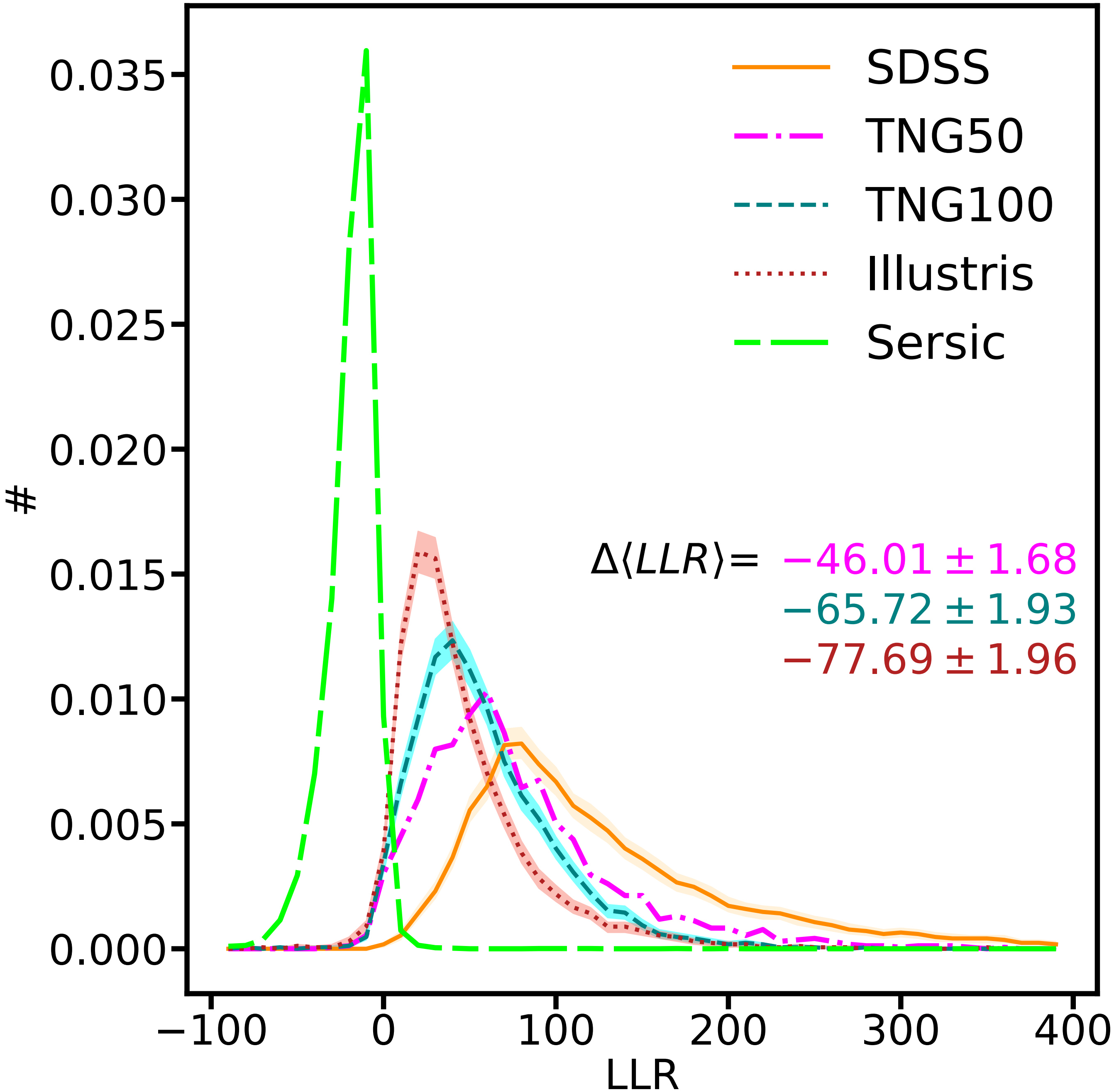
**"FAKE"**
**RANDOM SDSS GALAXIES OBTAINED THROUGH SAMPLING OF THE PDFs**

# DISTRIBUTION OF p(x) for SDSS GALAXIES

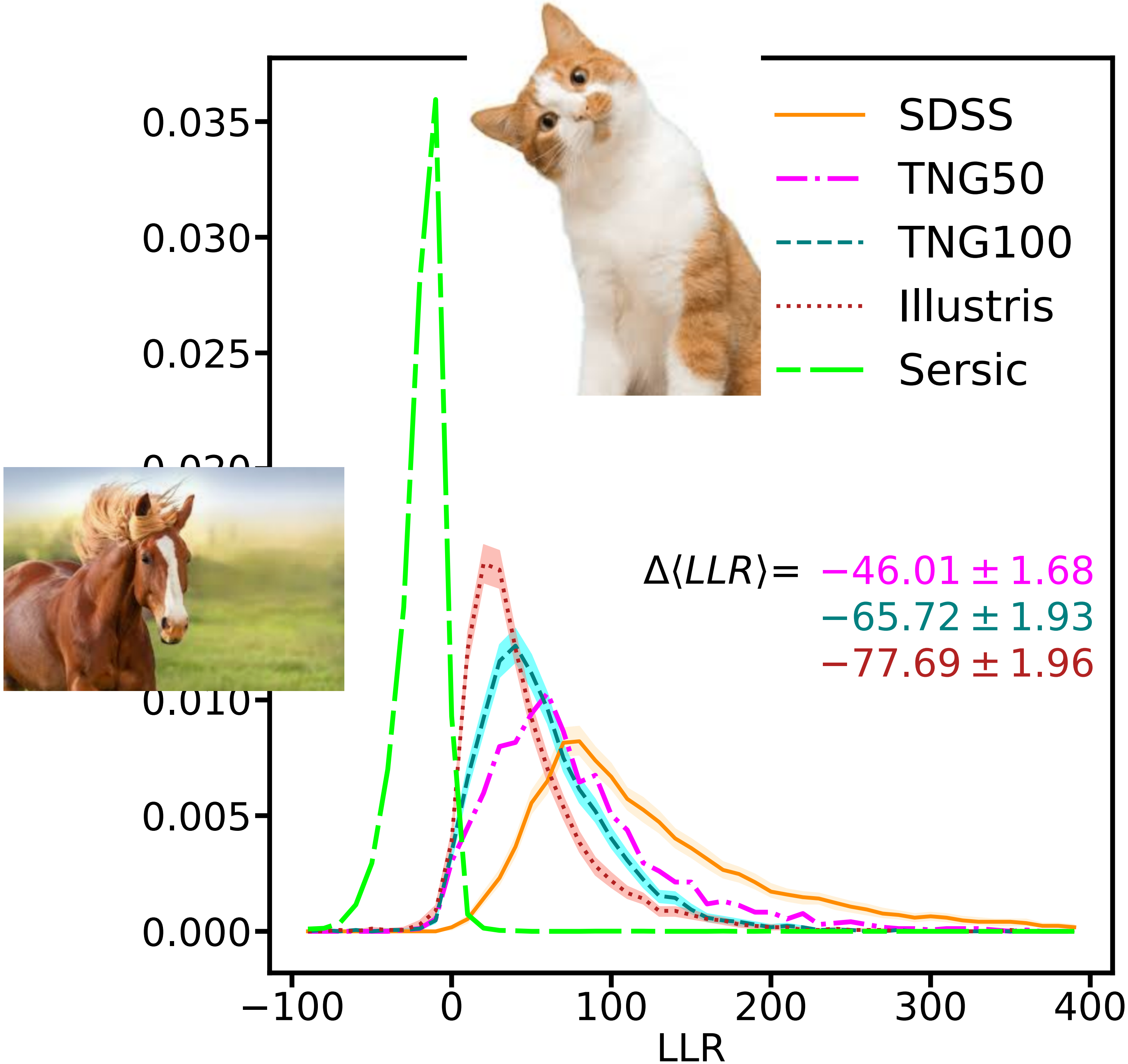



ILLUSTRIS
TNG
(z=0.05, Log(M*>10))
$Log(M_*/M_\odot) = 10.38$
$Log(M_*/M_\odot) = 10.73$
MOCK SDSS IMAGES
pixelCNN trained on SDSS
$p(x_{ILL}|\theta^r_{SDSS})$
$p(x_{TNG}|\theta^r_{SDSS})$


ILLUSTRIS
TNG
(z=0.05, Log(M*>10))
$Log(M_*/M_\odot) = 10.38$
$Log(M_*/M_\odot) = 10.73$
MOCK SDSS IMAGES
pixelCNN trained on SDSS
$p(x_{ILL}|\theta^r_{SDSS}, SKIRT)$
$p(x_{TNG}|\theta^r_{SDSS}, SKIRT)$
$\simeq$
$p(x_{ILL}|\theta^r_{SDSS})$
$p(x_{TNG}|\theta^r_{SDSS})$



SDSS
TNG50
TNG100
Illustris
Sersic
Δ⟨LLR⟩= −46.01 ± 1.68
−65.72 ± 1.93
−77.69 ± 1.96
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
#
−100
0
100
200
300
400
LLR


SDSS
TNG50
TNG100
Illustris
Sersic
Δ⟨LLR⟩= −46.01 ± 1.68
−65.72 ± 1.93
−77.69 ± 1.96
0.035
0.030
0.025
0.010
0.005
0.000
−100
0
100
200
300
400
LLR