PART IV: A VERY BRIEF INTRODUCTION TO DEEP UNSUPERVISED LEARNING

*elements taken from D. Kirkby lectures at KSPA19

WHAT DOES MACHINE LEARNING DO?

the machine is told what to look for

SUPERVISED

LEARNS A MAP FROM X [FEATURES] TO Y [LABELS]

$$P(X|Y)$$

the machine is NOT told what to look for

UN-SUPERVISED

NO LABELS - DISCOVER PATTERNS

$$P(X)$$

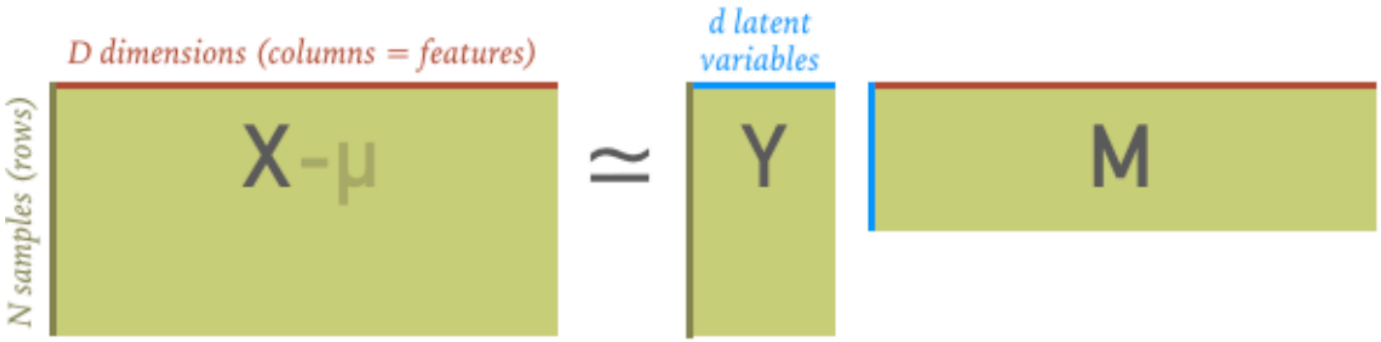
IN THIS LAST PART WE ARE GOING TO BRIEFLY INTRODUCE CURRENT TECHNIQUES OF UNSUPERVISED LEARNING WITH NEURAL NETWORKS

THERE ARE 3 MAJOR APPLICATIONS TO ASTRONOMY (THAT I CAN THINK OF):

- **DIMENSIONALITY REDUCTION:** HOW CAN I REPRESENT MY COMPLEX DATA MORE EFFICIENTLY TO GET NEW INSIGHTS INTO ITS STRUCTURE?

- **GENERATIVE MODELING:** HOW CAN I INTERPOLATE / EXTRAPOLATE A (SMALL, SPARSE) DATASET TO GENERATE NEW DATA SAMPLED FROM THE SAME (UNKNOWN) DISTRIBUTION?

- **PROBABILISTIC MODELING:**WHAT IS THE PROBABILITY THAT A NEW OBSERVATION IS DRAWN FROM THE SAME (UNKNOWN) DISTRIBUTION AS SOME REFERENCE (SMALL, SPARSE) DATASET?



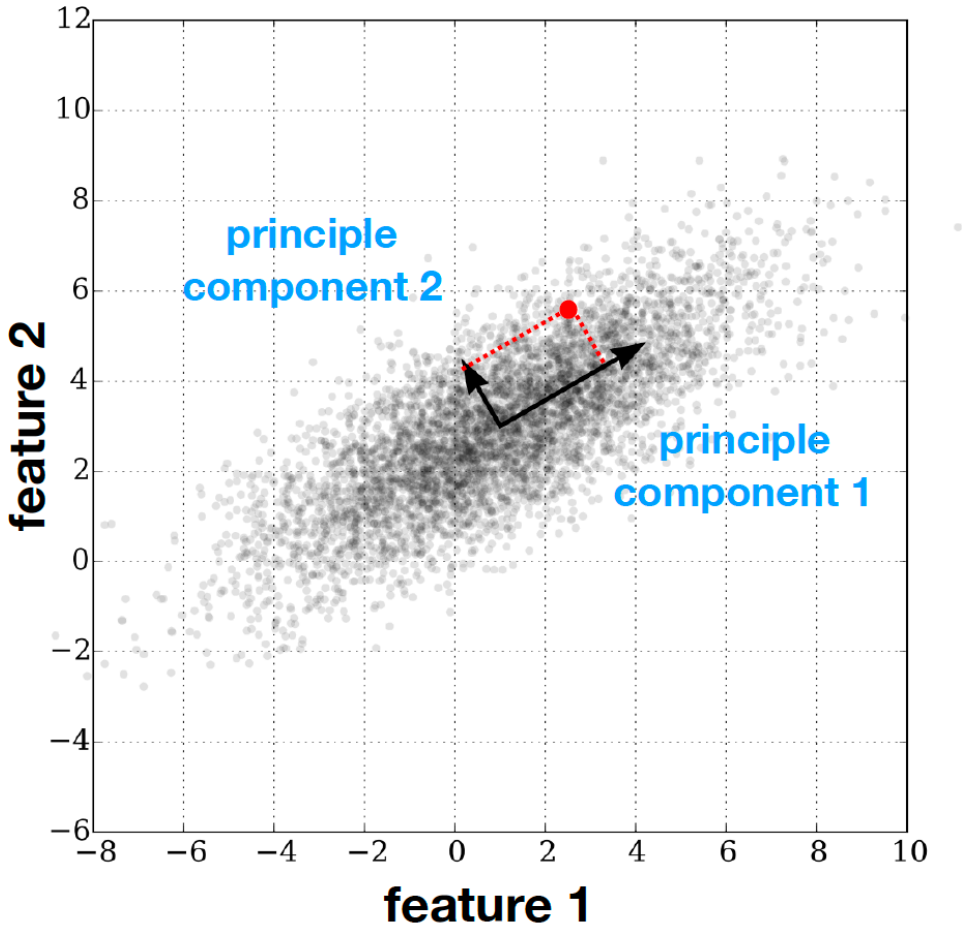
CLASSICAL METHODS FOR DIMENSIONALITY REDUCTION SEEK A LINEAR DECOMPOSITION THAT BEST EXPLAINS THE OBSERVATIONS X IN TERMS OF **LATENT VARIABLES Y**

PRINCIPAL COMPONENT ANALYSIS

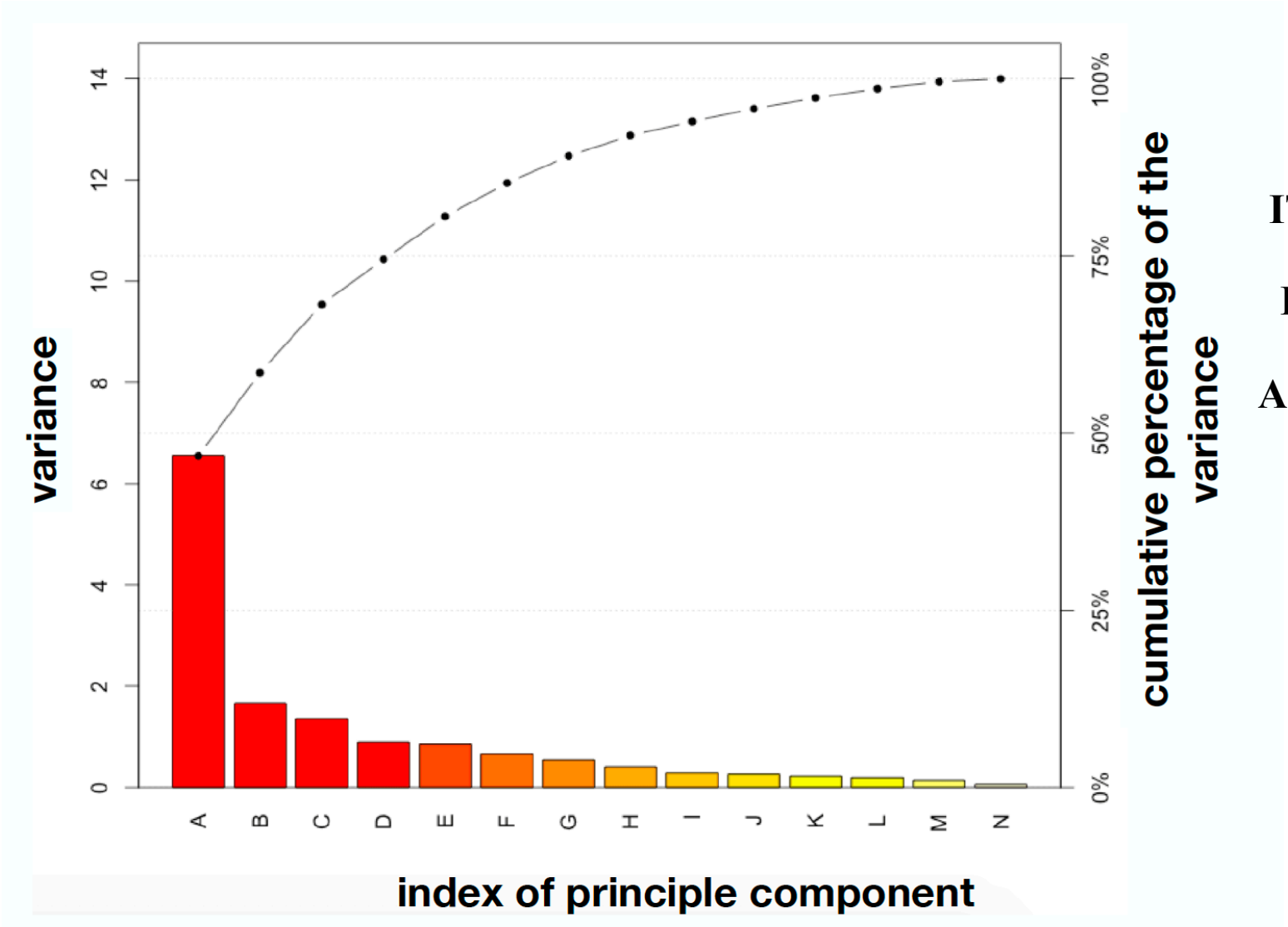
PCA CONVERT A SET OF (CORRELATED) VARIABLES INTO A SET OF VALUES LINEARLY UNCORRELATED

1. THE FIRST PRINCIPLE COMPONENT ("PROTOTYPE"), HAS THE LARGEST POSSIBLE VARIANCE

2. THE FOLLOWING COMPONENTS HAVE THE LARGEST VARIABLES WITH THE ADDITIONAL CONSTRAIN THAT THEY ARE ORTHOGONAL TO THE PRECEDING COMPONENTS



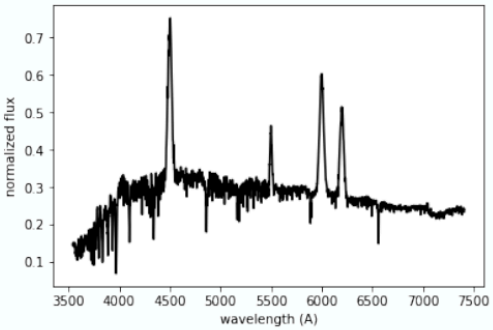
PRINCIPAL COMPONENT ANALYSIS



IT RESULTS IN DATA COMPRESSION, BY REPRESENTING EACH OBJECT AS A PROJECTION OF THE FIRST PRINCIPLE COMPONENTS

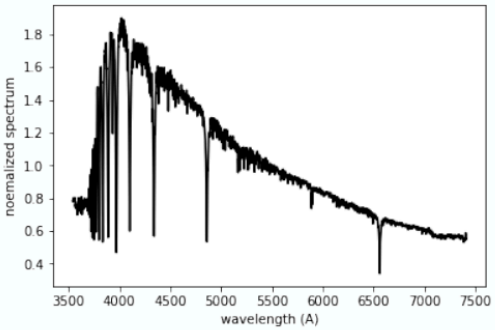
PRINCIPAL COMPONENT ANALYSIS

observed object



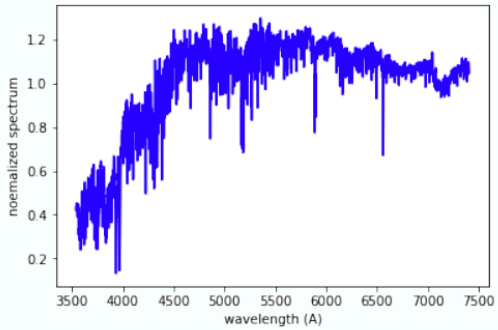
= A *

principle comp. 1



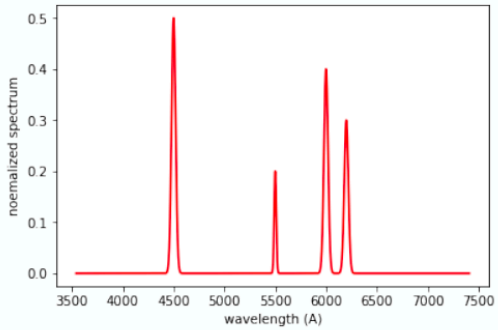
+ B *

principle comp. 2



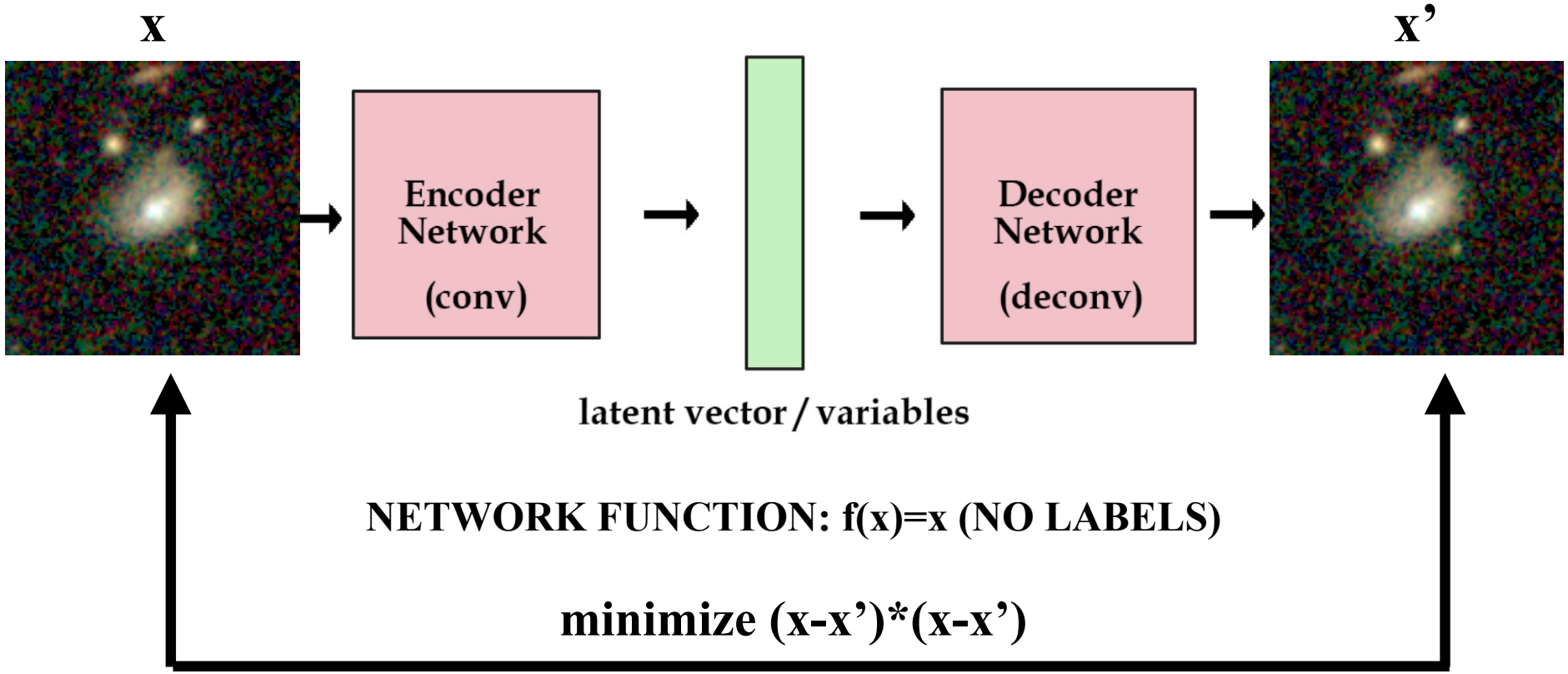
+ C *

principle comp. 3



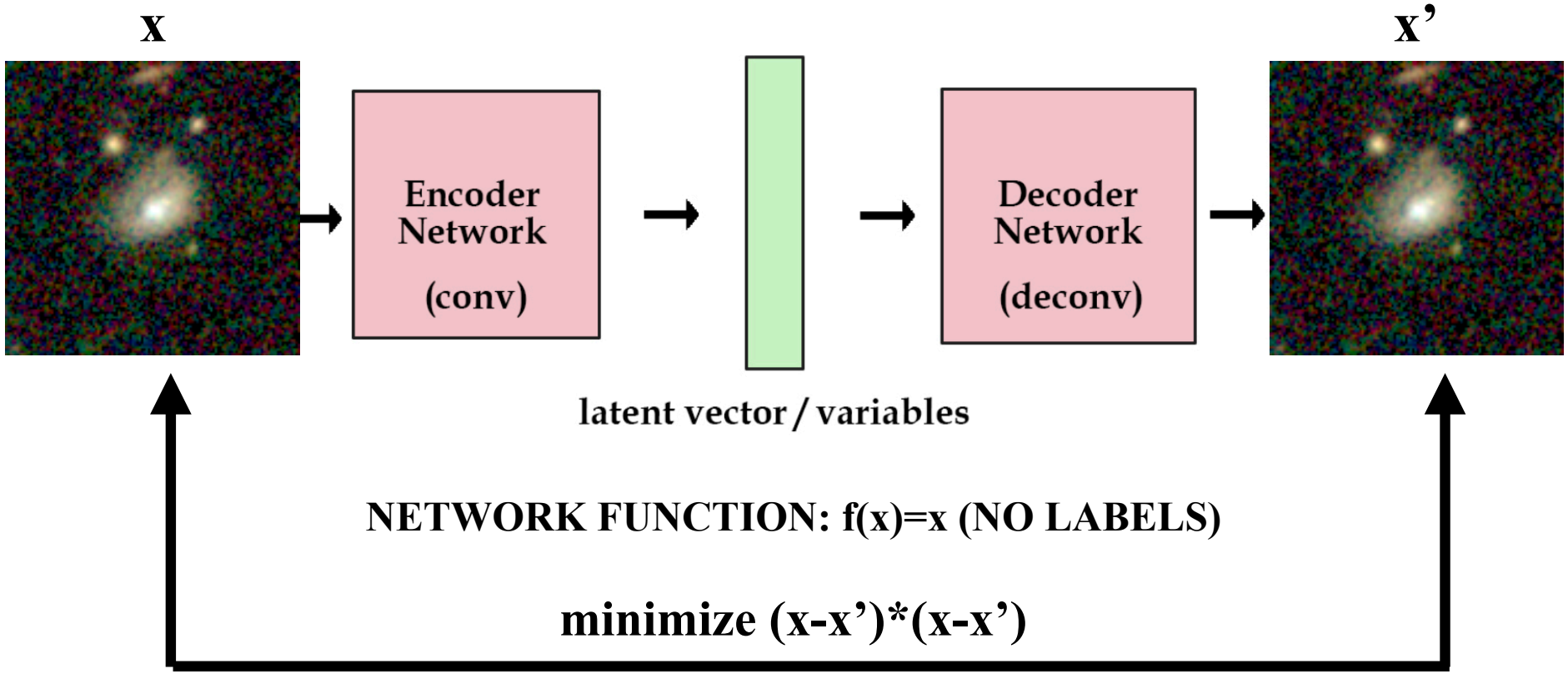
OBJECTS ARE DIVIDED INTO MAJOR PROTOTYPES AND ALL OBJECTS CAN BE OBTAINED AS LINEAR COMBINATIONS

AUTO-ENCODER



AN AUTO-ENCODER IS ANY NETWORK WITH IDENTICAL INPUT AND OUTPUT

AUTO-ENCODER



BY REDUCING THE DIMENSIONALITY IN THE LATENT SPACE WE FORCE THE NETWORK TO LEARN A REPRESENTATION OF THE INPUT DATA IN A LOWER DIMENSIONALITY SPACE

* NO NEED TO BE CONVOLUTIONAL - ANY NEURAL NETWORK WITH A BOTTLENECK WILL DO THE JOB

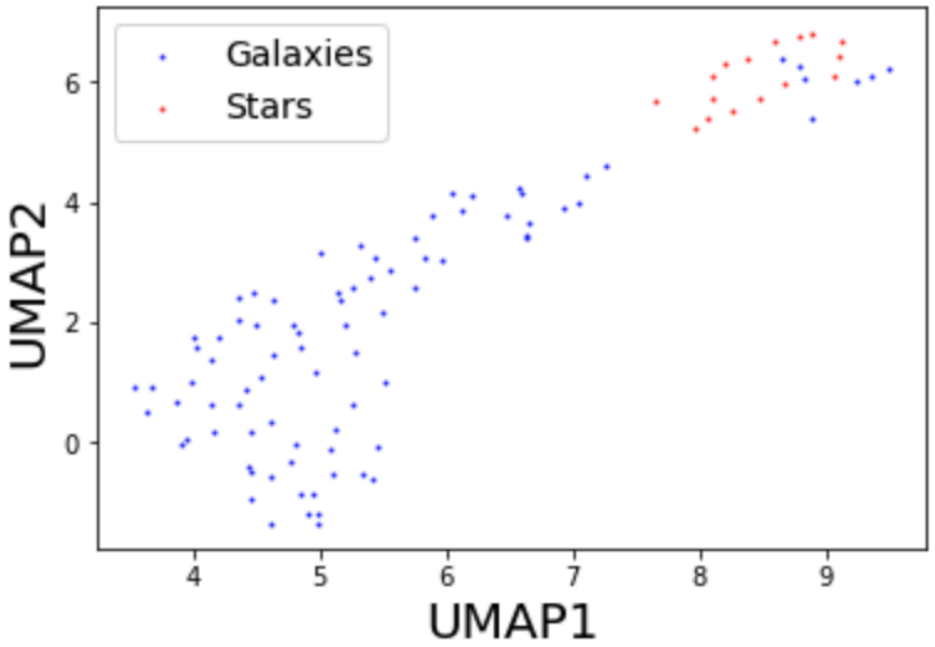
* **QUESTION**: WHAT WOULD HAPPEN IF WE SET AN AUTOENCODER WITH NO ACTIVATION FUNCTIONS?

SEE EXAMPLE FROM TUTORIALS FOR STAR-GALAXY SEPARATION

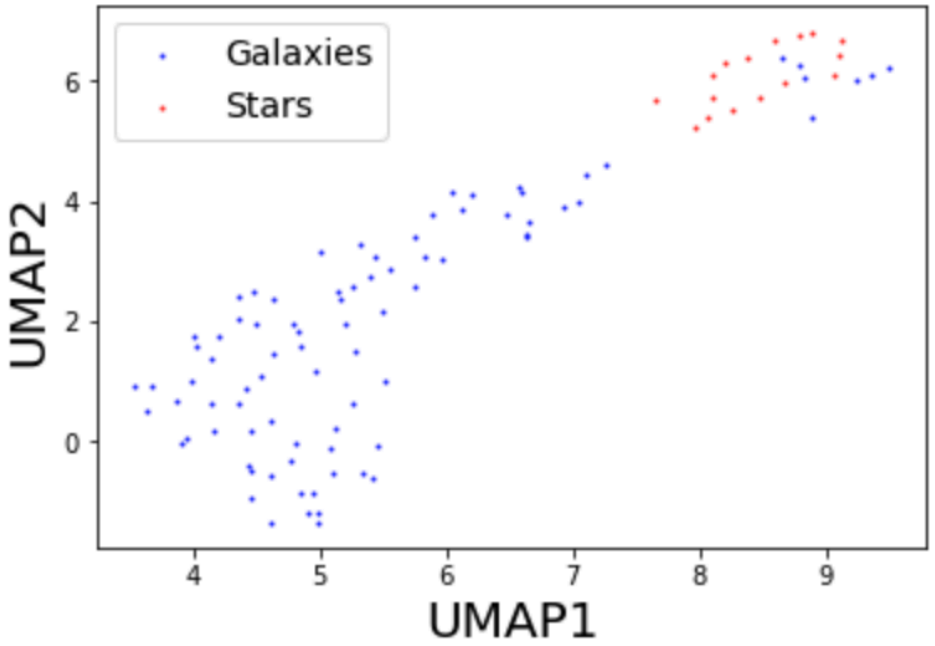
HOW MIGHT YOU SOLVE THESE RELATED PROBLEMS ONCE YOU HAVE THE LATENT SPACE COORDINATES FOR YOUR TRAINING SAMPLE?

GENERATE A RANDOM SAMPLE DRAWN FROM THE INPUT DISTRIBUTION ("**GENERATIVE MODEL**")

ESTIMATE THE PROBABILITY DENSITY OF AN ARBITRARY INPUT, RELATIVE TO THE INPUT DISTRIBUTION ("**PROBABILISTIC MODEL**")



Both of these problems could be solved using density estimation methods. Given the approximately Gaussian distributions above, a Gaussian mixture model (GMM) would be a good approach.



Both of these problems could be solved using density estimation methods. Given the approximately Gaussian distributions above, a Gaussian mixture model (GMM) would be a good approach.

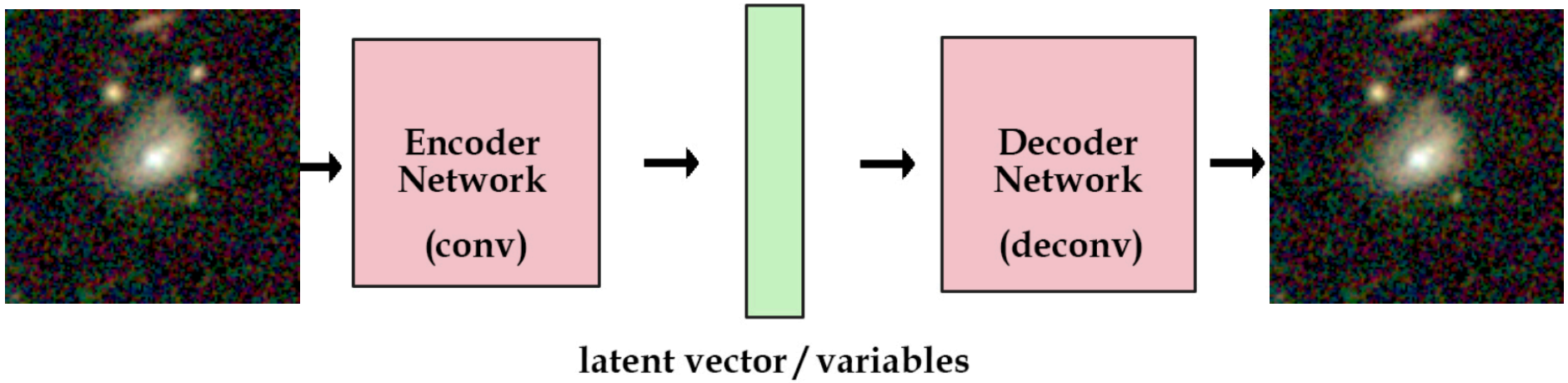
Much of the recent progress in unsupervised deep learning has been to invent network architectures that are capable of solving either or both of these related problems directly, without resorting to any auxiliary methods

VAE **GAN** **ARF**

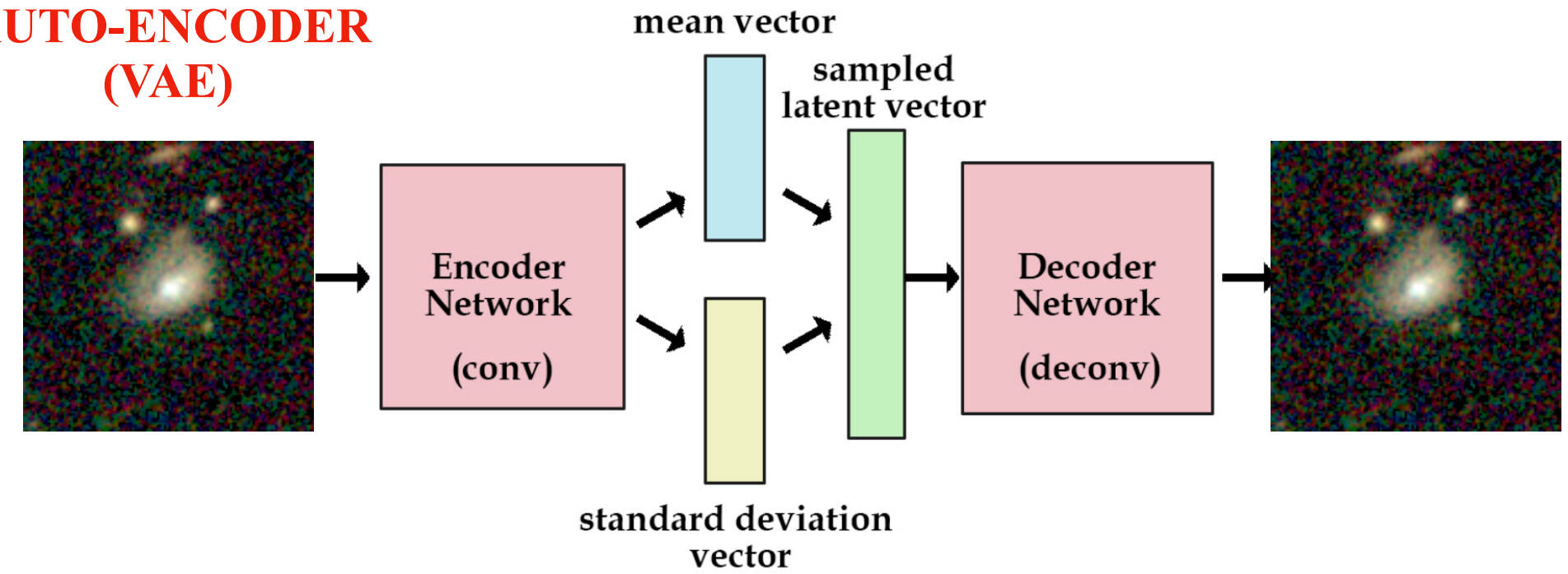
GENERATIVE MODELS

Generate a random sample drawn from the input distribution

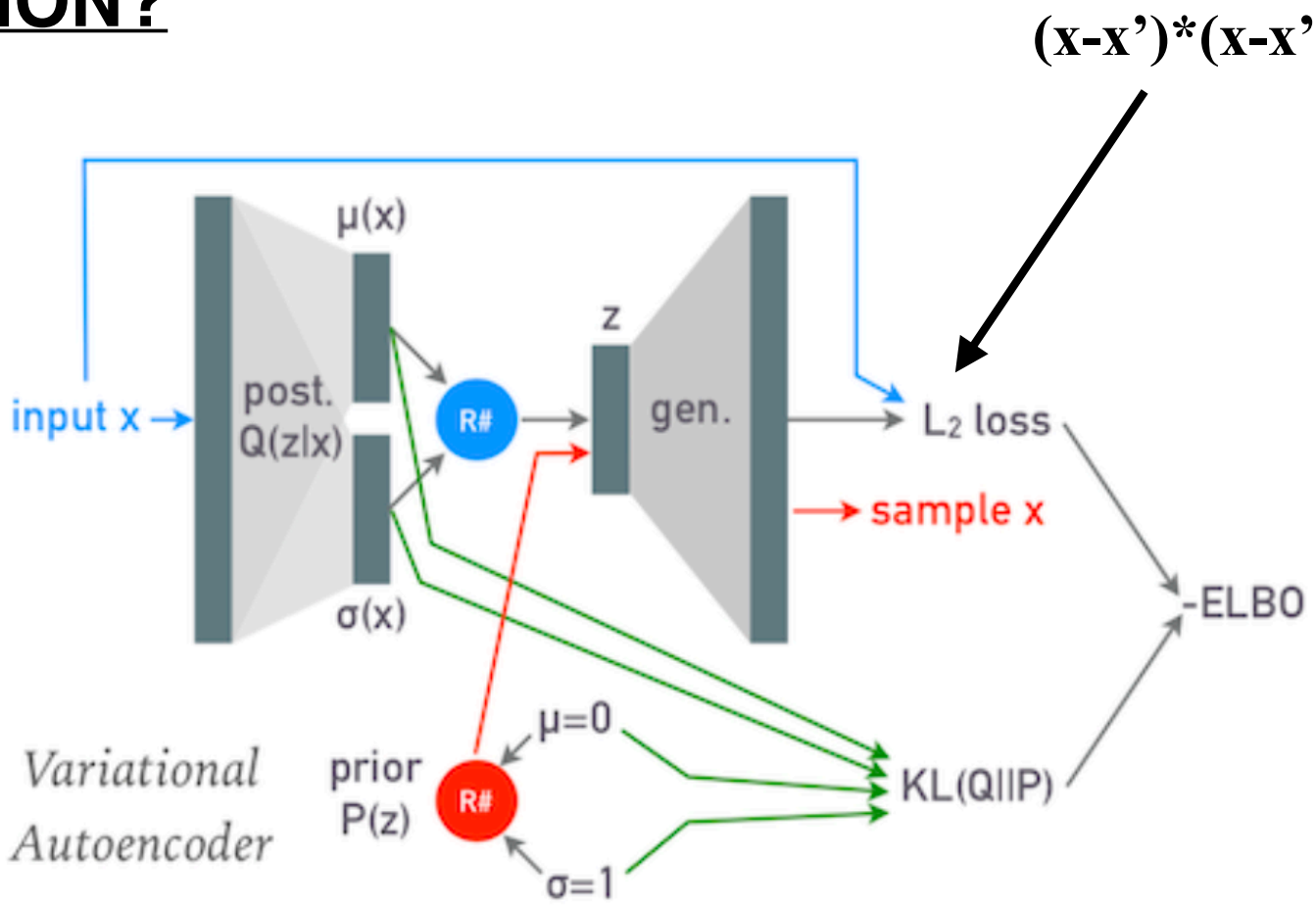
AUTO-ENCODER



VARIATIONAL AUTO-ENCODER (VAE)



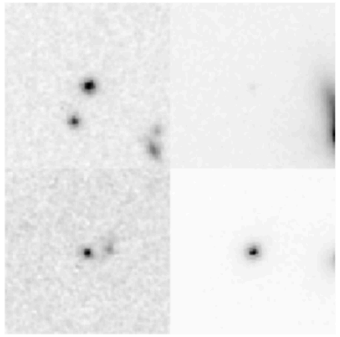
LOSS FUNCTION?



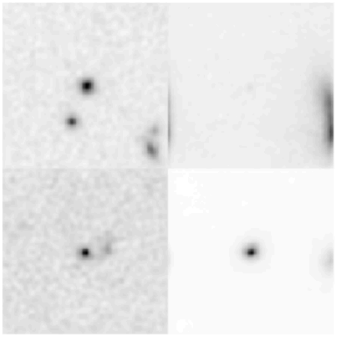
The key insight of VAE is that we are actually performing **variational inference** here, which then tells us what the loss function should be...

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

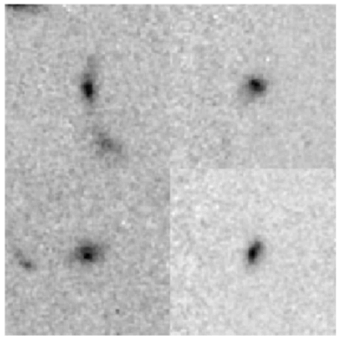
training data originals



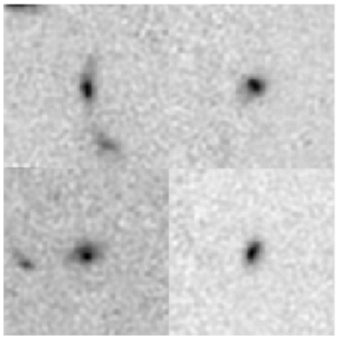
training data reconstructions



validation data originals

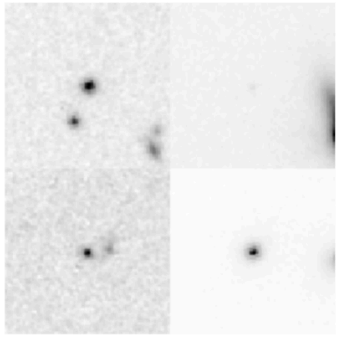


validation data reconstructions

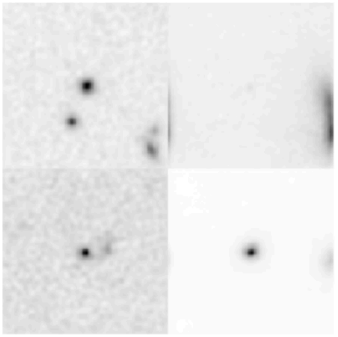


RECONSTRUCTION

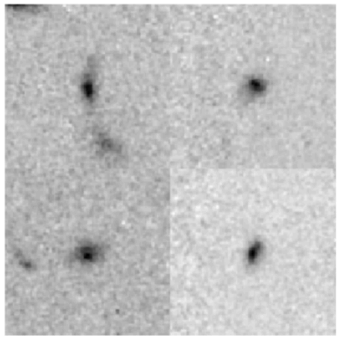
training data originals



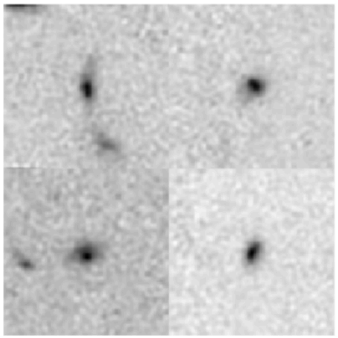
training data reconstructions



validation data originals

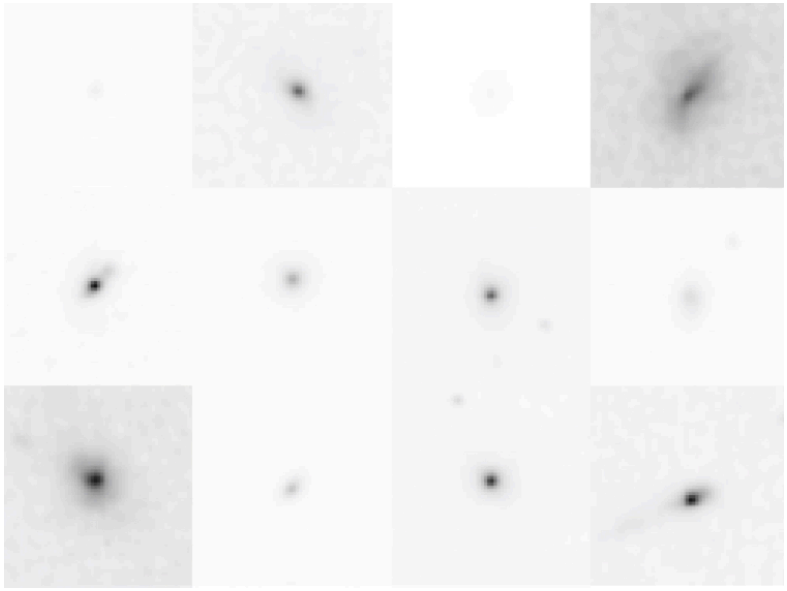


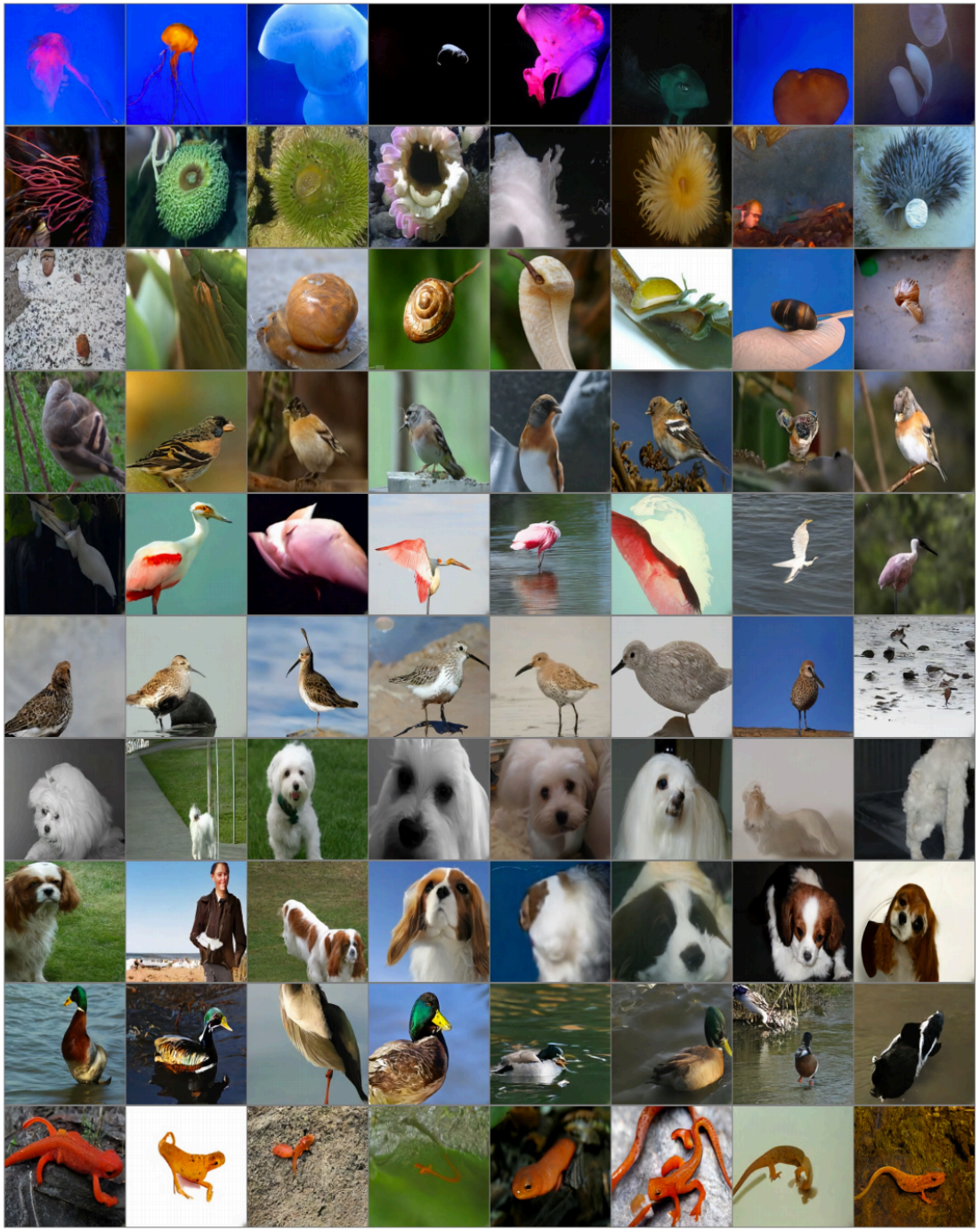
validation data reconstructions



RECONSTRUCTION

SAMPLING

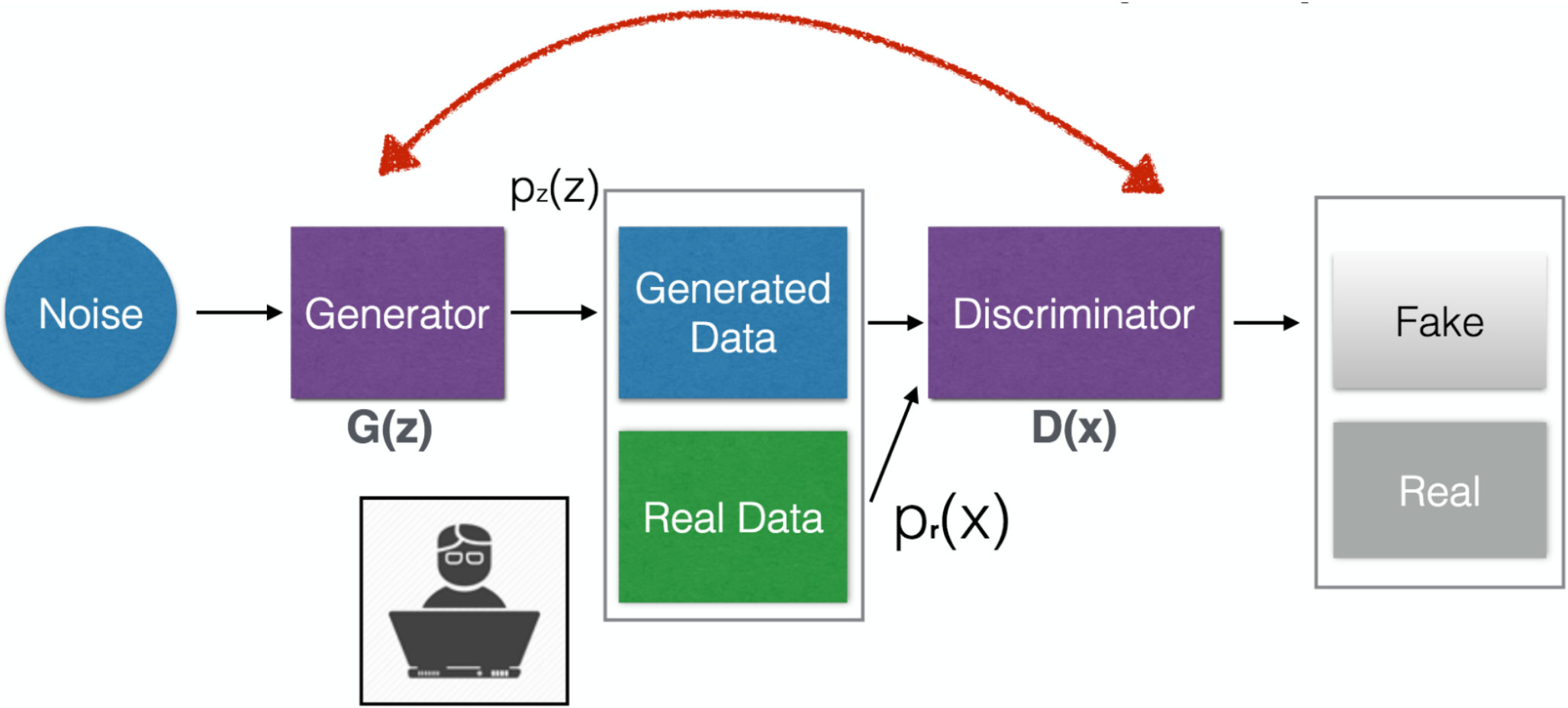




Razavi+19
(deepmind)

GENERATIVE ADVERSARIAL NETWORKS

(Goodfellow+14)

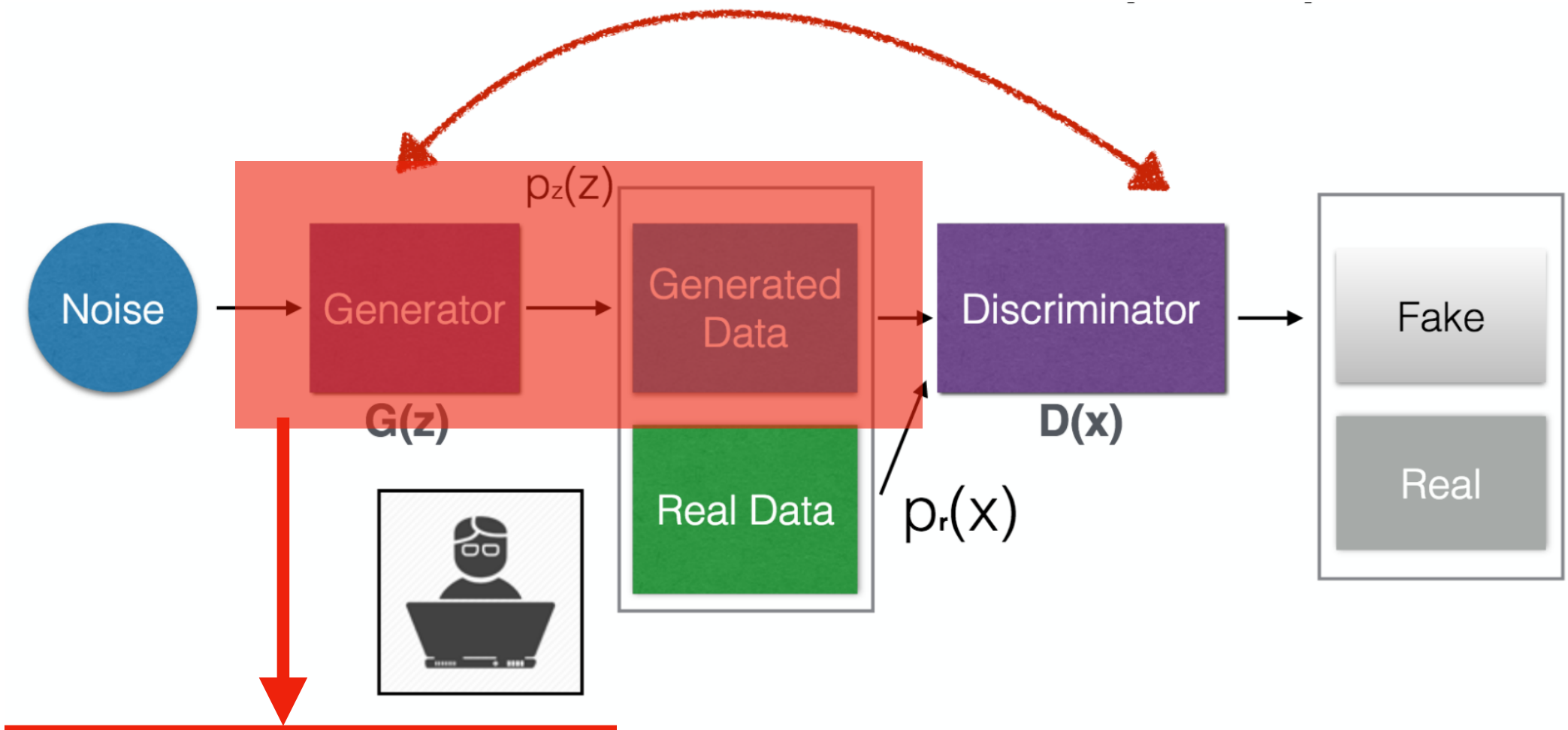


TWO COMPETING NETWORKS

GENERATIVE ADVERSARIAL NETWORKS

(Goodfellow+)

TWO COMPETING NETWORKS

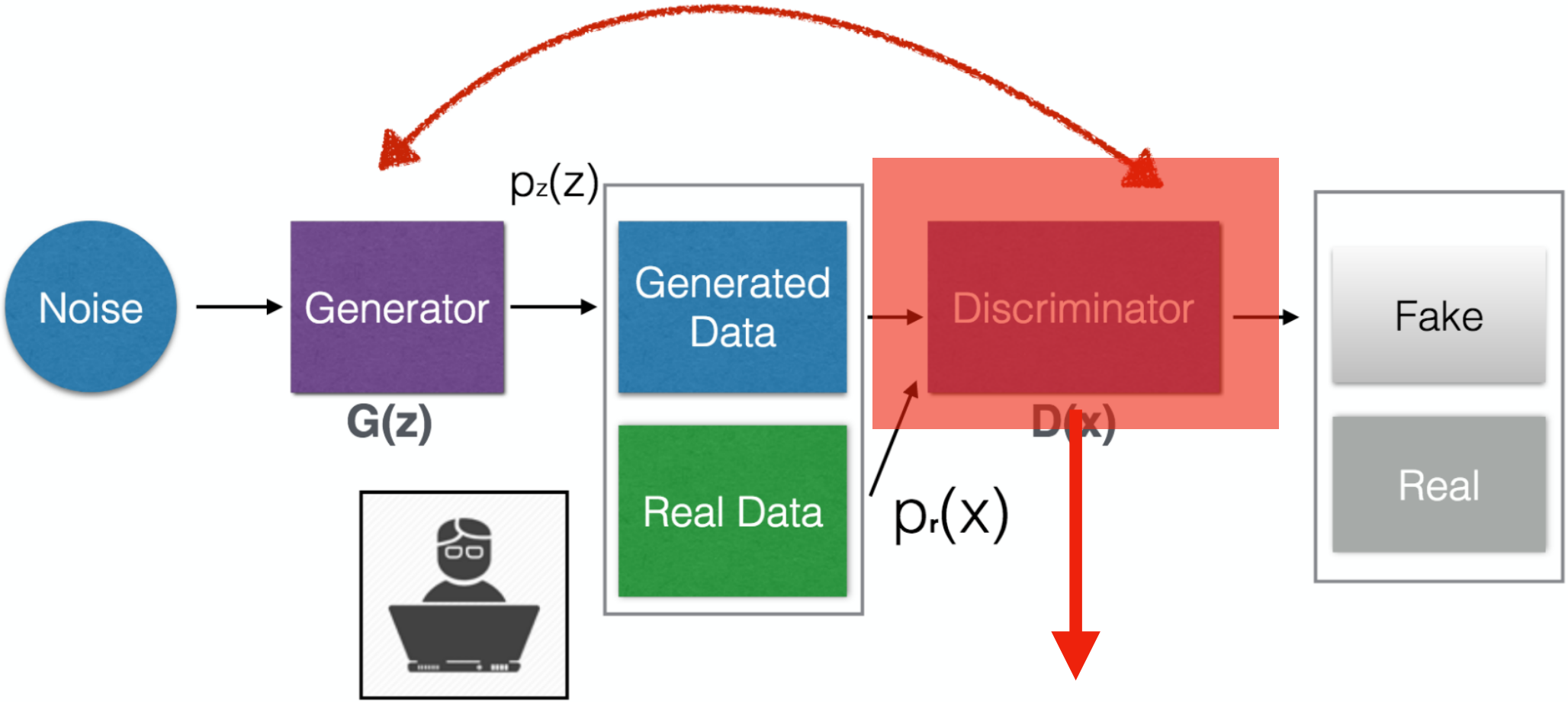


Every N iterations the generator is trained to force the discriminator to classify as real

GENERATIVE ADVERSARIAL NETWORKS

(Goodfellow+)

TWO COMPETING NETWORKS



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

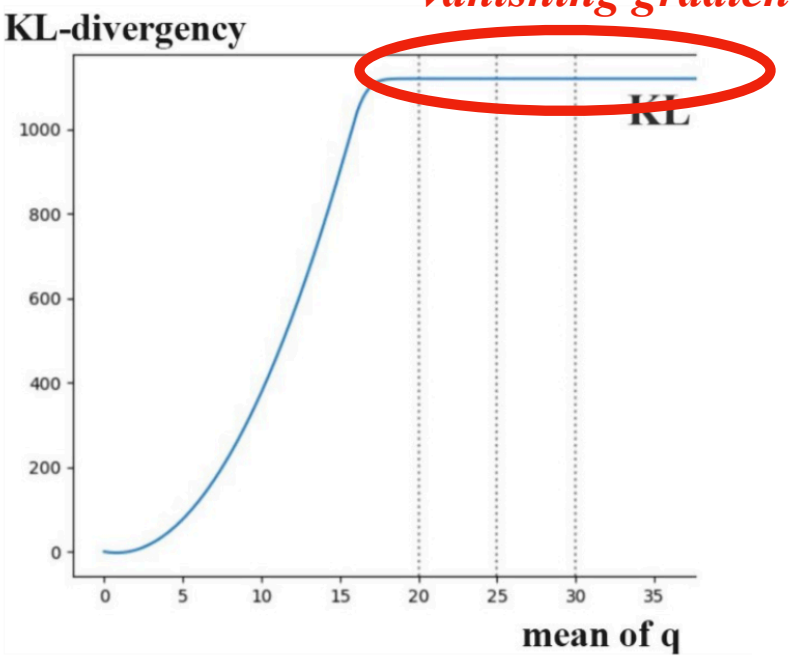
Karras+19

GANs ARE HARD TO TRAIN

KL Divergence

$$D_{KL}(P||Q) = \sum_{x=1}^{N} P(x) \log \frac{P(x)}{Q(x)}$$

vanishing gradient



Example from here

WGAN-GP

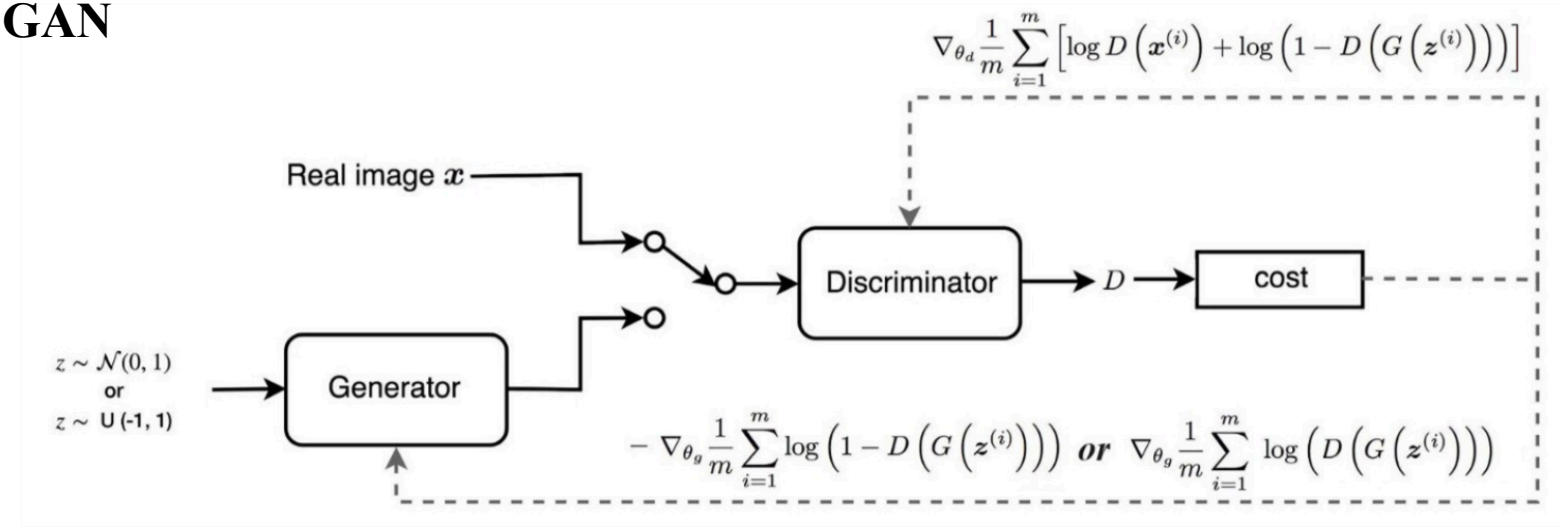
$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\, \|x - y\| \,\right],$$

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

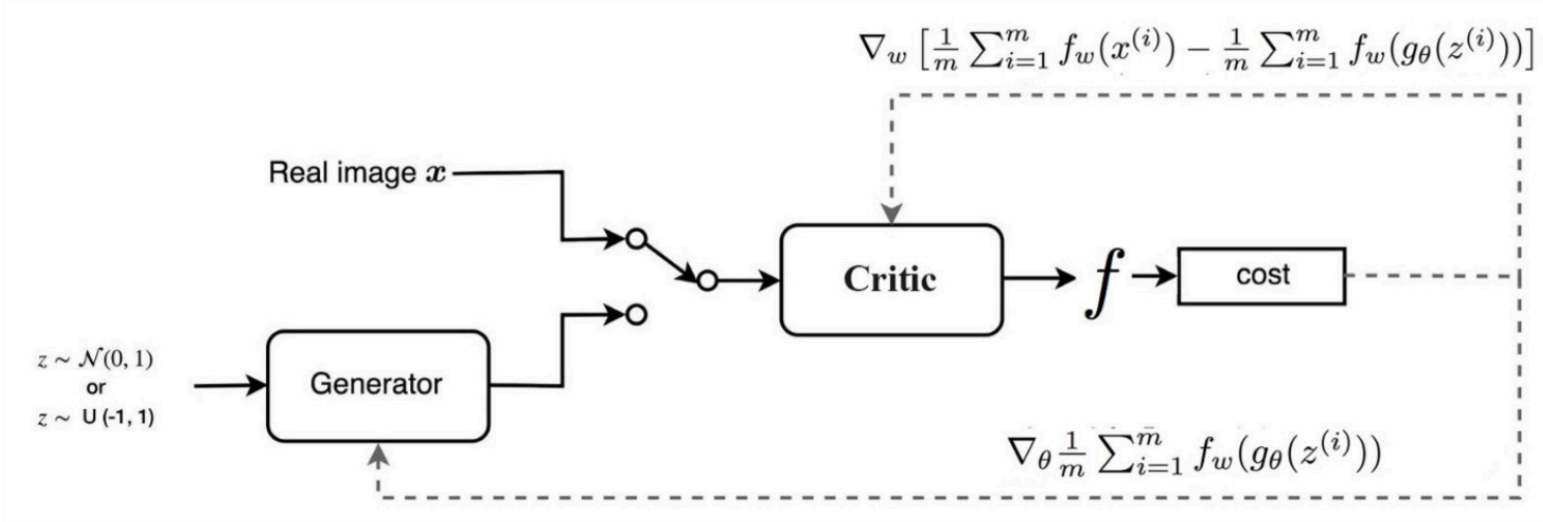
1-Lipschitz function

IN PRACTICE

GAN



WGAN



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	✓	✓	✓	✓

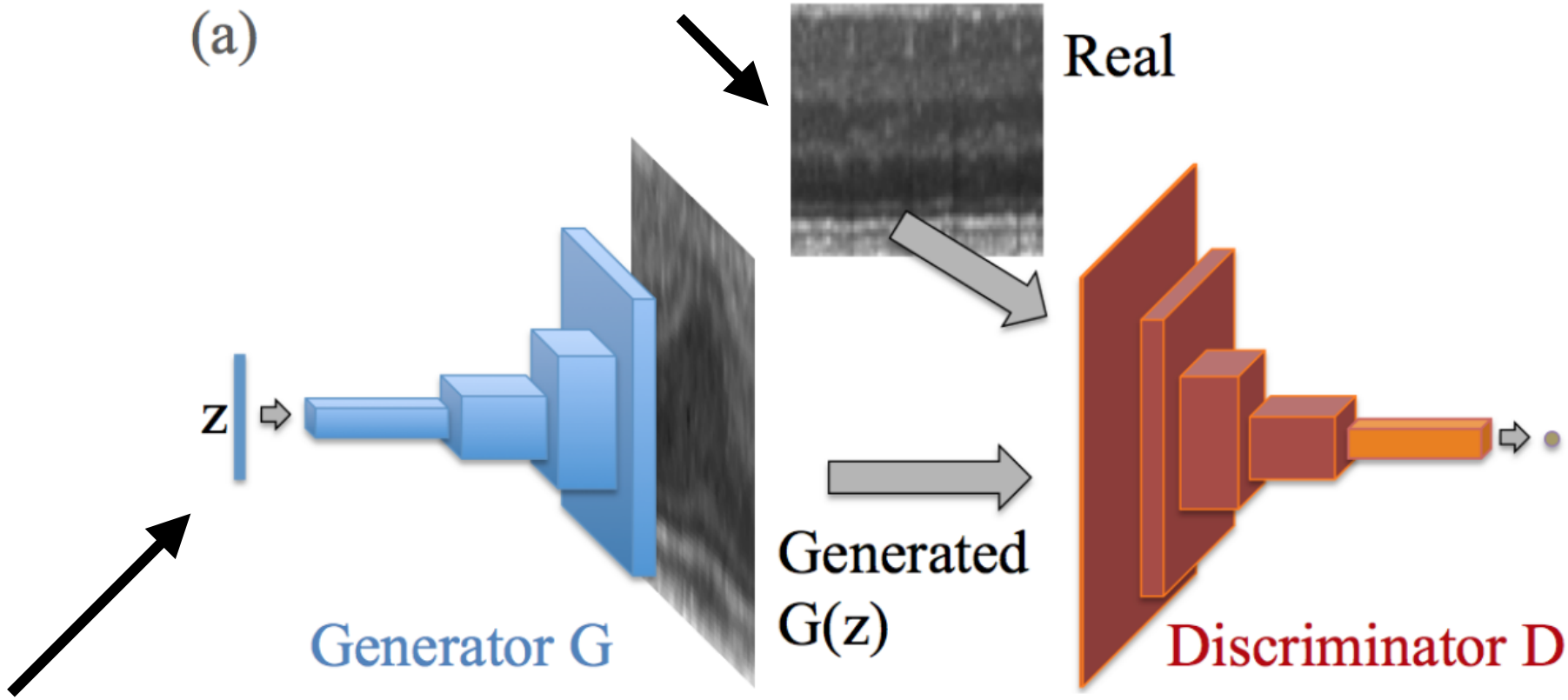
Table 1: A comparison of the abilities of generative modeling approaches.

Grathwohl+18

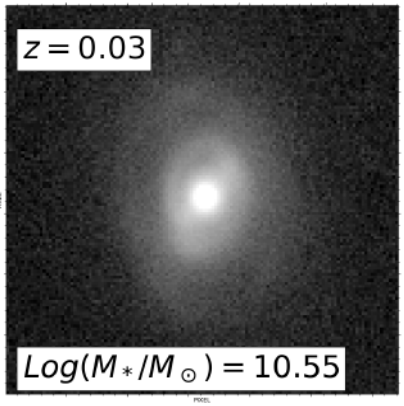
MAIN CAVEAT: INFERENCE IS VERY SLOW!

ANOMALY DETECTION WITH GANs

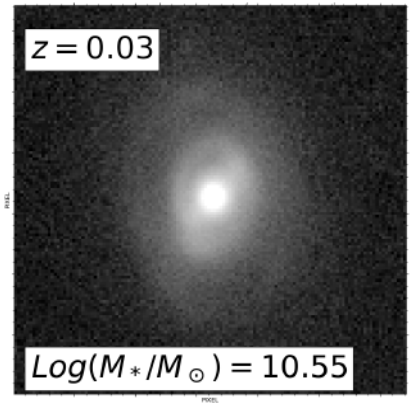
OBSERVATIONS



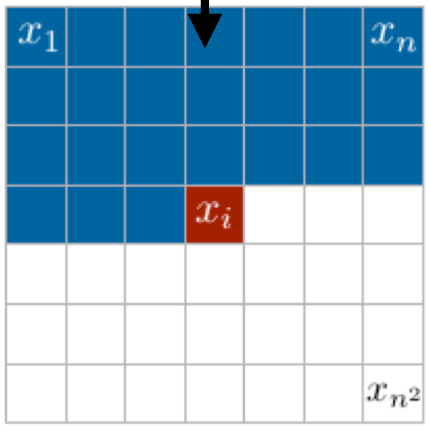
SIMULATIONS **[EAGLE, Illustris, FIRE, VELA…]**

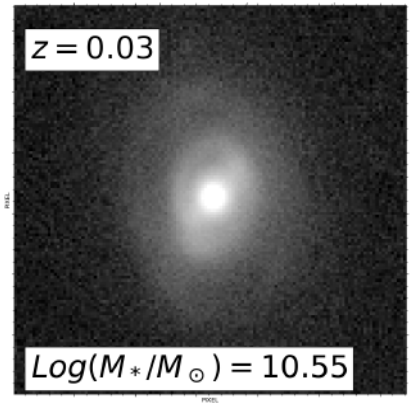


SDSS GALAXY

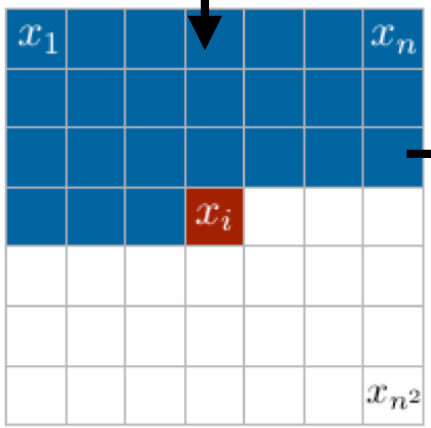


SDSS GALAXY

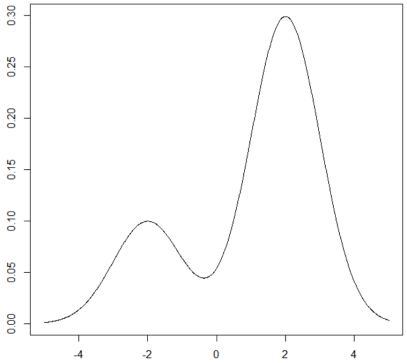


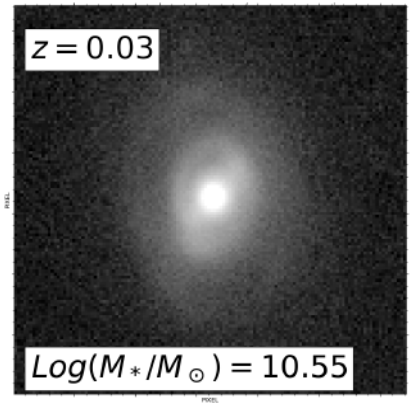


SDSS GALAXY

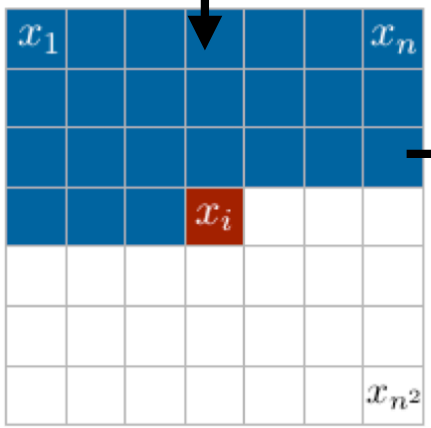


PDF FOR ONE PIXEL

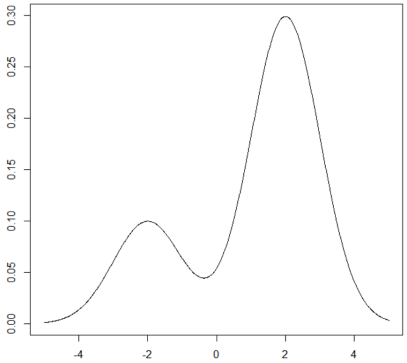




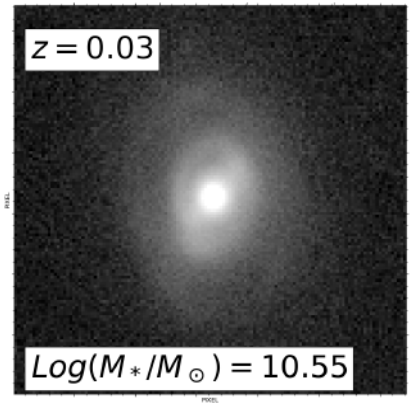
SDSS GALAXY



PDF FOR ONE PIXEL

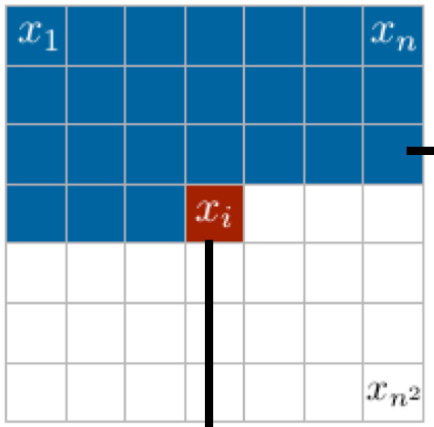


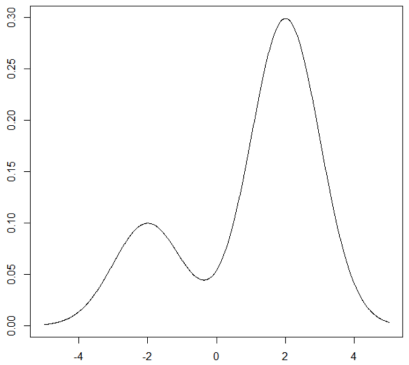
$$p(x) = p(x_0, x_1, ..., x_{n^2} | \theta_{SDSS})$$



SDSS GALAXY

PDF FOR ONE PIXEL

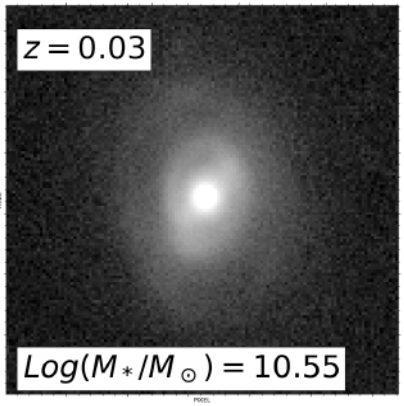




$$p(x) = p(x_0, x_1, ..., x_{n^2} | \theta_{SDSS})$$

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

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

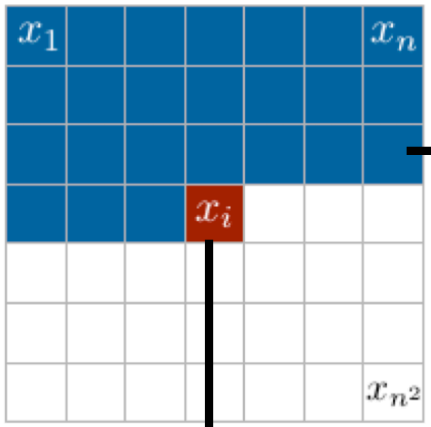


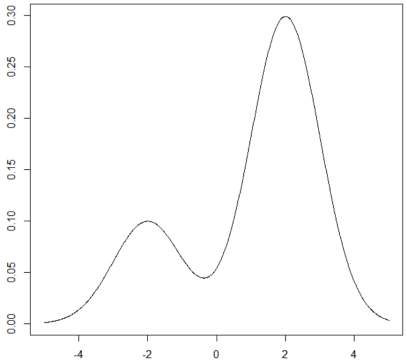
SDSS GALAXY

AUTOREGRESSIVE IMAGE GENERATION: pixelCNN

[van der Oord+16, Salimans+17]

PDF FOR ONE PIXEL





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

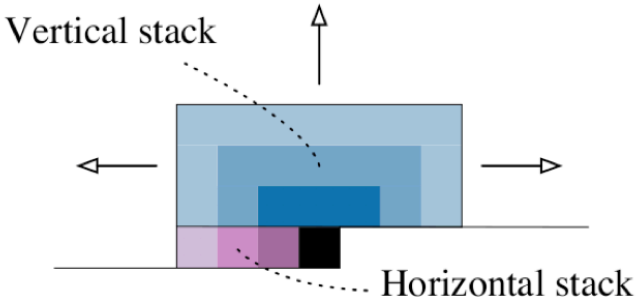
$$p(x_i | x_1, ..., x_{i-1})$$

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

IN PRACTICE….

Masked convolutions

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



Makes learning conditioned only on previous pixels

This implementation gives a “blind spot”

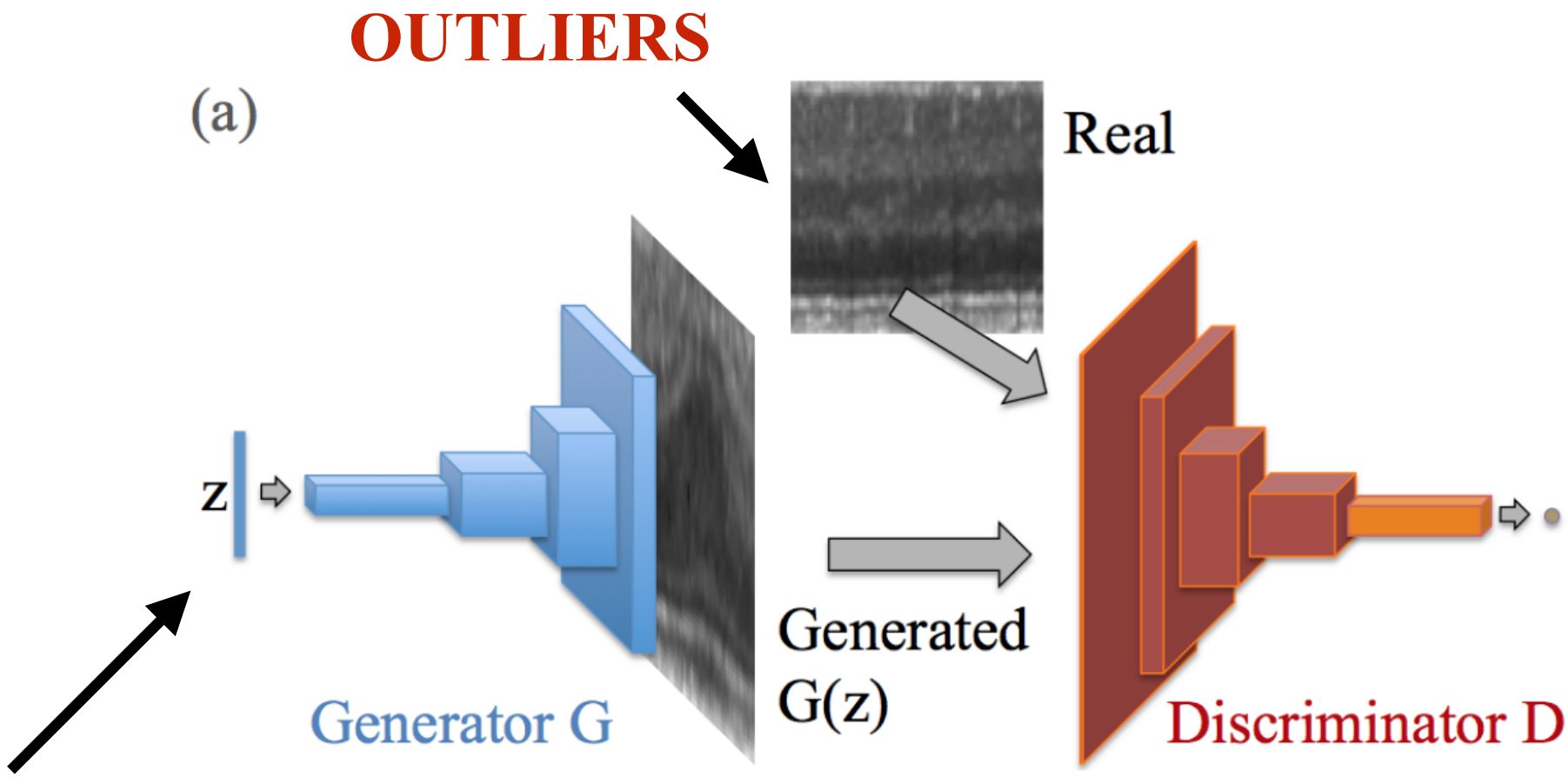
Solution: sum two rectangular convolutions

SOME APPLICATIONS OF GENERATIVE AND PROBABILISTIC MODELS TO ASTRONOMY

ANOMALY DETECTION WITH GANs

- 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

ANOMALY DETECTION WITH GANs



LEARN "NORMAL" DATA

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*($r \sim 26$)
Deep	27	15	*grizy*+4NBs ($r \sim 27$)
Ultradeep	3.5	2	*grizy*+4NBs ($r \sim 28$)

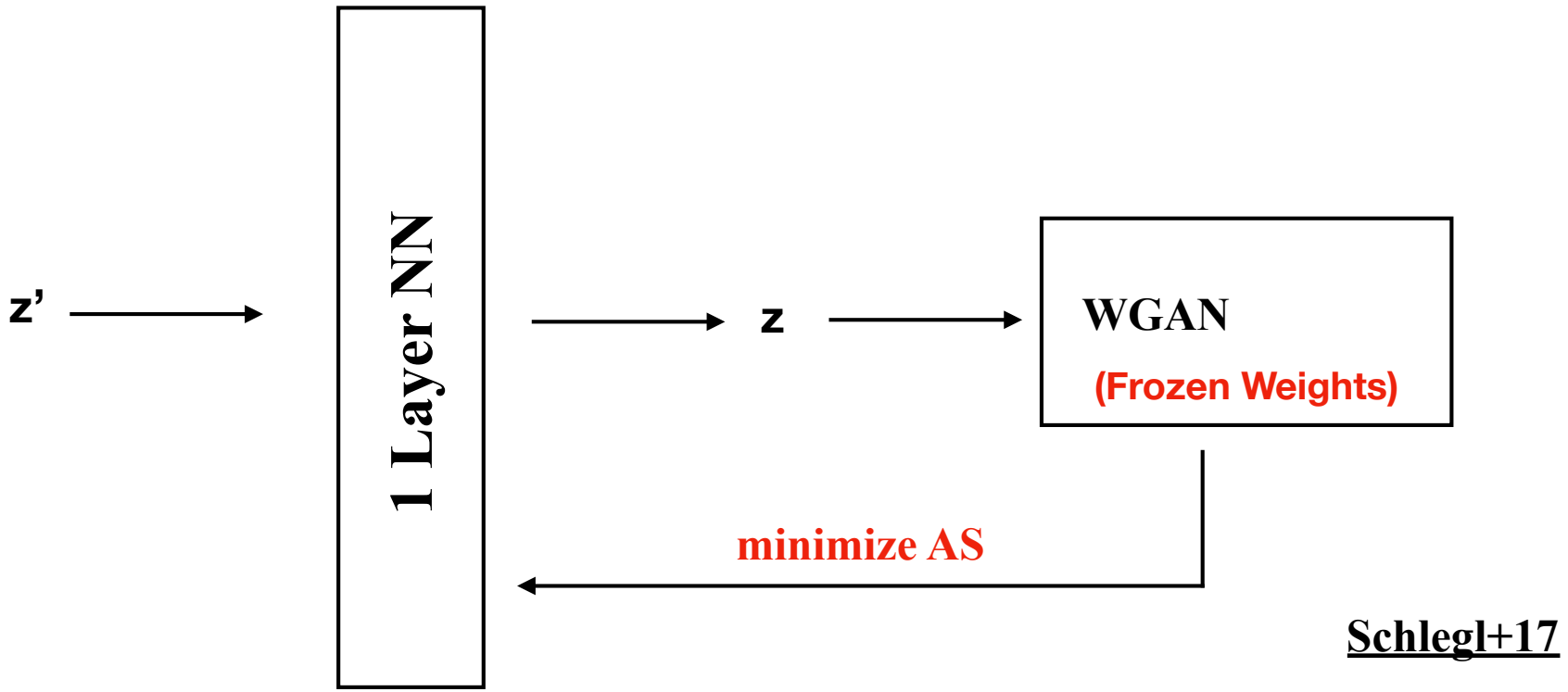
COMPUTE ANOMALY SCORE

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

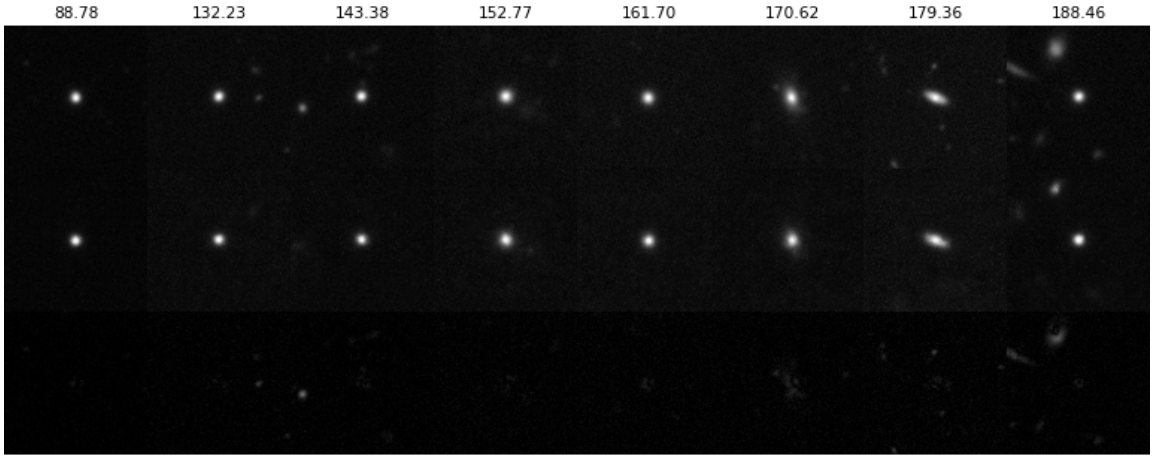
Hyperparameter

RMS btw input and generated

RMS btw critic features from input and generated



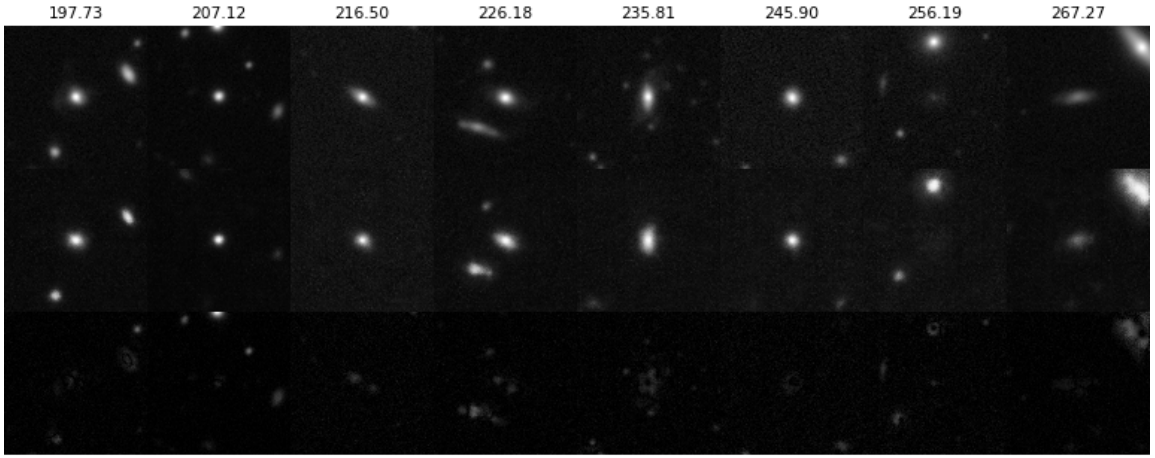
Schlegl+17



REAL

RECONSTRUCTED

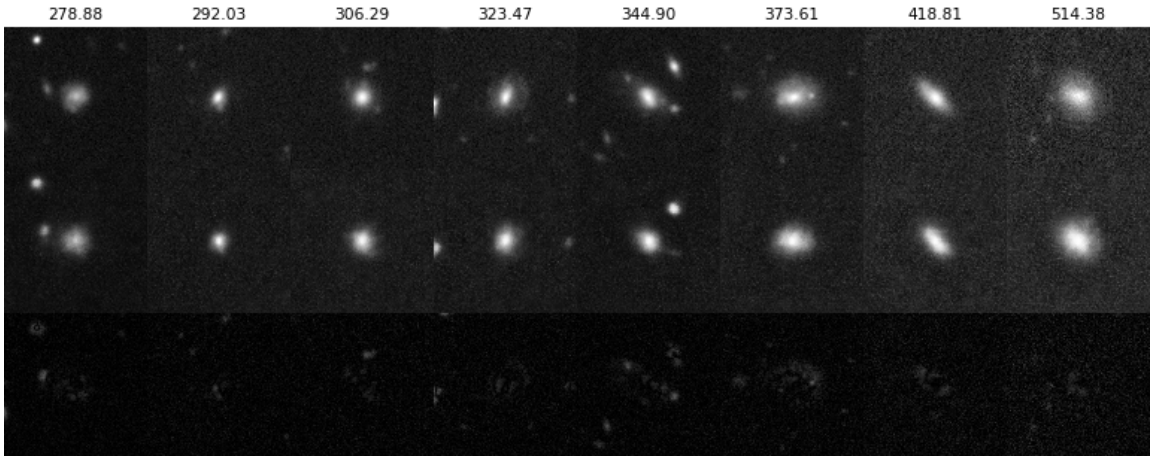
RESIDUALS



REAL

RECONSTRUCTED

RESIDUALS

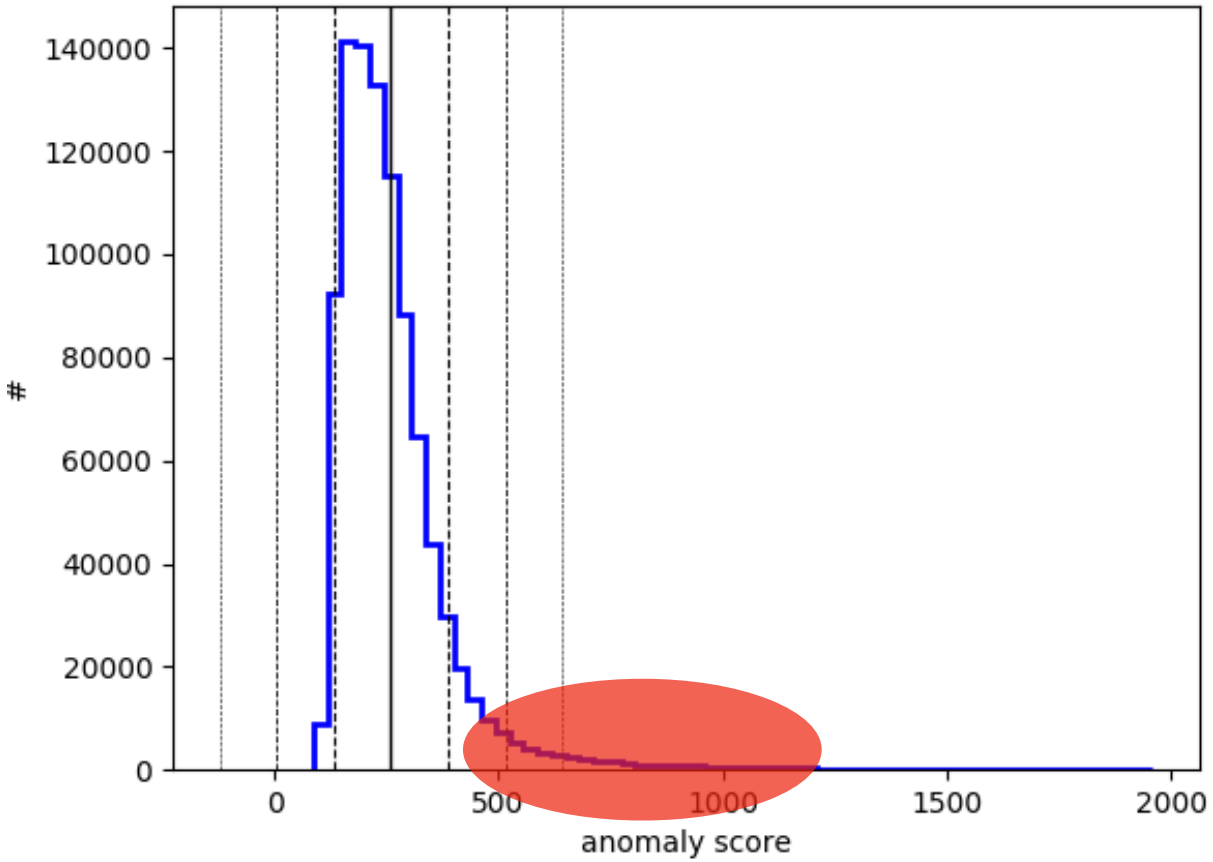


REAL

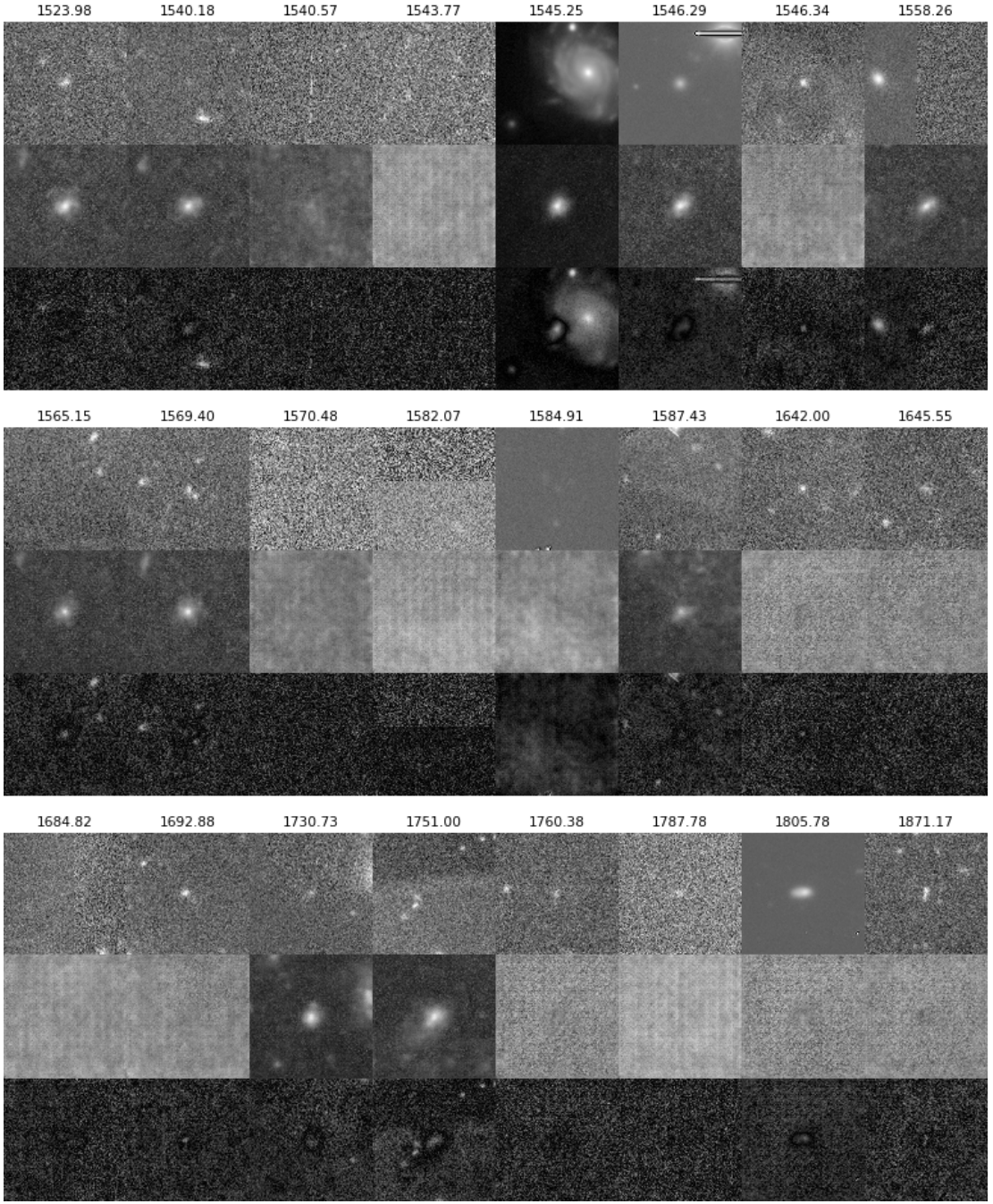
RECONSTRUCTED

RESIDUALS

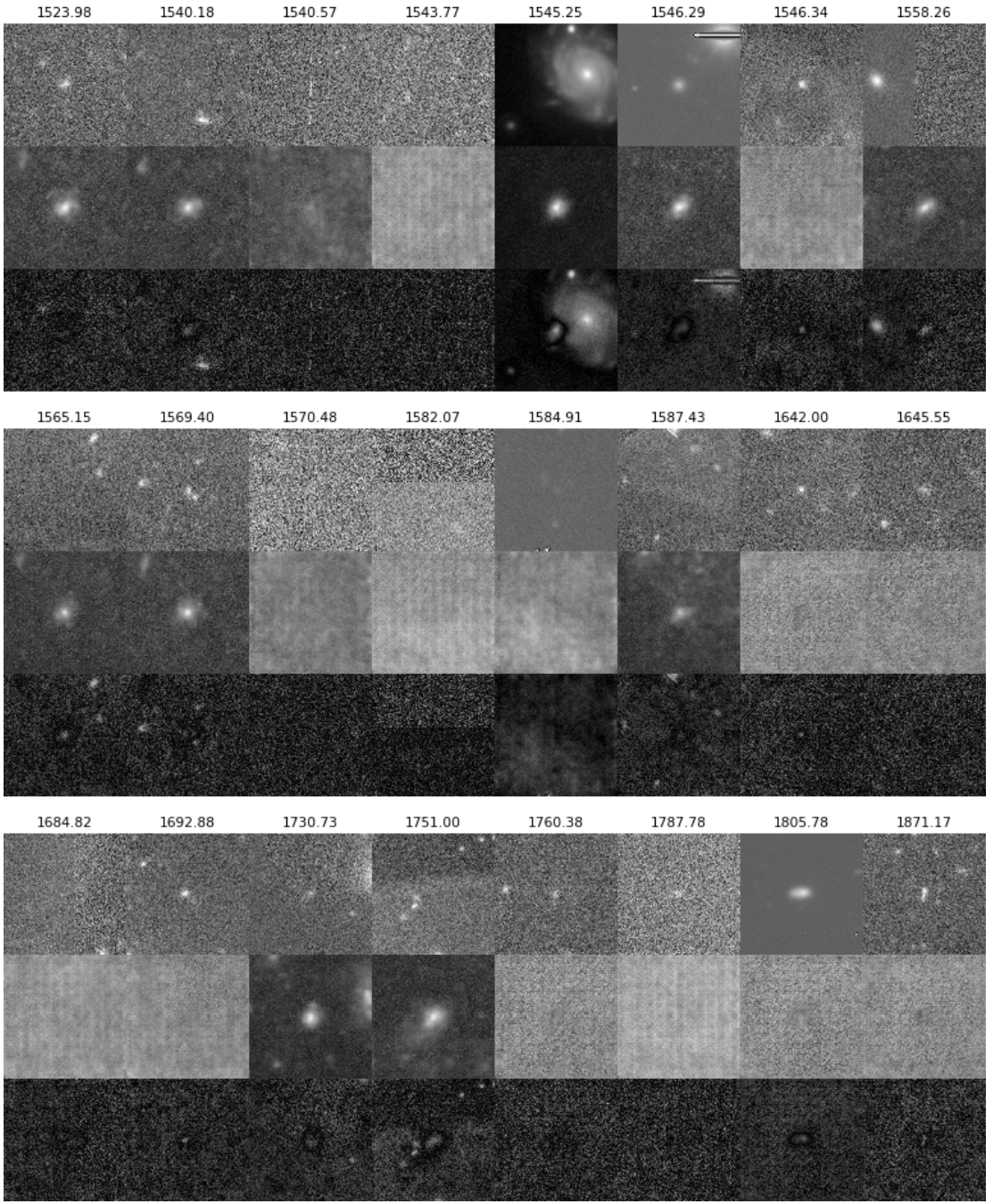
“ANOMALY SCORE” DISTRIBUTION FOR HSC GALAXIES

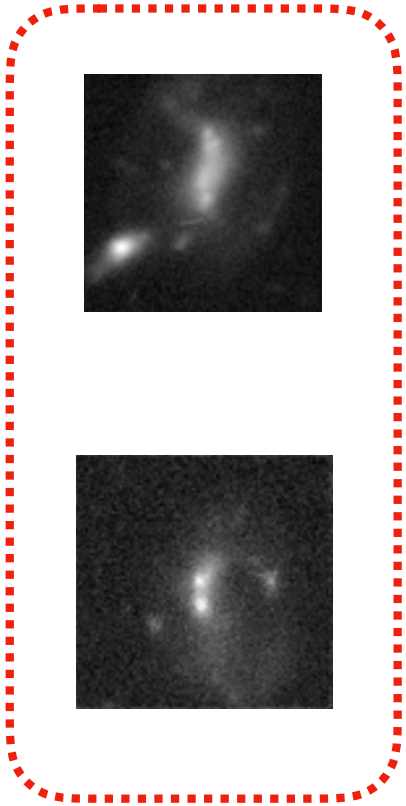


LARGER ANOMALIES ARE TYPICALLY PIPELINE ERRORS….



LARGER ANOMALIES ARE TYPICALLY PIPELINE ERRORS….





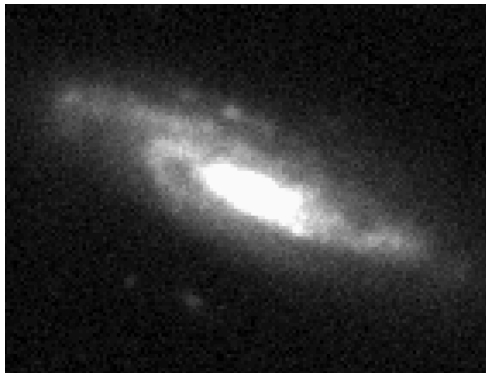
COMPARING SIMULATIONS OF GALAXY FORMATION WITH OBSERVATIONS WITH REGRESSIVE FLOWS

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

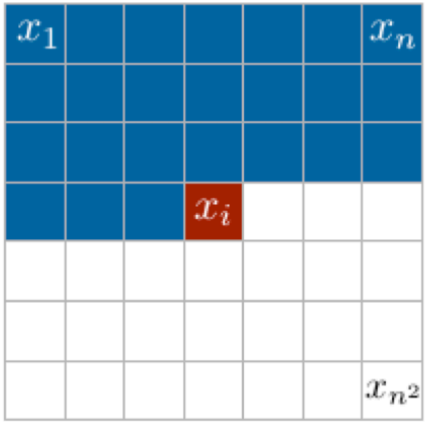
SDSS DR7 DATASET

Log(M*)>10
0.02<Z<0.08

~ 100,000 galaxies



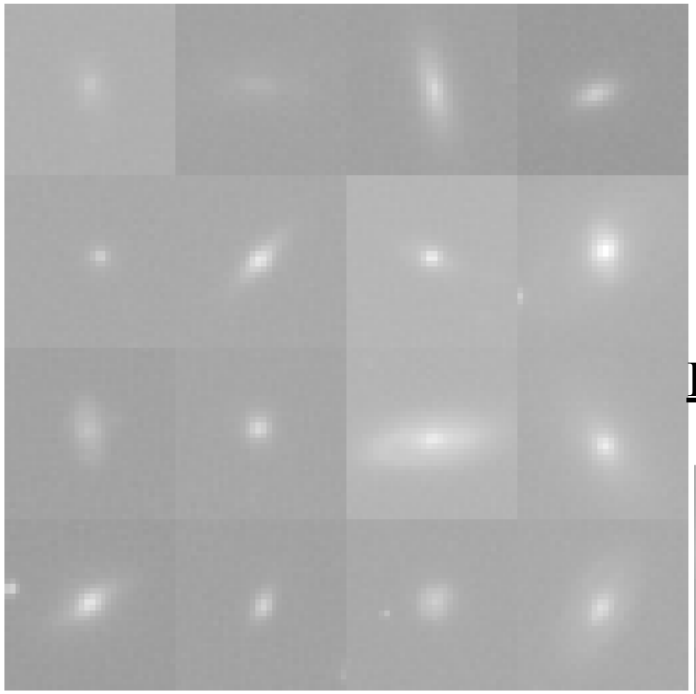
r-band images



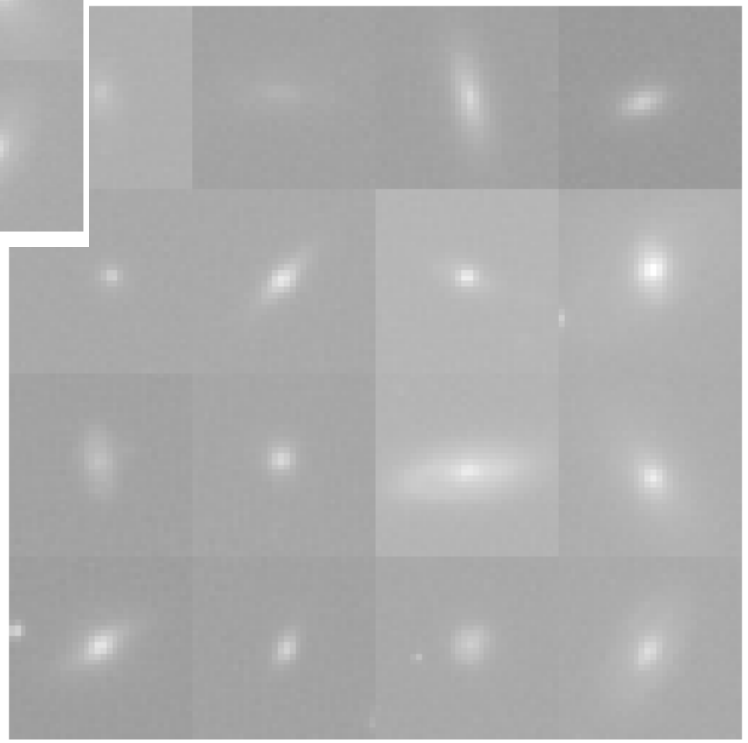
pixelCNN

$$p(x_0, x_1, \ldots, x_{n^2} | \theta^r_{SDSS})$$

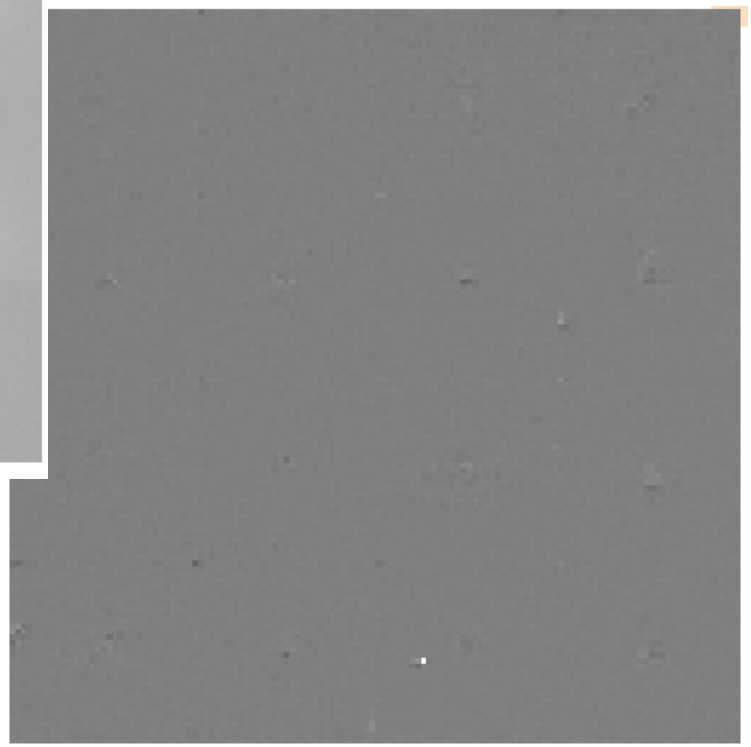
ORIGINAL SDSS



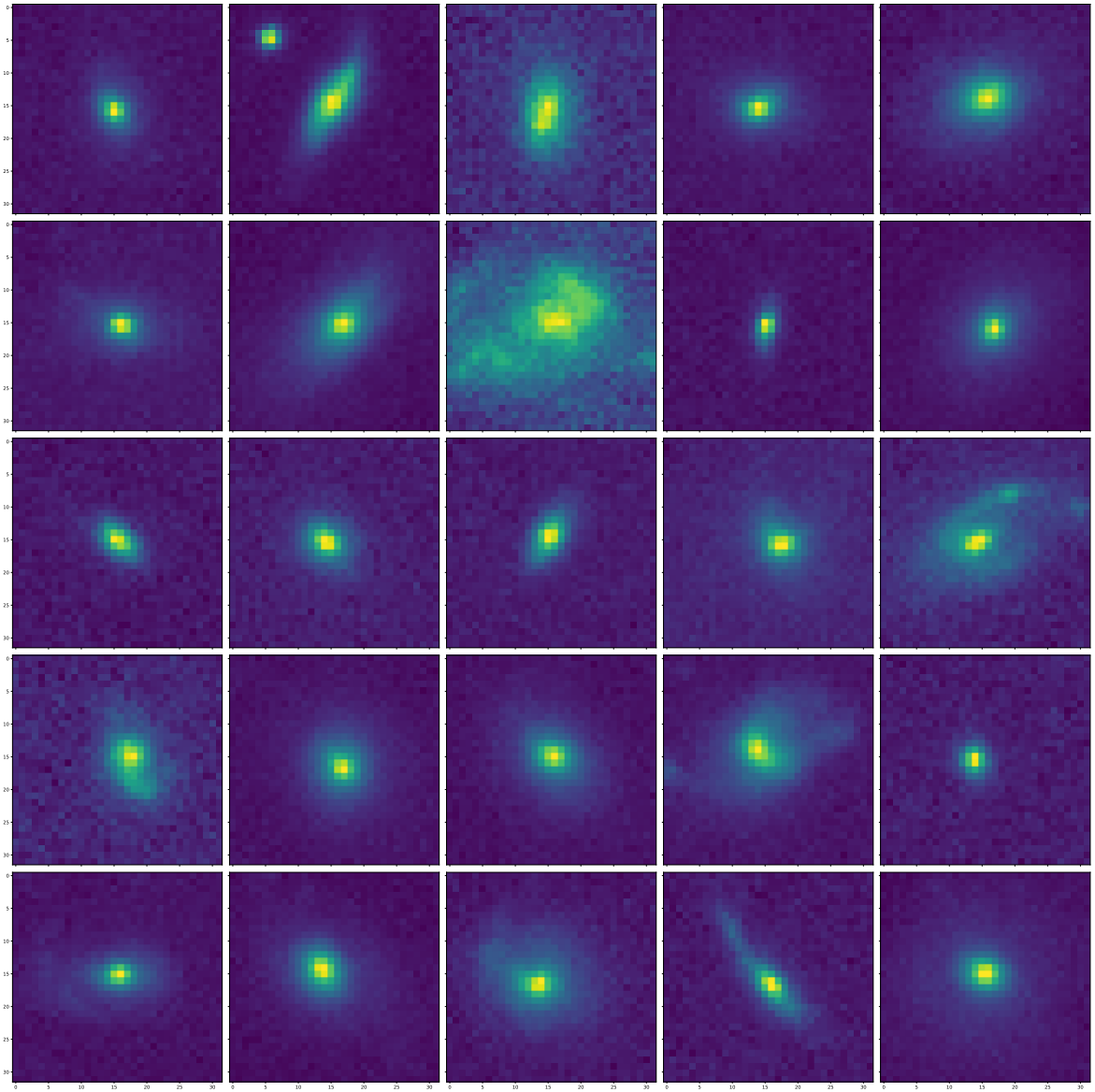
RECONSTRUCTED WITH PIXELCNN



RESIDUALS

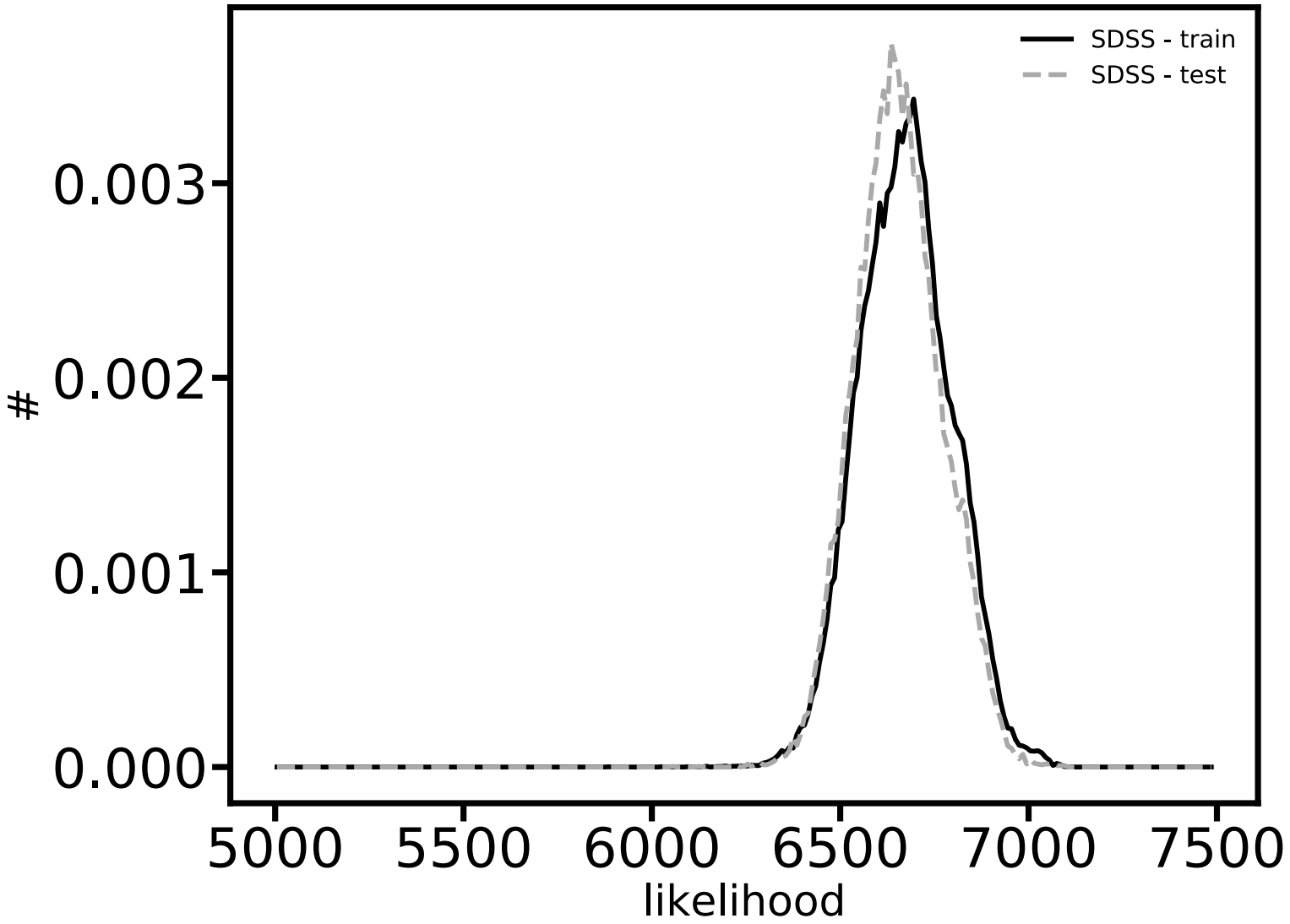


RANDOM SAMPLES FROM THE MODEL θ^r_{SDSS}

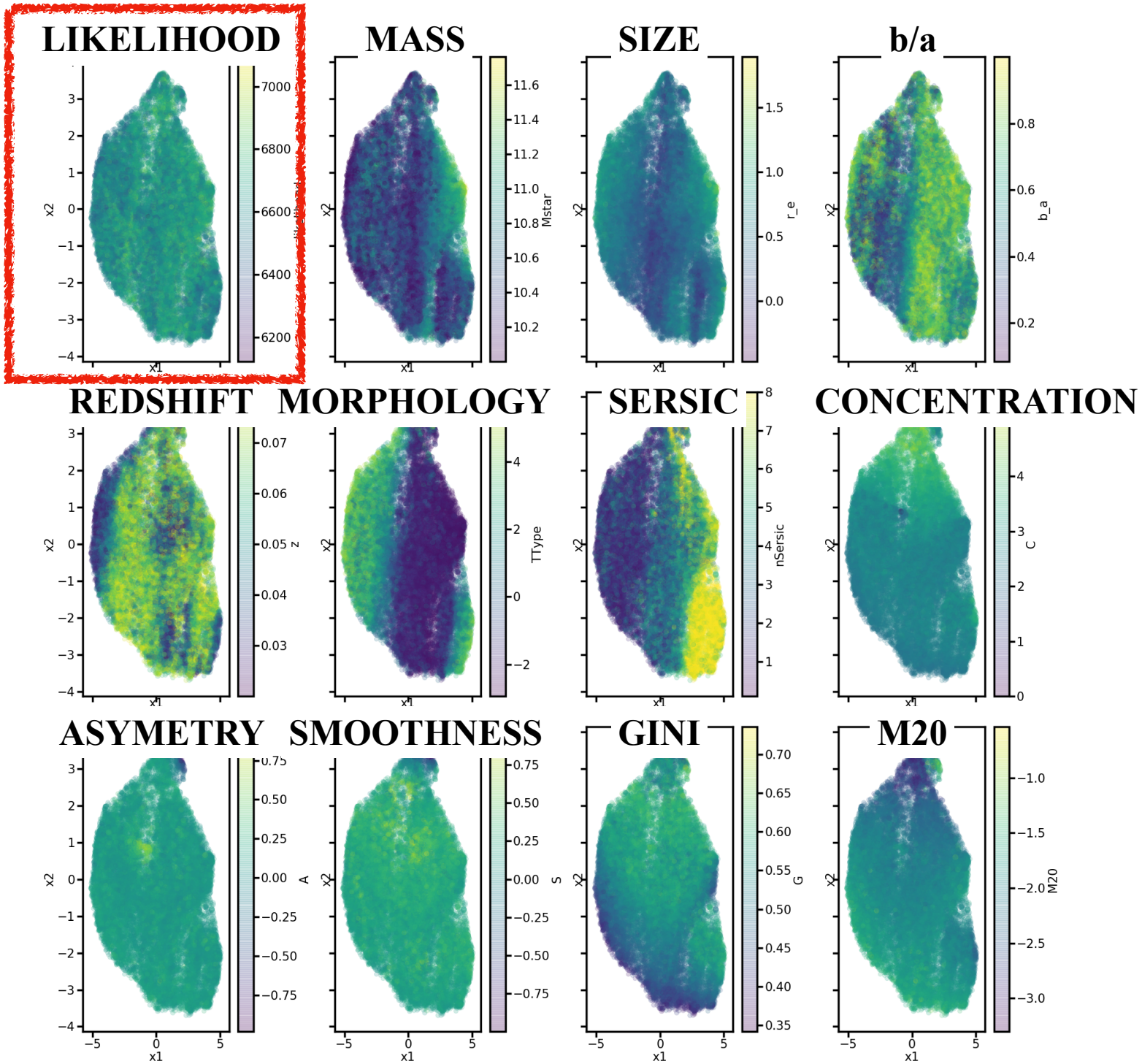


“FAKE” RANDOM SDSS GALAXIES OBTAINED THROUGH SAMPLING OF THE PDFs

DISTRIBUTION OF p(x) for SDSS GALAXIES



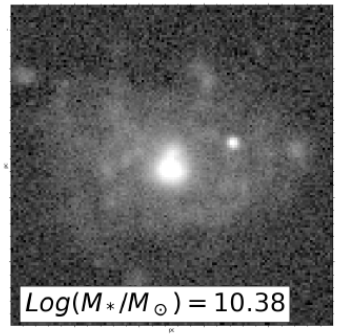
WHAT MAKES AN SDSS GALAXY "MORE LIKELY" FOR THE MODEL?

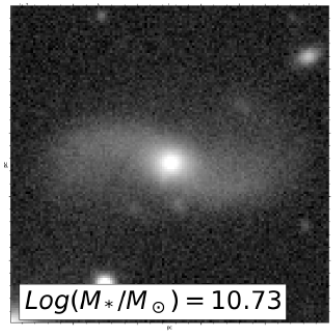


ILLUSTRIS

TNG

(z=0.05, Log(M*>10))





MOCK SDSS IMAGES

pixelCNN trained on SDSS

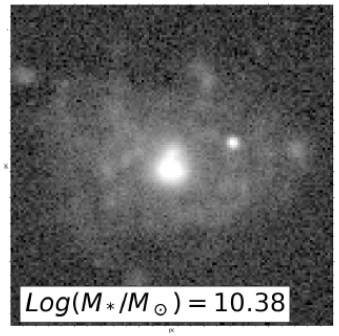
$$p(x_{ILL}|\theta^r_{SDSS})$$

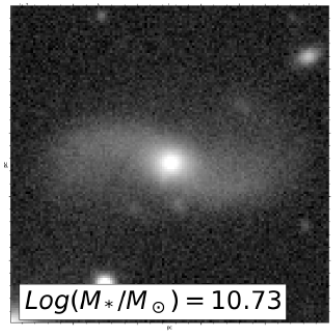
$$p(x_{TNG}|\theta^r_{SDSS})$$

ILLUSTRIS

TNG

(z=0.05, Log(M*>10))



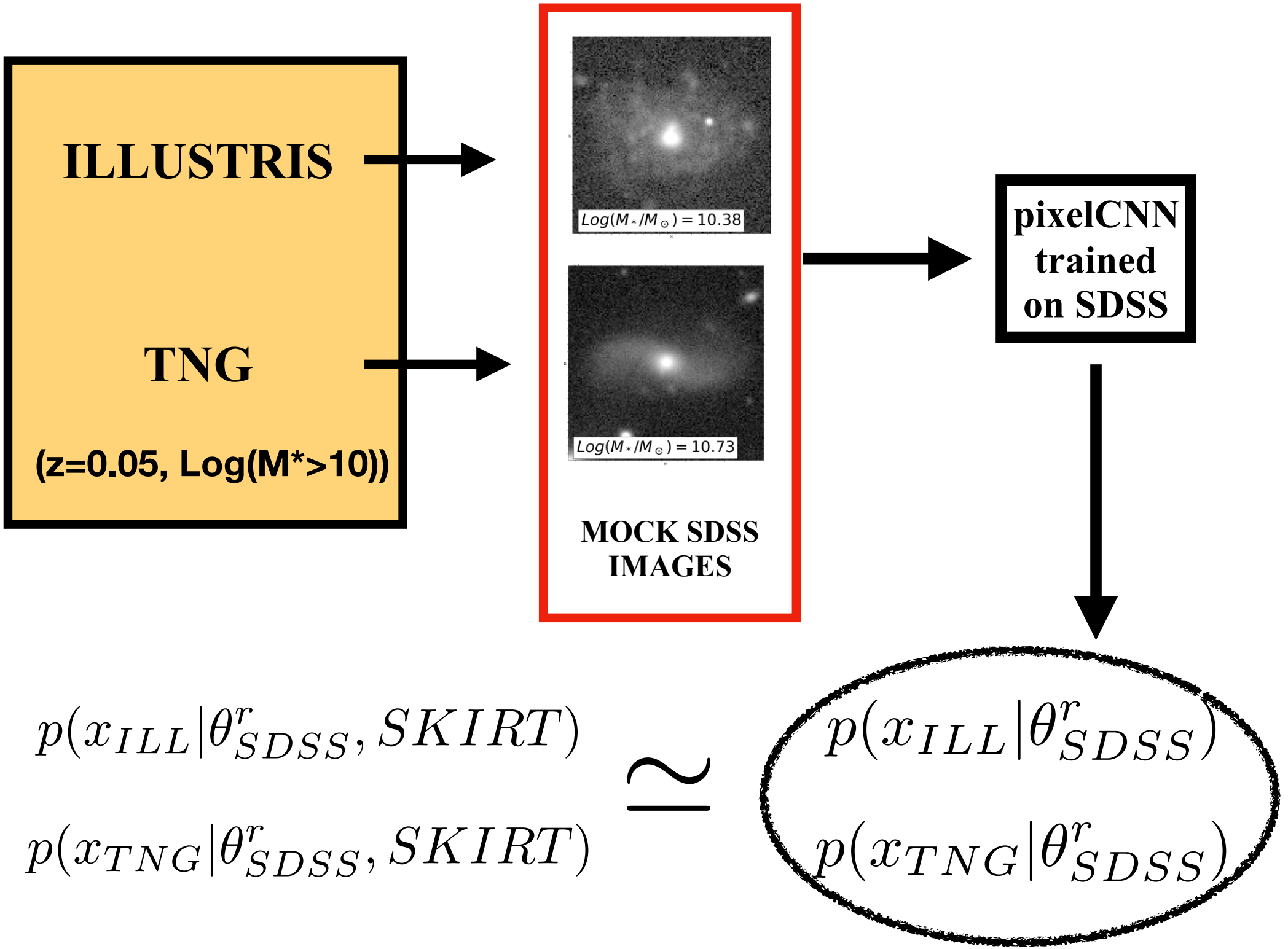


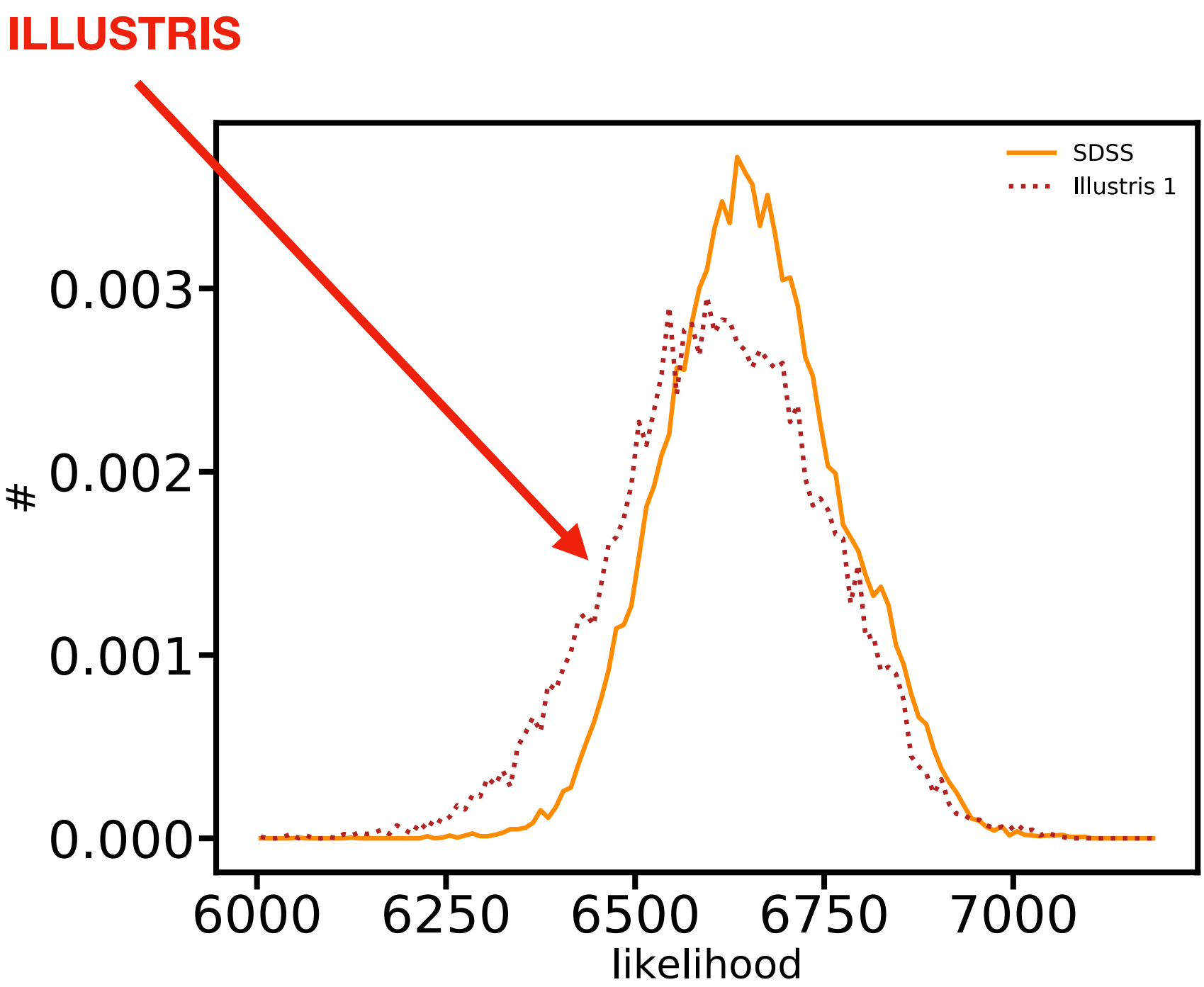
MOCK SDSS IMAGES

pixelCNN trained on SDSS

$$p(x_{ILL}|\theta^r_{SDSS})$$

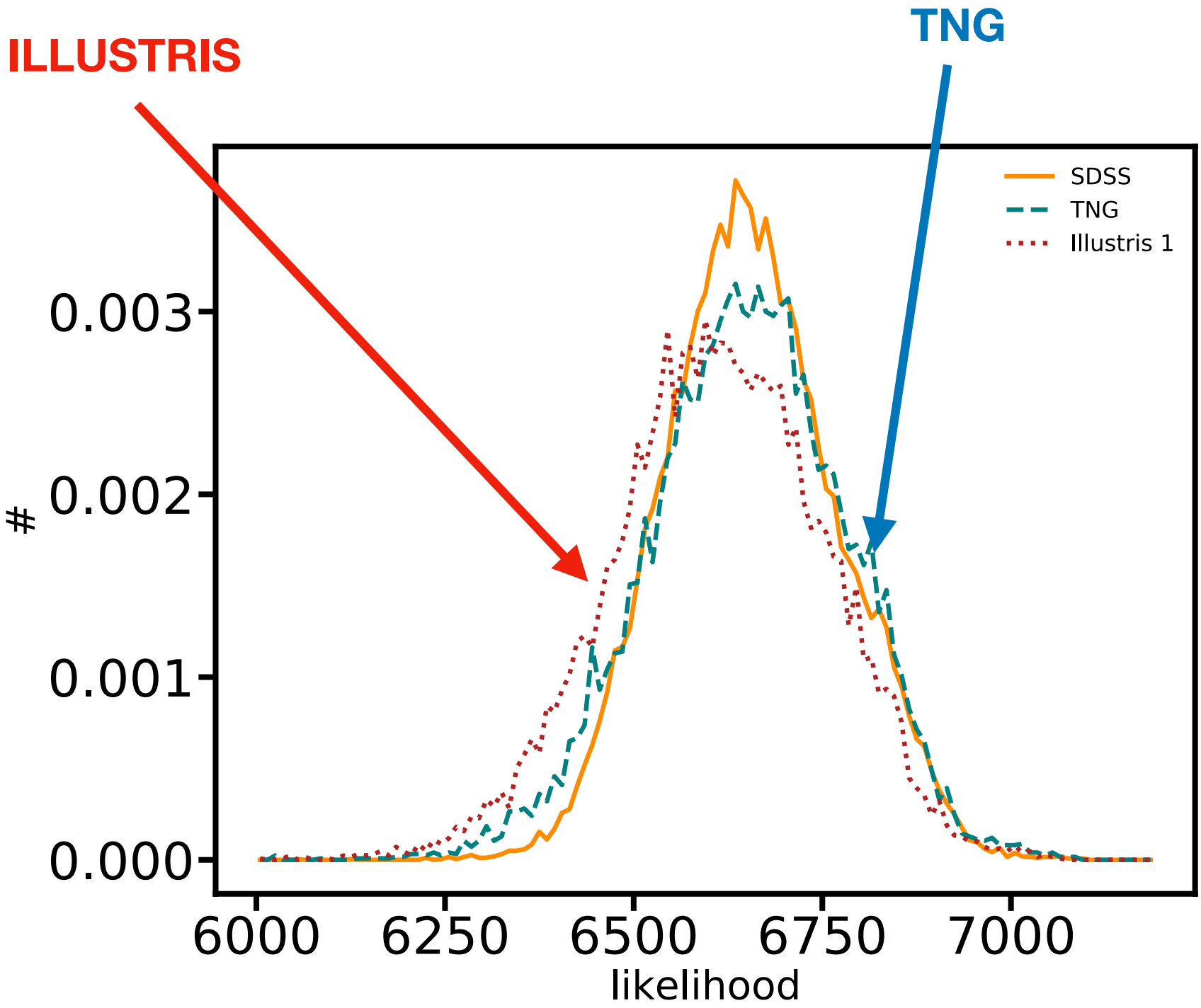
$$p(x_{TNG}|\theta^r_{SDSS})$$


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
$p(x_{ILL}|\theta^r_{SDSS}, SKIRT)$
$p(x_{TNG}|\theta^r_{SDSS}, SKIRT)$
$\simeq$
$p(x_{ILL}|\theta^r_{SDSS})$
$p(x_{TNG}|\theta^r_{SDSS})$


ILLUSTRIS
SDSS
Illustris 1
0.003
0.002
0.001
0.000
#
6000
6250
6500
6750
7000
likelihood

ILLUSTRIS

SDSS
Illustris 1
0.002
0.001
0.000
6000
6250
6500
6750
7000
likelihood


ILLUSTRIS
TNG
SDSS
TNG
Illustris 1
0.003
0.002
0.001
0.000
#
6000
6250
6500
6750
7000
likelihood