Unsupervised Learning: Generative models and anomaly detection

Gregor Kasieczka
Email: gregor.kasieczka@uni-hamburg.de
Twitter: @GregorKasieczka
Mastodon: @gregor_k@sciencemastodon.com

ÖAW AI Winter School 2023

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

PUNCH4NFDI

DASHH

Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CDCS
CENTER FOR DATA AND COMPUTING IN NATURAL SCIENCES

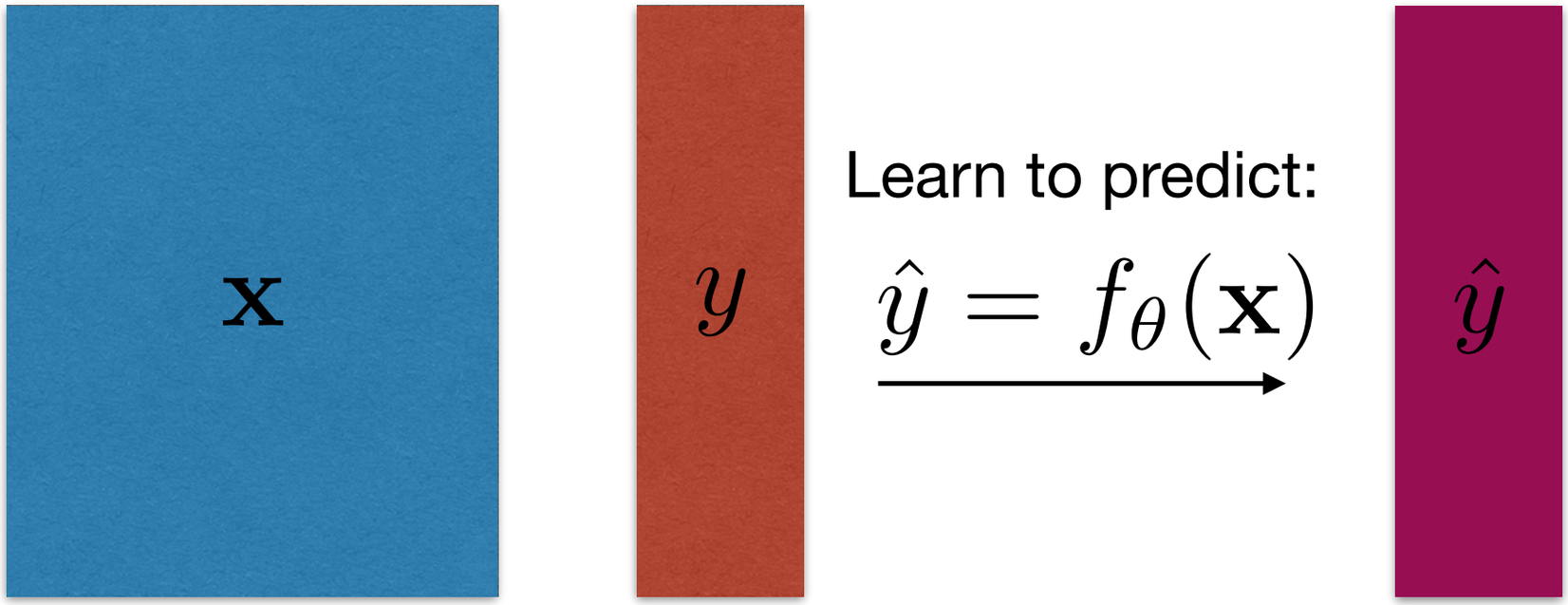
PIER
Partnership of Universität Hamburg and DESY

Bundesministerium für Bildung und Forschung

Loss function: Supervised

Supervised Learning:
Attempt to infer some target *(truth label)*:
classification, regression (often also clustering/inference)

Use training data with known labels
(often from Monte Carlo simulation)



observable features such as kinematics, tracks,…

truth label
(e.g. true energy)

predicted energy

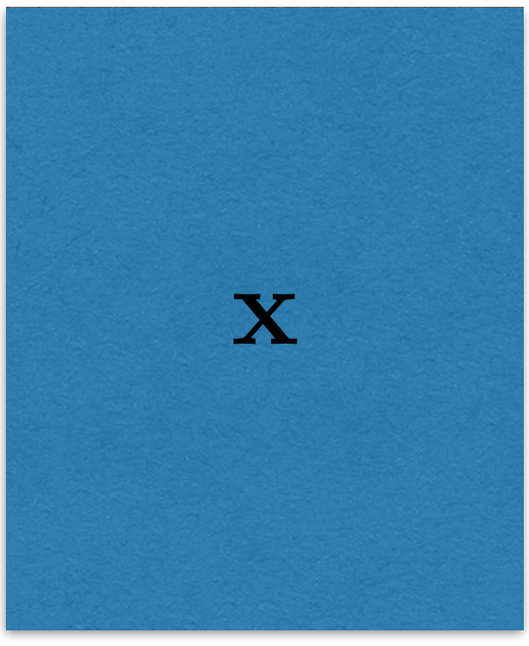
Regression: Minimize mean squared error:

$$\mathcal{L} = (y - \hat{y})^2$$

Loss function: Unsupervised

Unsupervised Learning:
No target, learn the probability distribution (directly from data)

Can use for sampling, anomaly detection, unfolding, …



Learn to predict:

$$\hat{p}(\mathbf{x}) = f_\theta(\mathbf{x})$$

$$p(\mathbf{x})$$

True probablity density

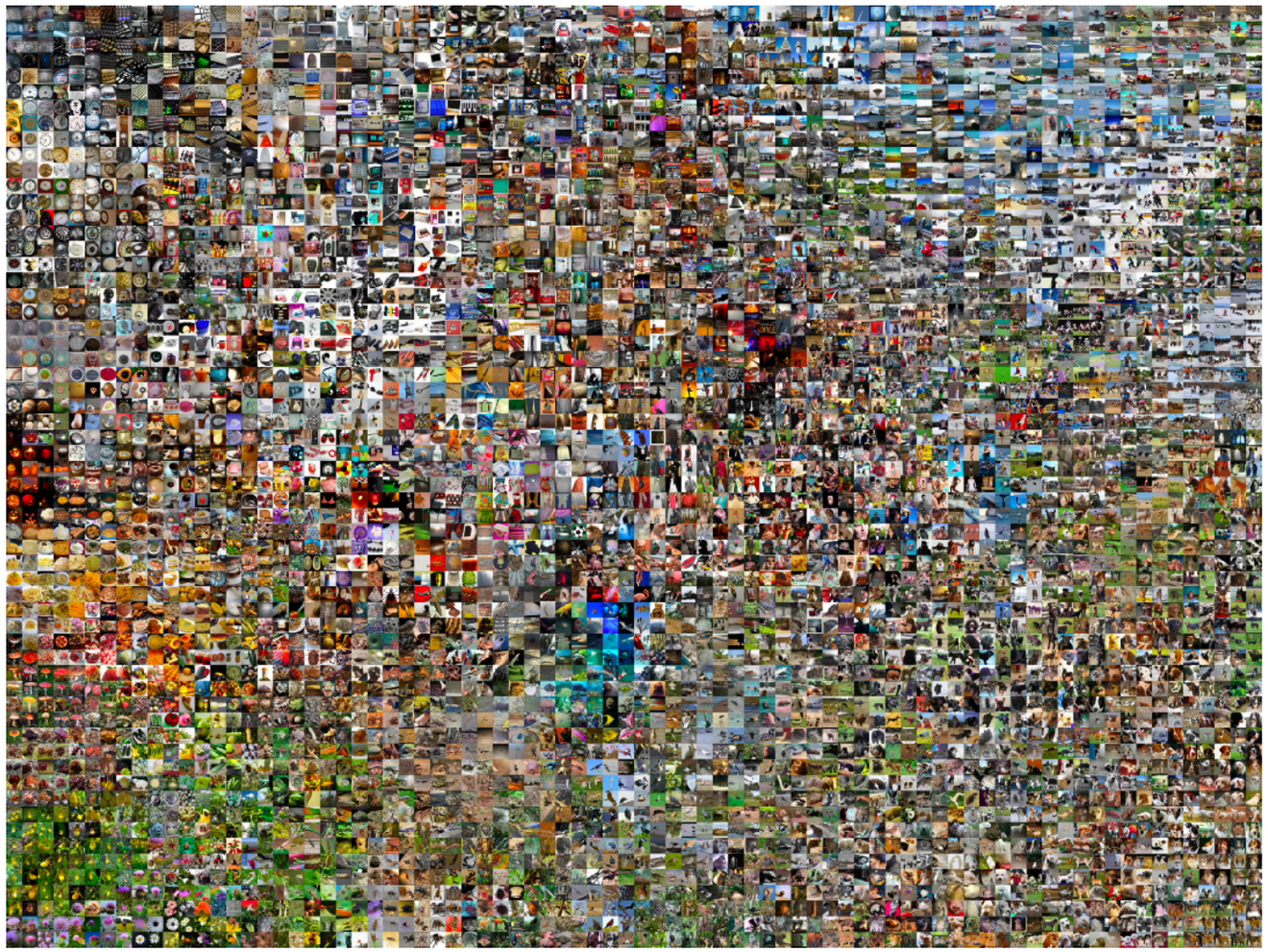
Distribution learning: Maximise likelihood (minimize log-likelihood):
(either directly or with approximations)

$$\mathcal{L} = -\log\left(\hat{p}(\mathbf{x})\right)$$

Part III: Generative Models

Motivation

We **have**:
many images
(or collision events,
or detector readouts, …)



We **want**: more “images”

(Specifically: New examples that are similar to the examples, but not exact copies)

How to encode in neural net?

Uses:

- Fast approximative simulation
- Differentiable surrogate models

Examples

https://www.thispersondoesnotexist.com/

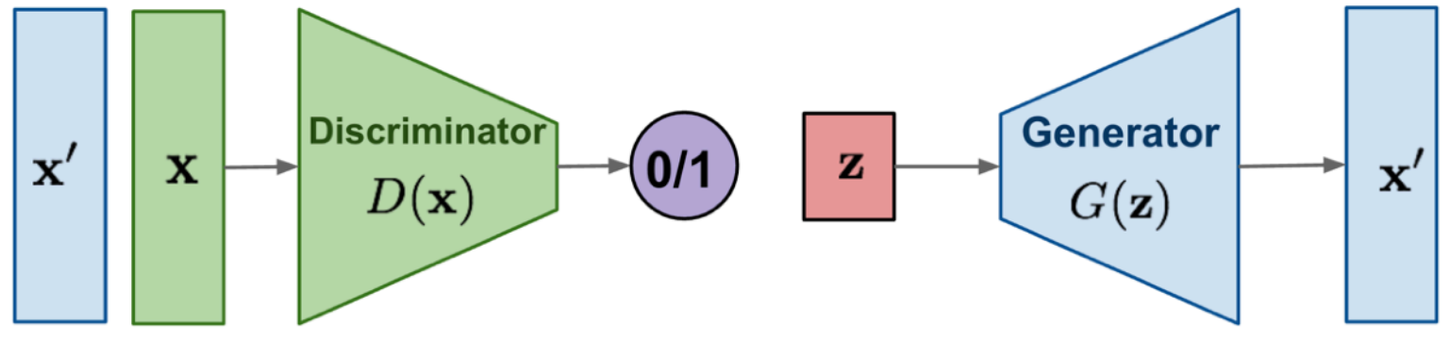
Examples

DALL-E 2

Generative Adversarial Networks

Generative models

GAN: Adversarial training

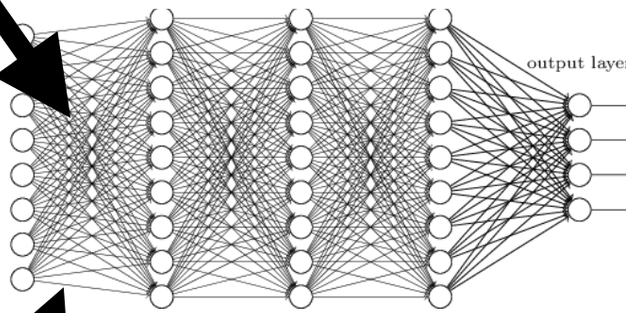


GAN Training

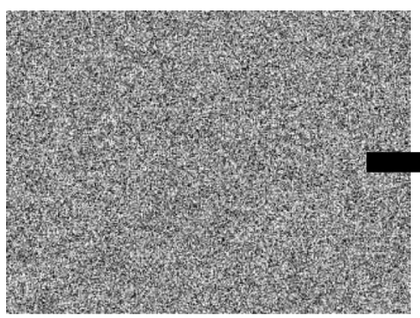
Train Generator
(Freeze Discriminator)
Then
Train Discriminator
(Freeze Generator)
Repeat

Real Images

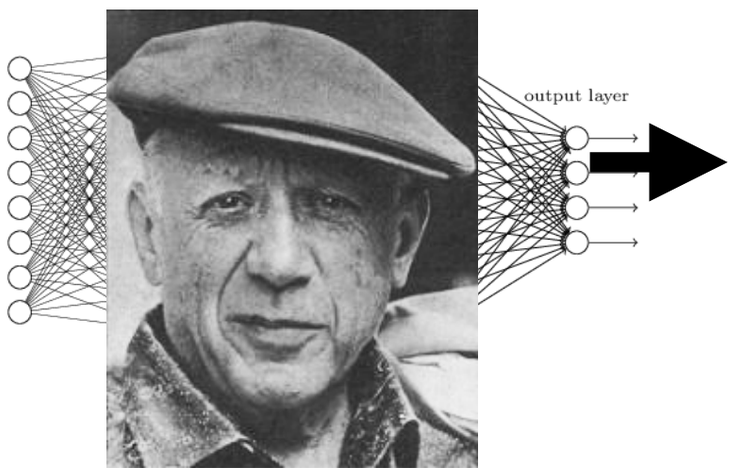
Discriminator



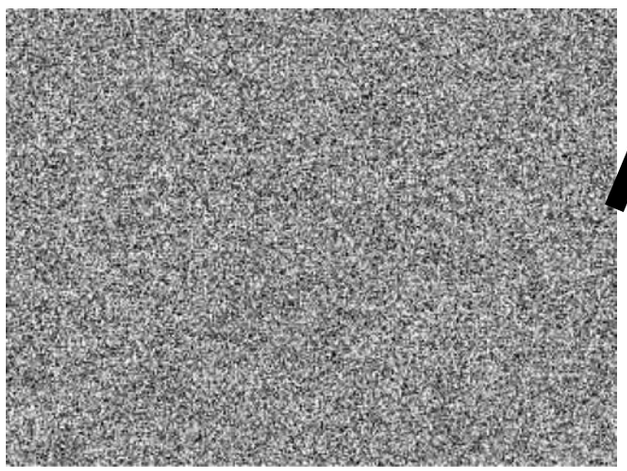
Noise



Generator



Fake Images

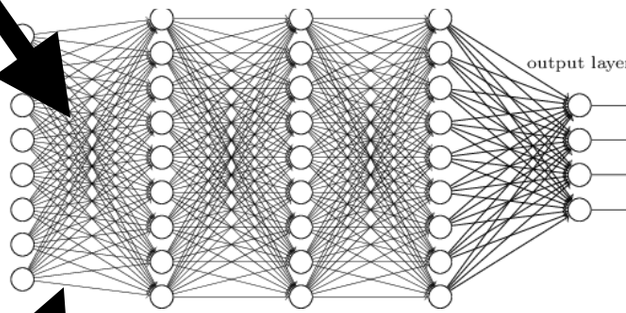


GAN Training

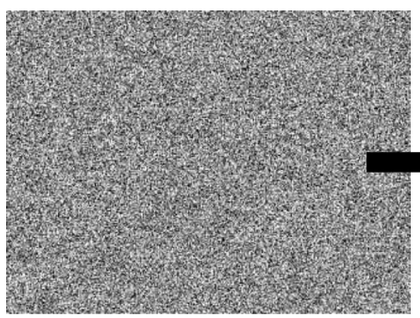
Train Generator
(Freeze Discriminator)
Then
Train Discriminator
(Freeze Generator)
Repeat

Real Images

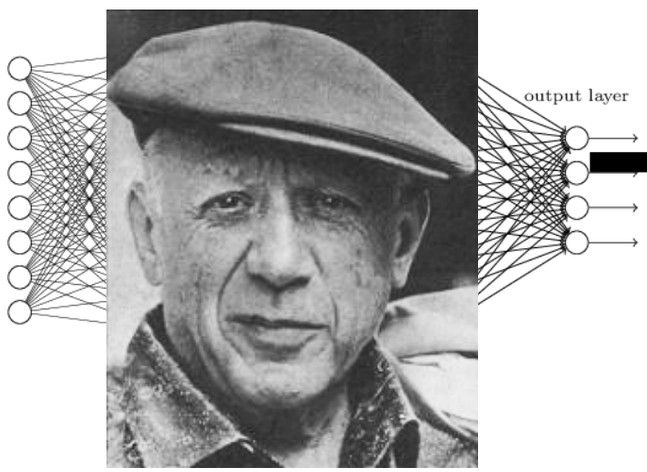
Discriminator



Noise



Generator



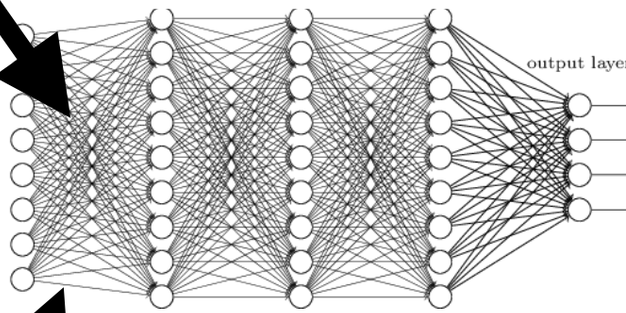
Fake Images

GAN Training

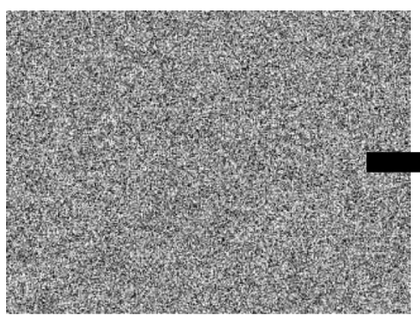
Train Generator
(Freeze Discriminator)
Then
Train Discriminator
(Freeze Generator)
Repeat

Real Images

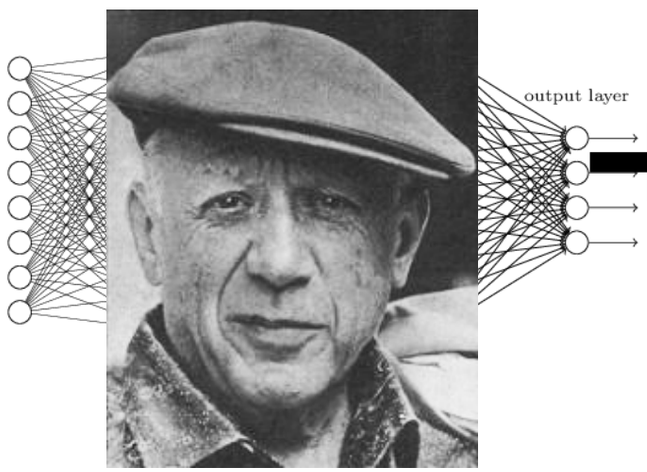
Discriminator



Noise



Generator



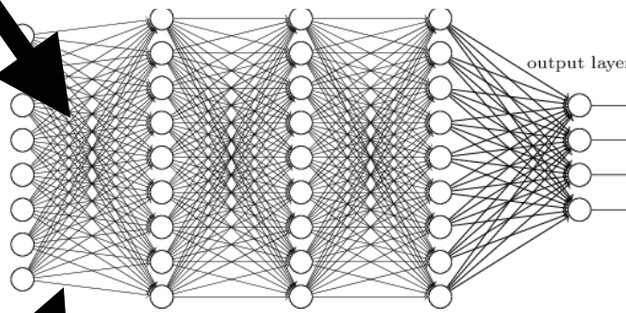
Fake Images

GAN Training

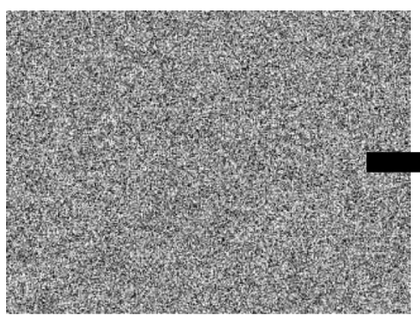
Train Generator
(Freeze Discriminator)
Then
Train Discriminator
(Freeze Generator)
Repeat

Real Images

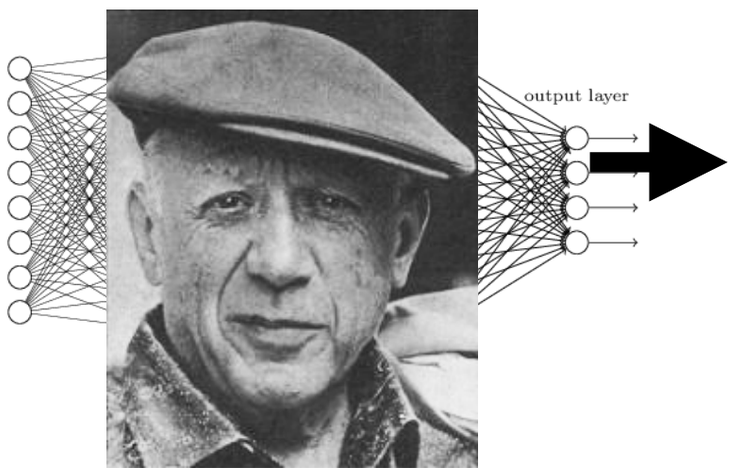
Discriminator



Noise

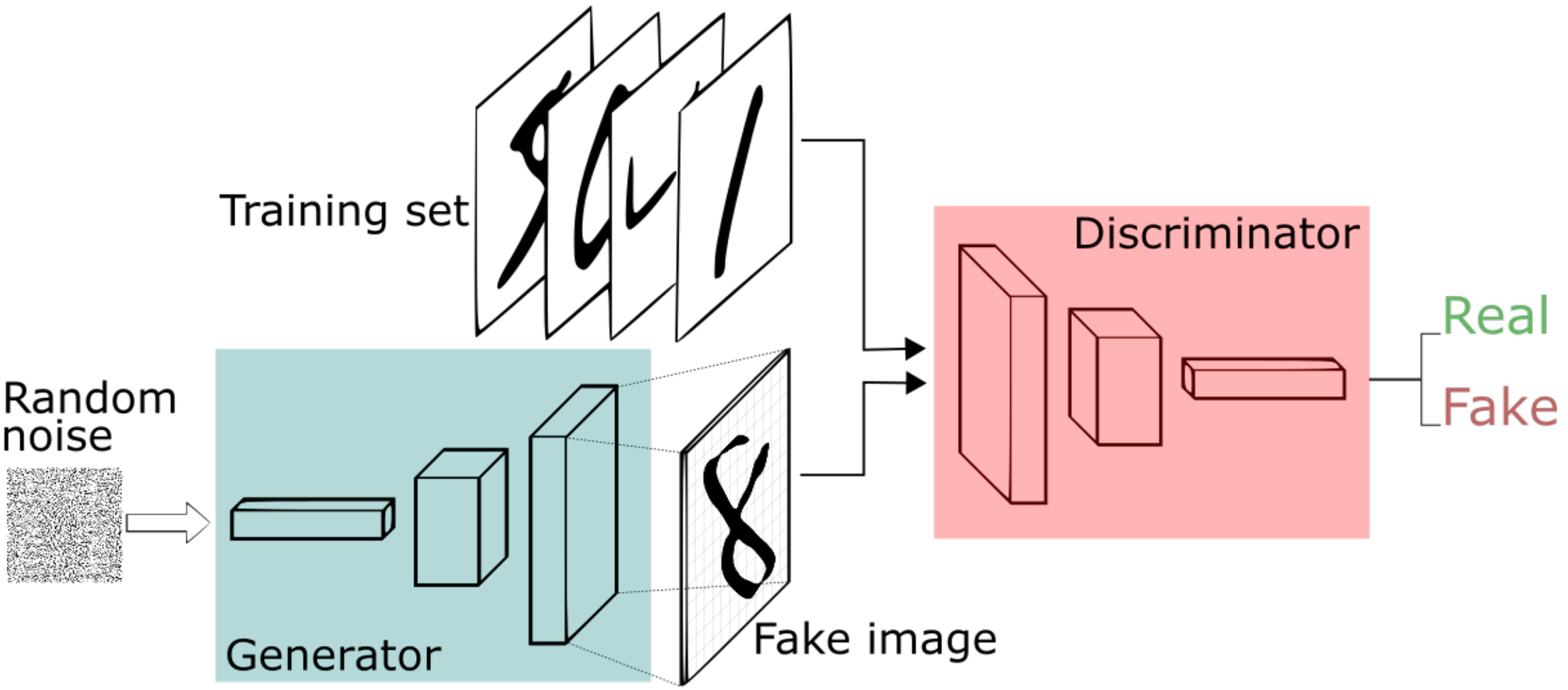


Generator



Fake Images

Structure



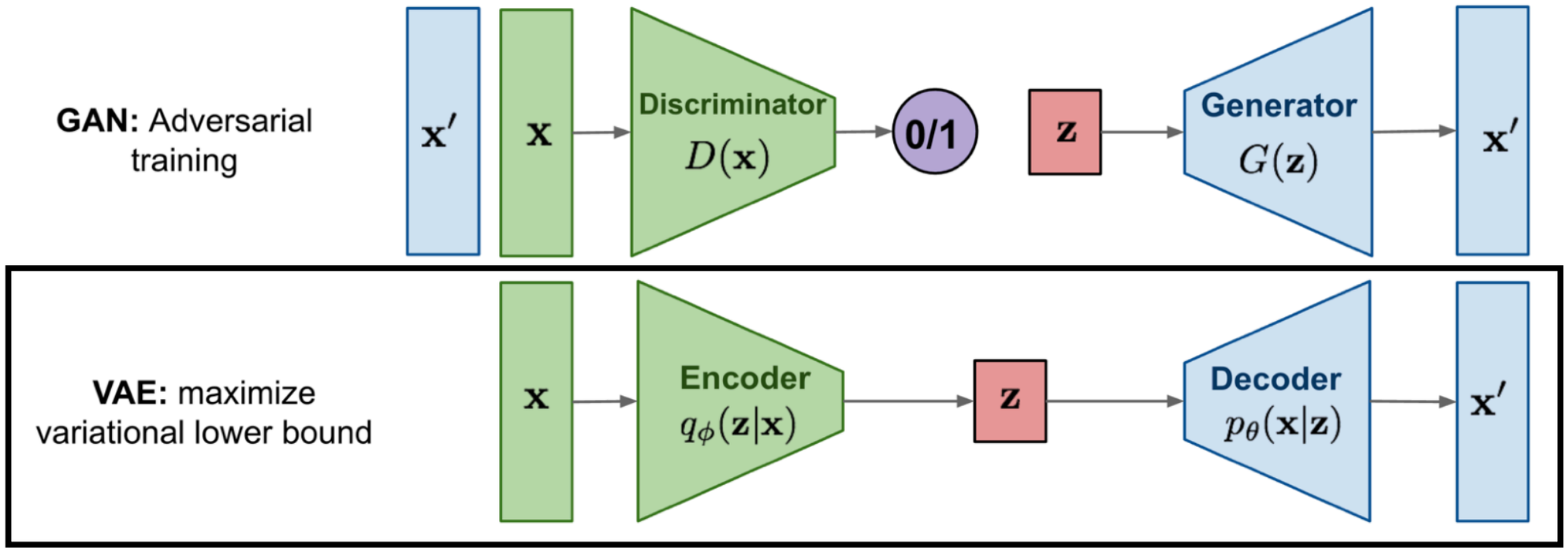
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Real Samples

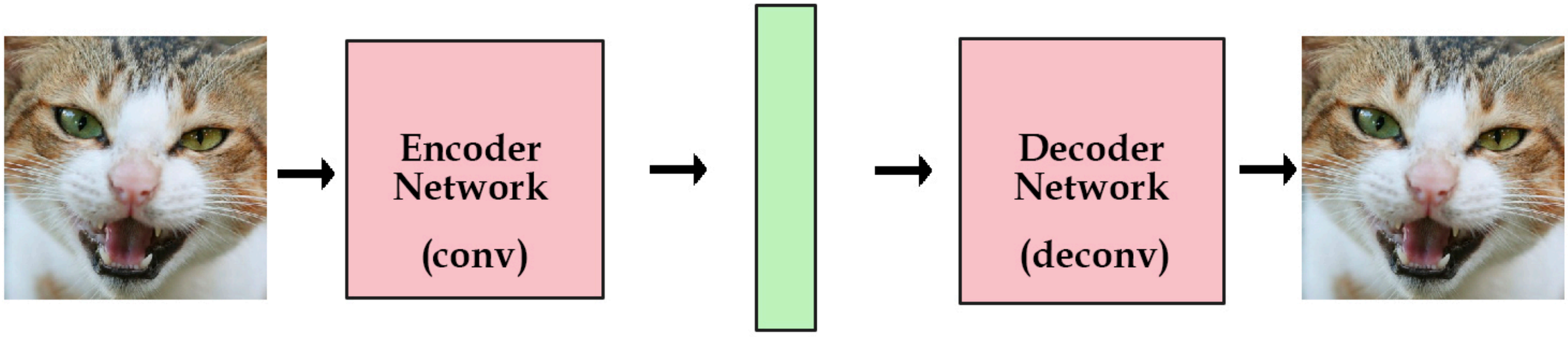
Generated Fake Samples

Generative Adversarial Networks
IJ Goodfellow et al
1406.2661

Generative models



Autoencoder

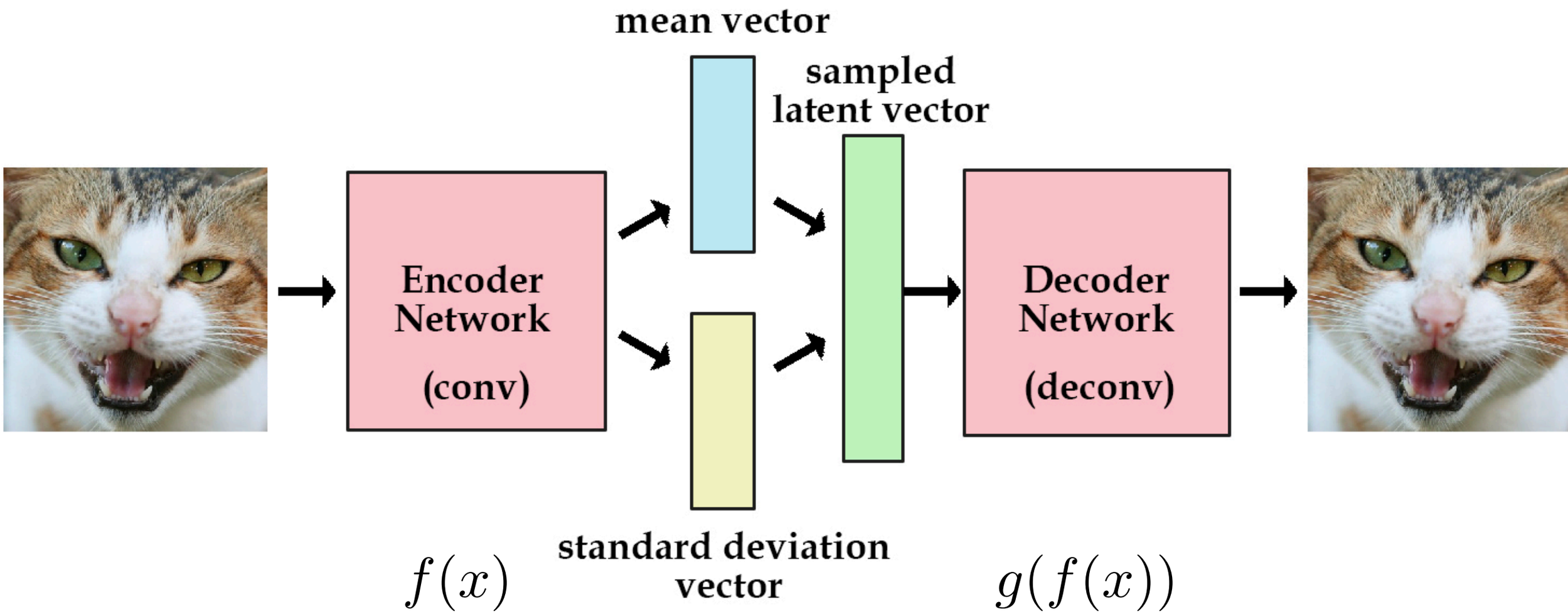


$f(x)$ **latent vector / variables** $g(f(x))$

$$L = (\hat{y} - g(f(x)))^2$$

- *Latent space/bottleneck* with compressed representation
- Dimension reduction
- Denoising
- Will encounter again soo for anomaly detection

Variational Autoencoder



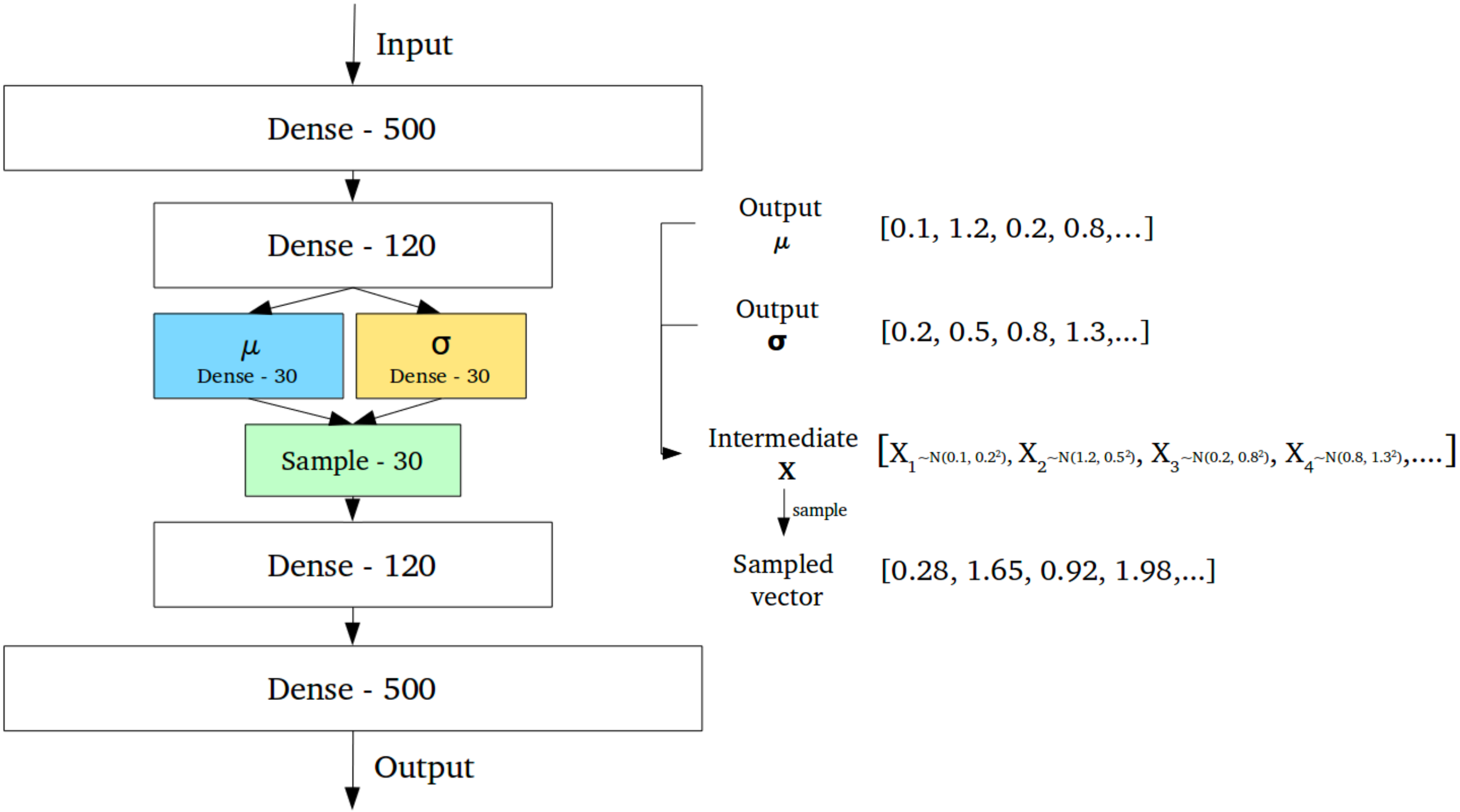
$f(x)$ $g(f(x))$

- Want to sample from latent space
- Split into mean and standard deviation
- Add penalty term (Kullback-Leibler divergence) so mean/std are close to unit Gaussian
- Compare to a flow:
  - Similar
  - Encoder/Decoder are not exact inverse of each other
  - Dimension of data space and latent space can differ

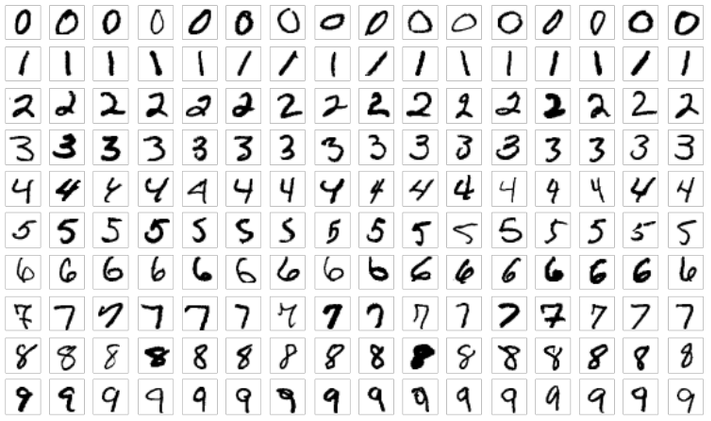
$$L = (x - g(f(x)))^2$$

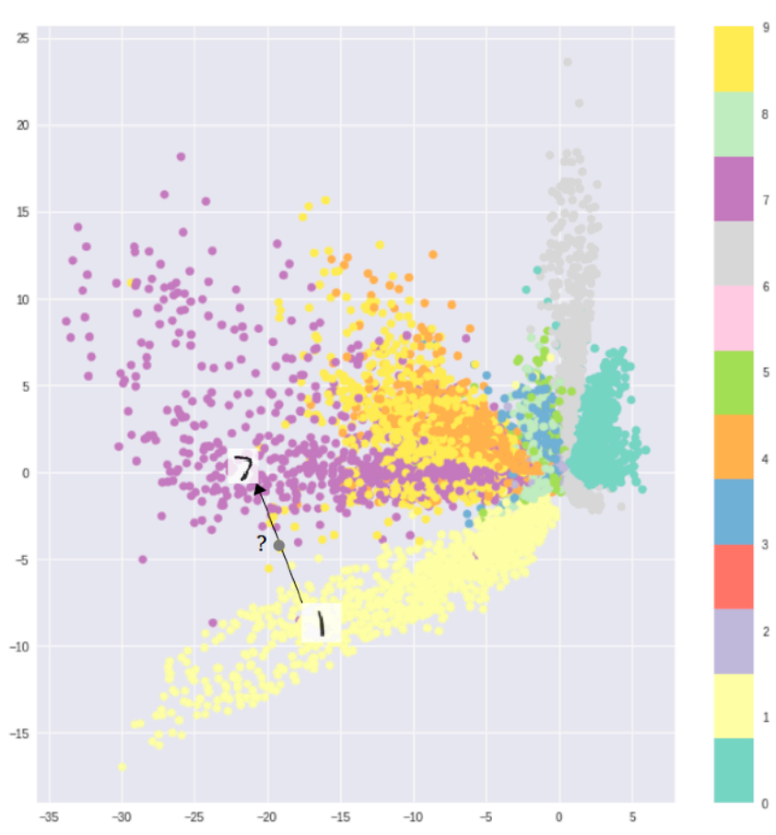
$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$

Concrete Example

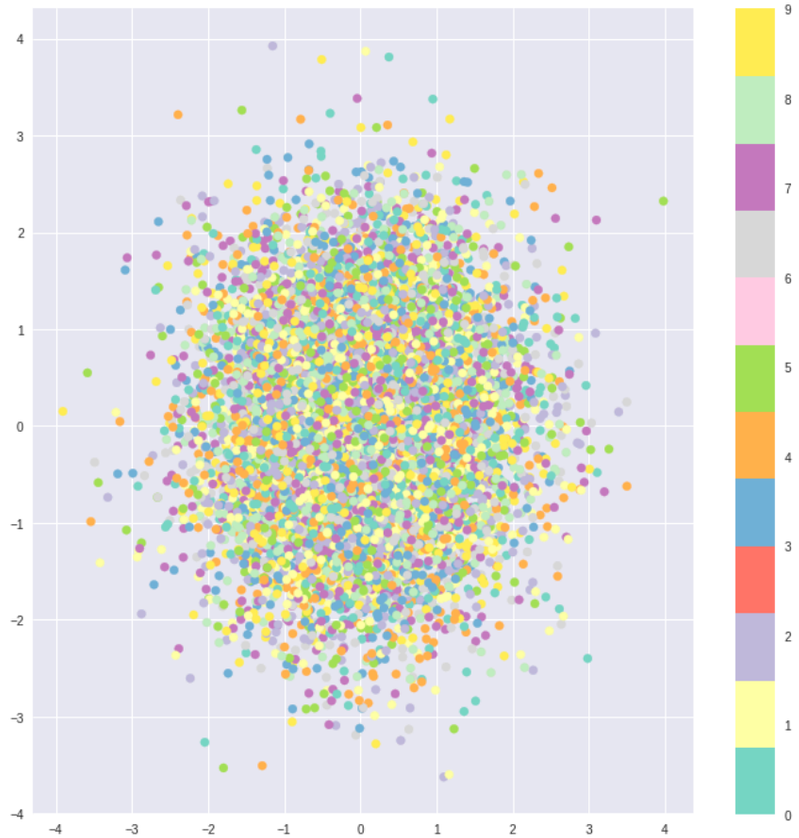


Variational Autoencoder

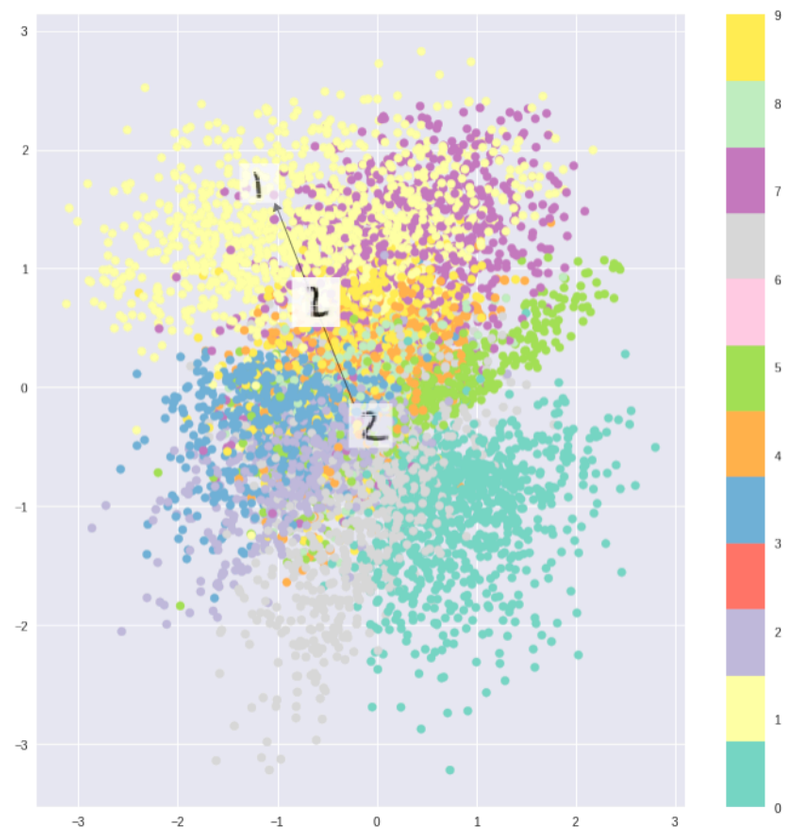




Reconstruction Loss Only



KL Loss Only



Combined Loss

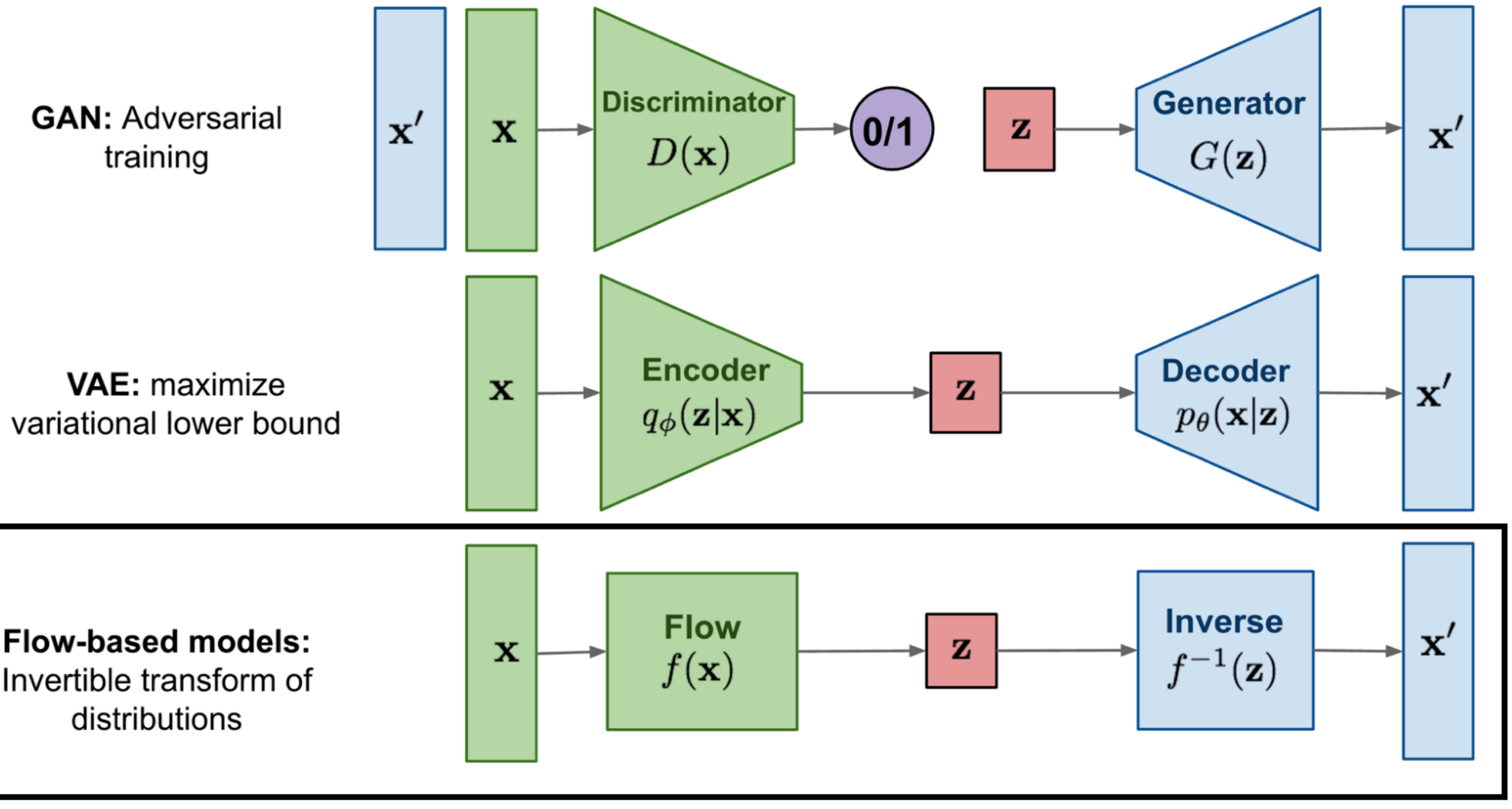
$$L = (\hat{y} - g(f(x)))^2$$

$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$

kvfrans
towardsdatascience.com

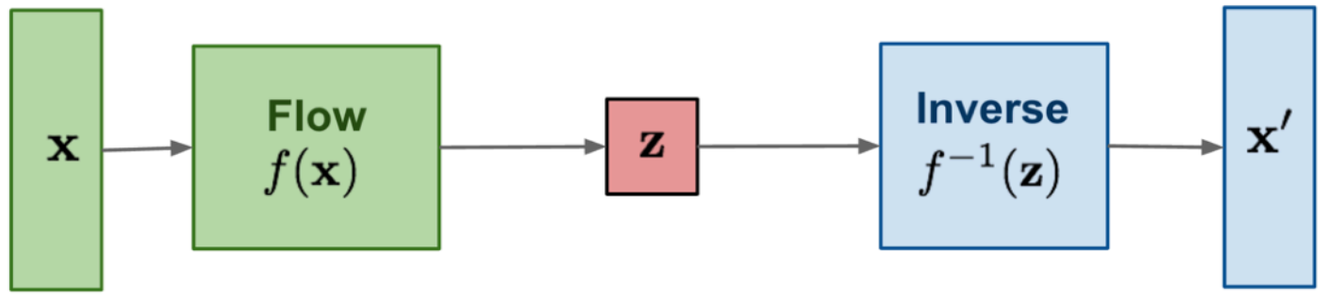
Normalising Flows

Generative models



Introduction to NF

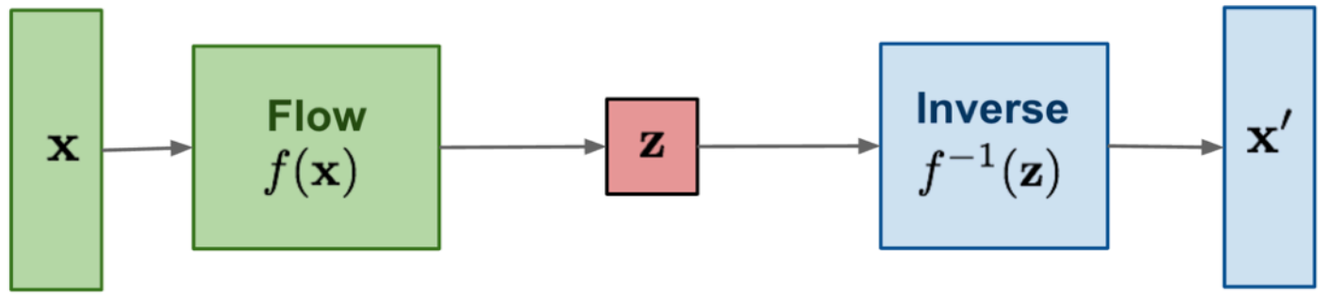
Flow-based models:
Invertible transform of distributions



- Basic idea: Learn a mapping between data and an intial latent-space distribution (e.g. Gaussians)
- Bijective, so that it is invertable
  (f^{-1} is not a learned approximated inversion, but the exact inverse of f by construction)
- Actually a diffeomorphism
- Take into account Jacobian determinant (change of prob. variable formula) to evaluate probability density in data space
  (need to construct f to allow easy calculation of Jacobian determinant)

Introduction to NF

Flow-based models:
Invertible transform of distributions

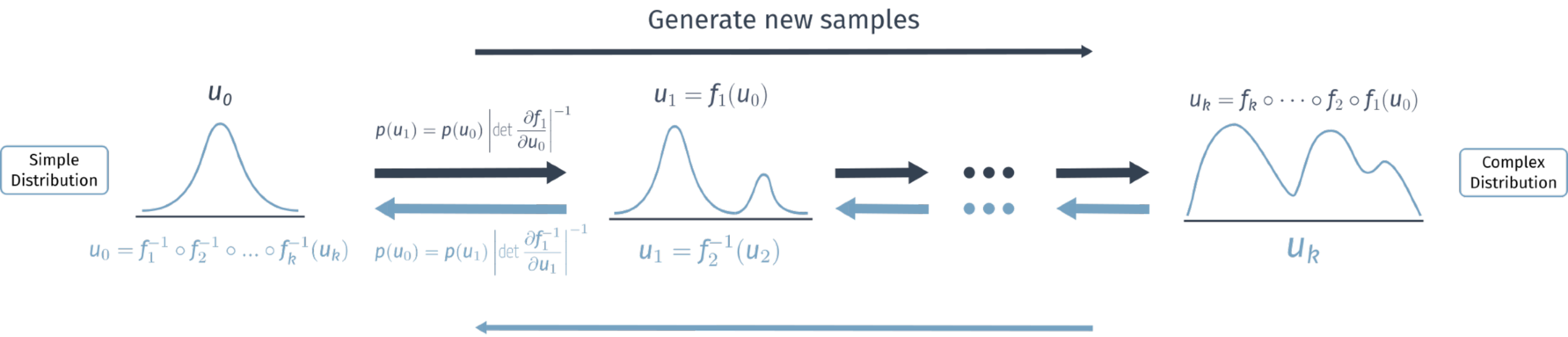


- Why could this be useful?
  - Can sample from latent space and transform with f^{-1} into data space for use as generative model
  - Can assign likelihood to data points by applying f

- Will see some physics applications later

- (See e.g. D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, International Conference on Machine Learning 37, 1530 (2015); I. Kobyzev, S. Prince, and M. Brubaker, Normalizing Flows: An Introduction and Review of Current Methods, IEEE Transactions on Pattern Analysisand Machine Intelligence , 1 (2020))

Introduction to NF



- Goal: assign probability density to each datapoint
- Learn bijective transformation between data and a latent space with tractable probability
- Build from simple invertible transformations with tractable Jacobian

$$p(\boldsymbol{x}) = p(\boldsymbol{f}^{-1}(\boldsymbol{x})) \prod_i \left| \det \left(\frac{\partial \boldsymbol{f}_i^{-1}}{\partial \boldsymbol{x}} \right) \right| =$$

$$p(\boldsymbol{u}) \prod_i \left| \det \left(\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{u}} \right) \right|^{-1}$$

Thanks to T. Loesche

Introduction to NF

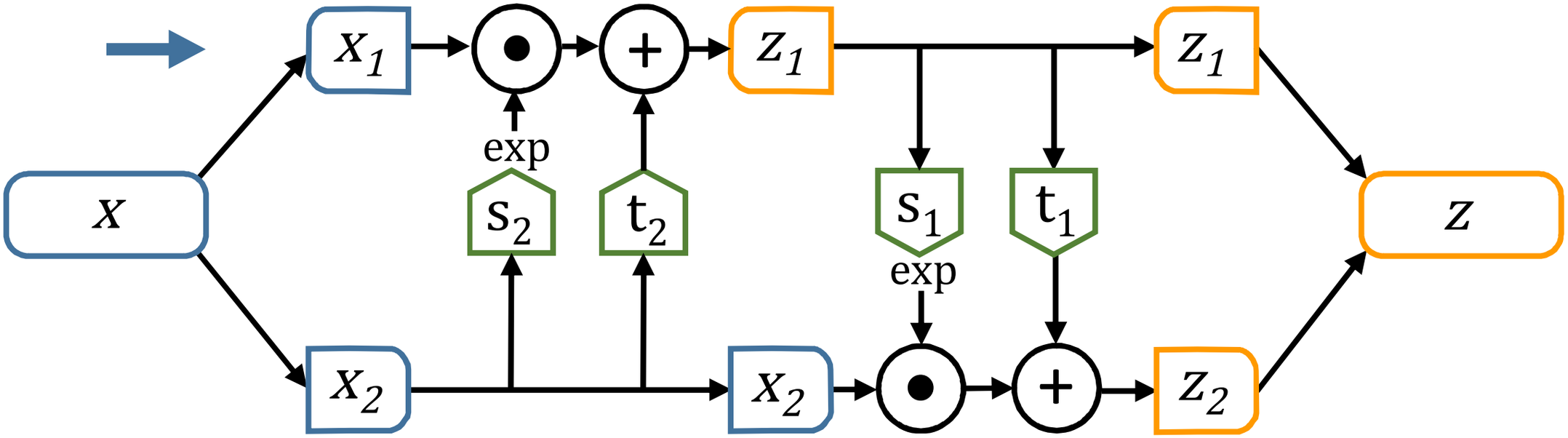
- Goal: assign probability density to each datapoint
- Learn bijective transformation between data and a latent space with tractable probability
- Build from simple invertible transformations with tractable Jacobian

$$p(\boldsymbol{x}) = p(\boldsymbol{f}^{-1}(\boldsymbol{x})) \prod_i \left| \det \left(\frac{\partial \boldsymbol{f}_i^{-1}}{\partial \boldsymbol{x}} \right) \right| =$$

$$p(\boldsymbol{u}) \prod_i \left| \det \left(\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{u}} \right) \right|^{-1}$$

Thanks to T. Loesche

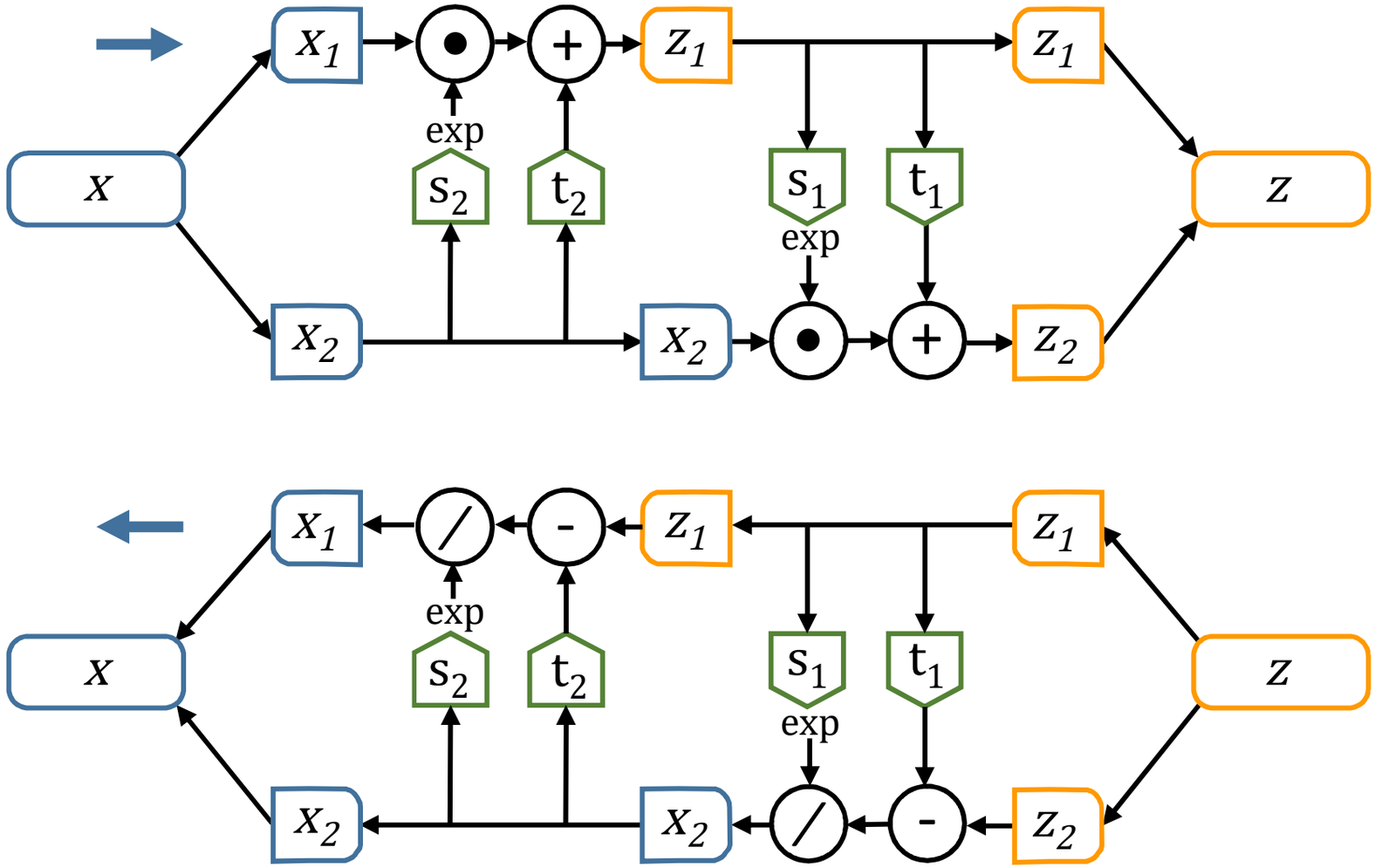
Coupling Flows



- Coupling flows / real NVP
  - Practically not the most widely used flow, but useful for illustration/understanding
  - Will use an alternative (masked autoregressive flows) for exercise

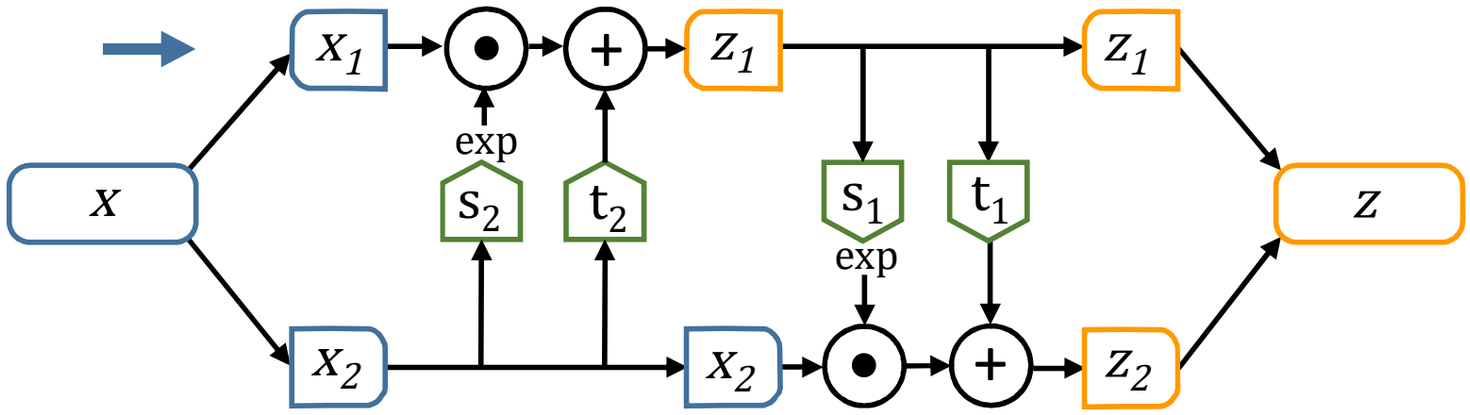
- Forward direction
- *s* and *t* are standard (e.g. fully connected) neural networks

Coupling Flows



- Forward and backward direction
- Can already see invertability
- What about Jacobian determinant?

Calculating Jacobian determinant



$$\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \xrightarrow{f_1} \begin{pmatrix}\mathbf{z}_1\\ \mathbf{x}_2\end{pmatrix} \xrightarrow{f_2} \begin{pmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{pmatrix}$$ with

$$\mathbf{x}_1 \xrightarrow{f_1} \mathbf{z}_1 = \mathbf{x}_1 \odot \exp(s_2(\mathbf{x}_2)) + t_2(\mathbf{x}_2)$$

$$\mathbf{x}_2 \xrightarrow{f_1} \mathbf{x}_2.$$

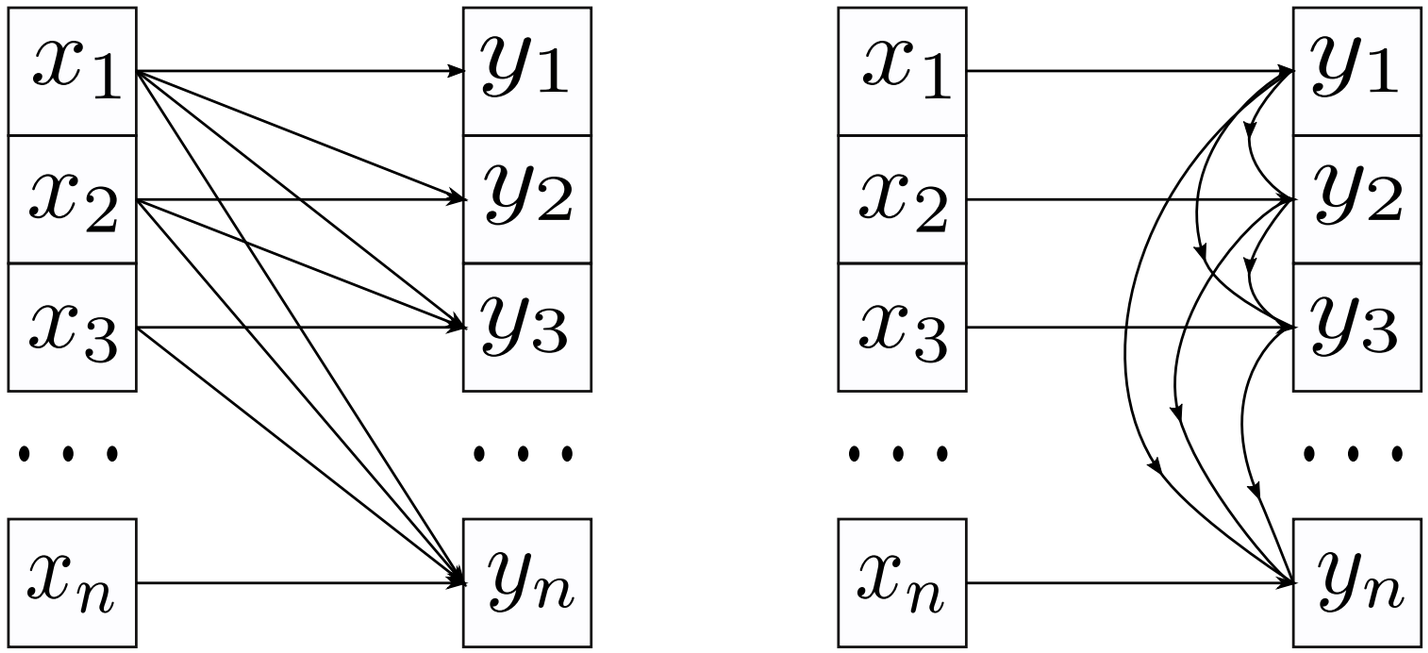
$$\mathbf{J_1} = \begin{pmatrix} \frac{\partial \mathbf{z}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{x}_2} \\ \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_2} \end{pmatrix} = \begin{pmatrix} \mathrm{diag}(\exp(s_2(\mathbf{x}_2))) & \frac{\partial \mathbf{z}_1}{\partial \mathbf{x}_2} \\ 0 & \mathbb{1} \end{pmatrix}$$

Triangular matrix by construction

$$\det \mathbf{J_1} = \prod \exp(s_2(\mathbf{x}_2)) = \exp\left(\sum s_2(\mathbf{x}_2)\right)$$

Similarly simple for J_2. Composition of functions means multiplying their det J.

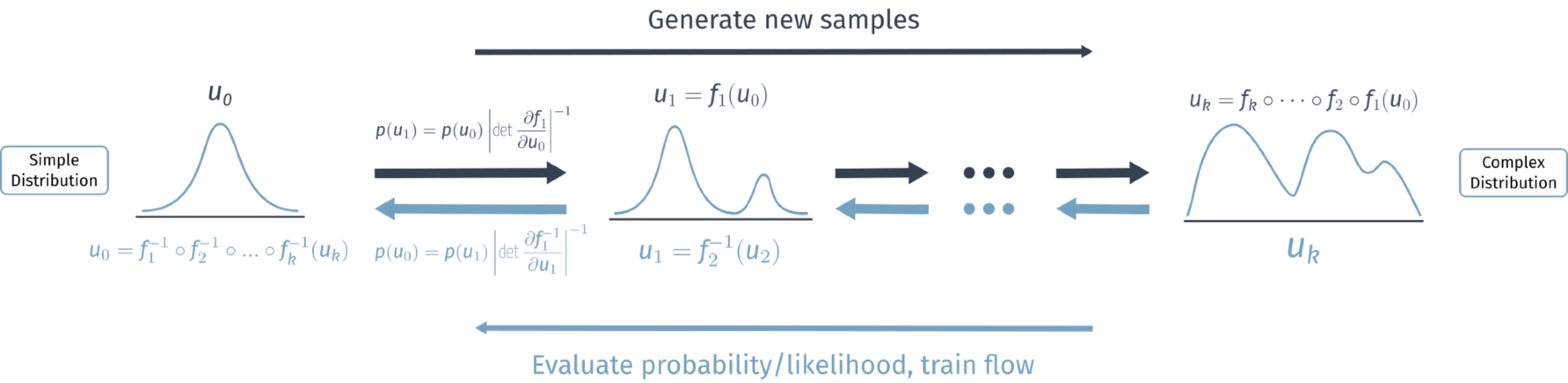
Autoregressive Flows



- Autoregressive property: Outputs conditioned on previous inputs
- Again, leads to simple Jacobian and invertible functions
- MAF: Masked Autoregressive Flow
  - Forward direction (data->latent) fast, backward slow
- IAF: Inverse Autoregressive Flow
  - Sampling direction (latent->data) fast
- **Many** other constructions exist as well (1908.09257 for an overview)

1908.09257

How to train NF?

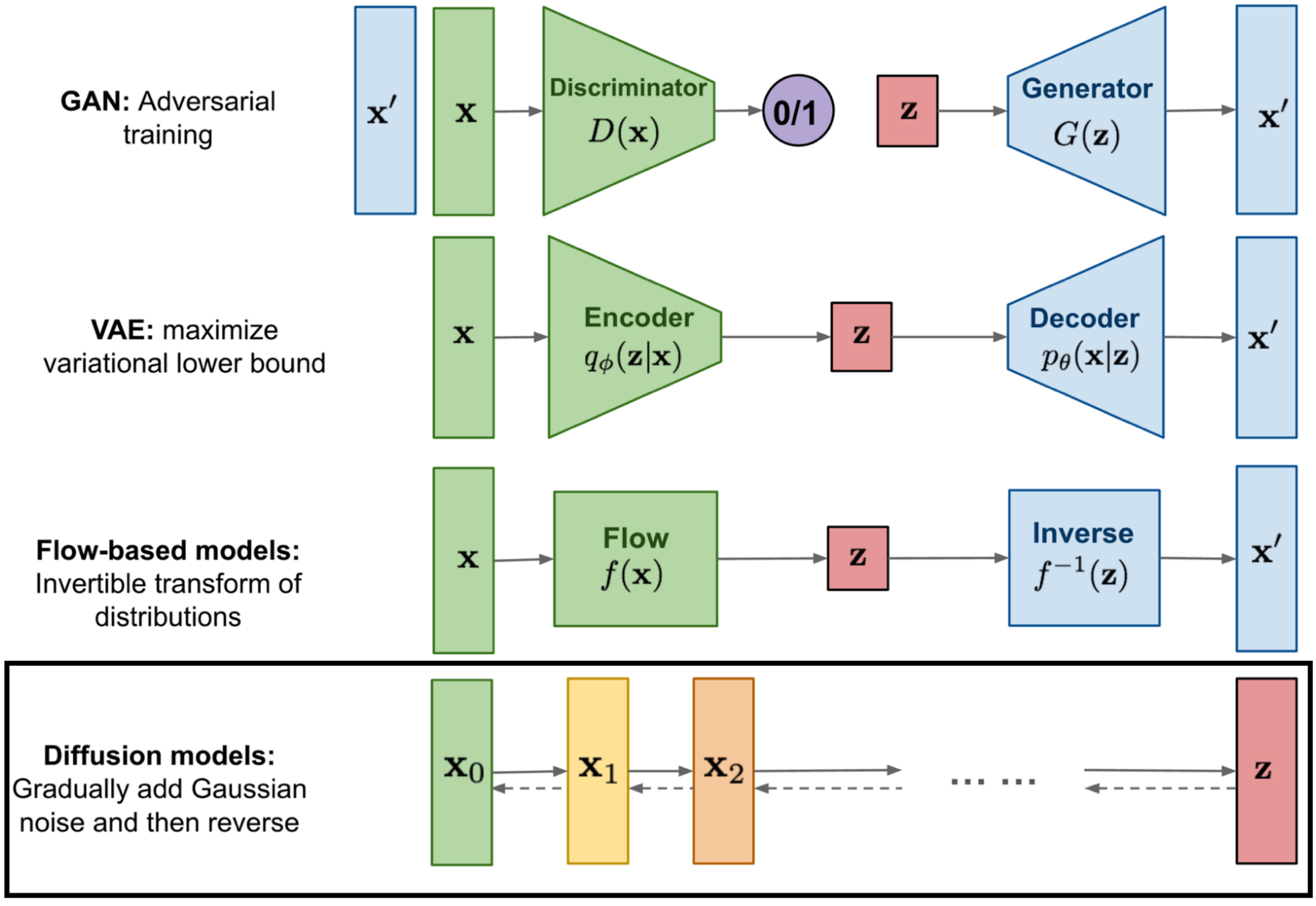


- Loss is the negative log likelihood, assume Gaussian latent space distribution
- Sample points from the training dataset
- Transform into latent space using flow (and keep track of det J)

$$\mathcal{L} = -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[-\frac{1}{2}||f(\mathbf{x}))||_2^2 + \sum s(\mathbf{x})\right]$$

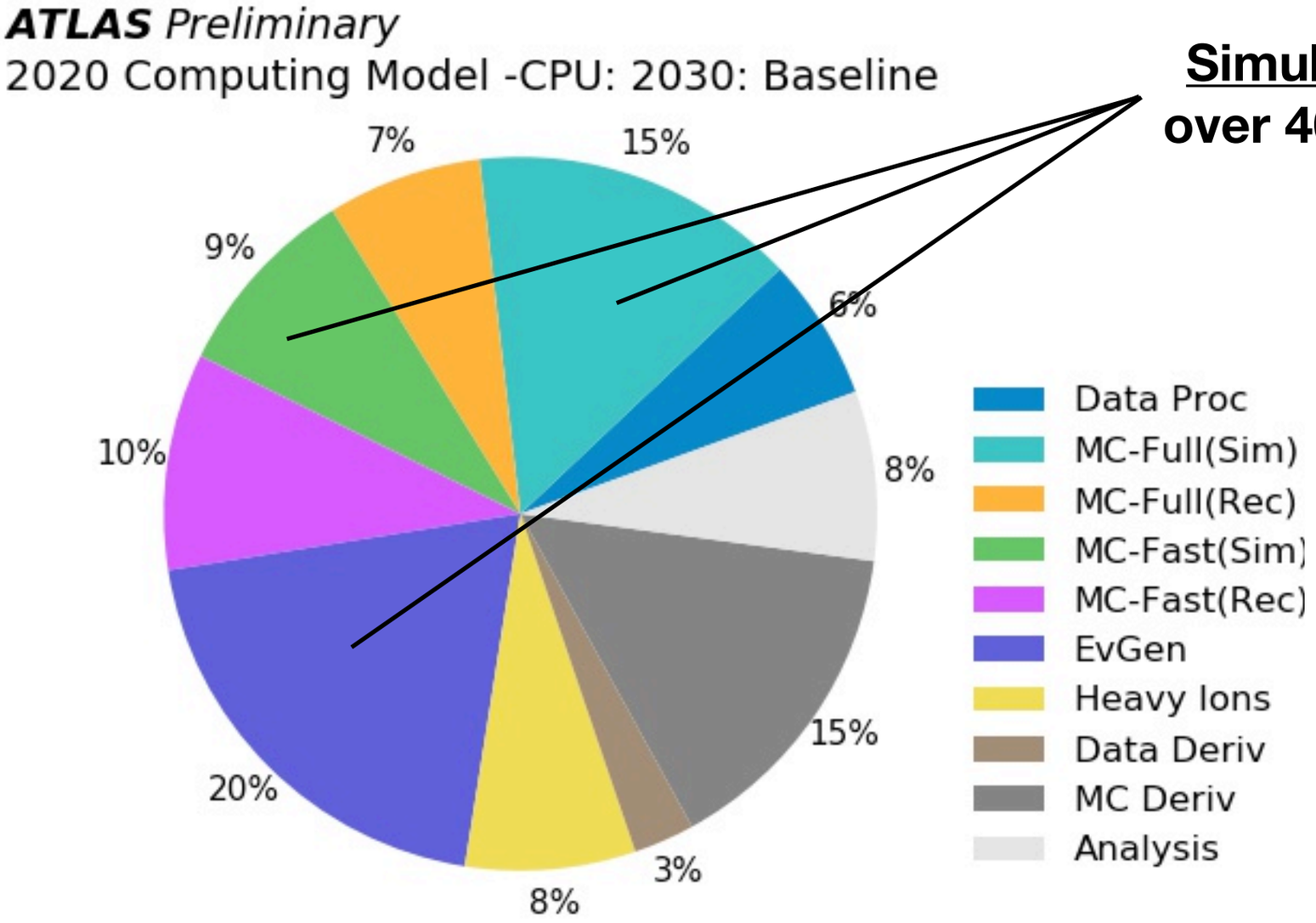
Thanks to T. Loesche

Generative models

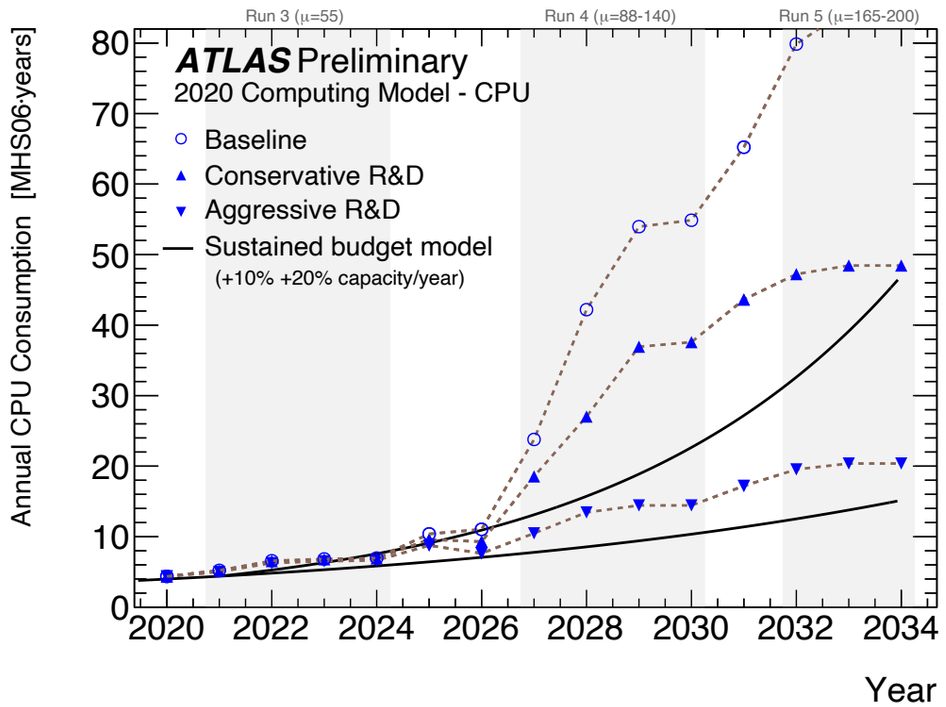


Application Example

Computing Bottleneck



Simulation and Generation steps over 40% of ATLAS compute effort

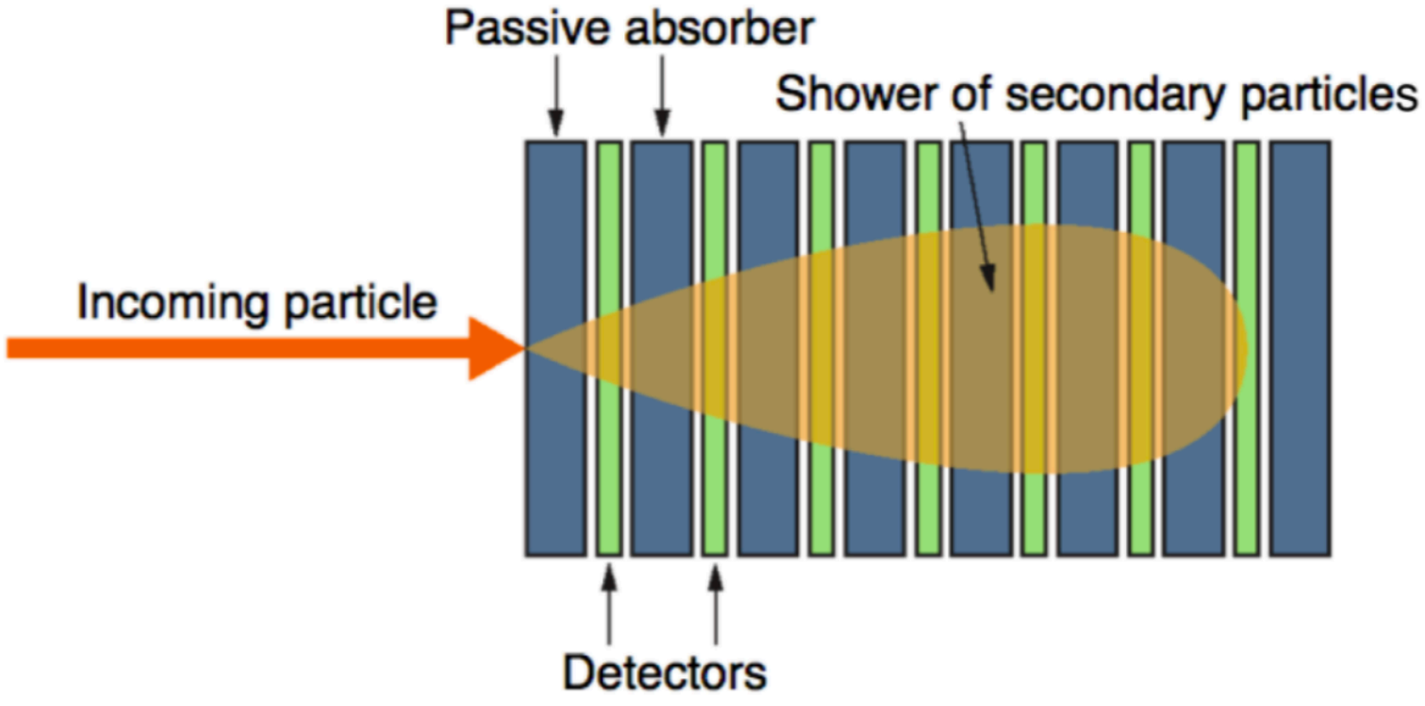


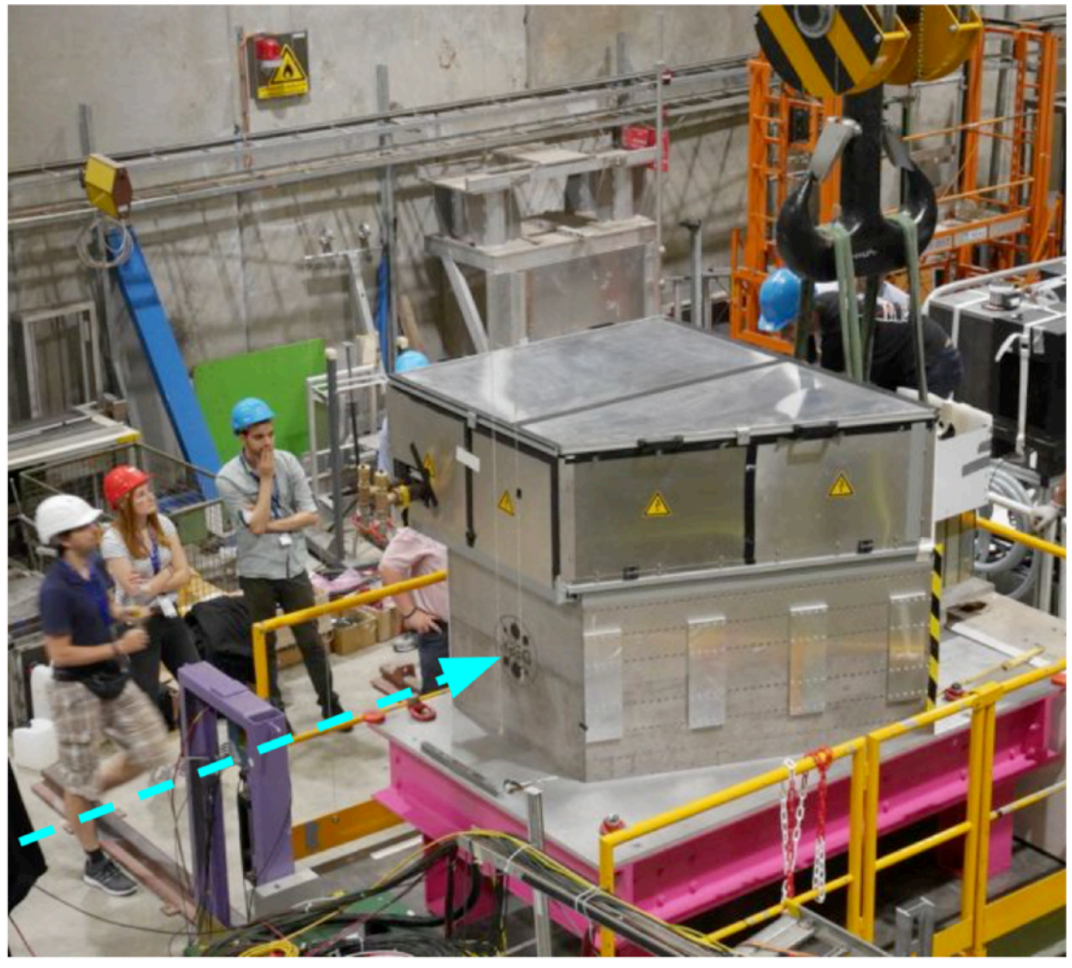
Compute needs measured in million-years of 2006 reference computer

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults
http://w3.hepix.org/benchmarking

Particle Showers

Main motivation:
Fast simulation of interaction between particles and material in complex calorimeter detectors

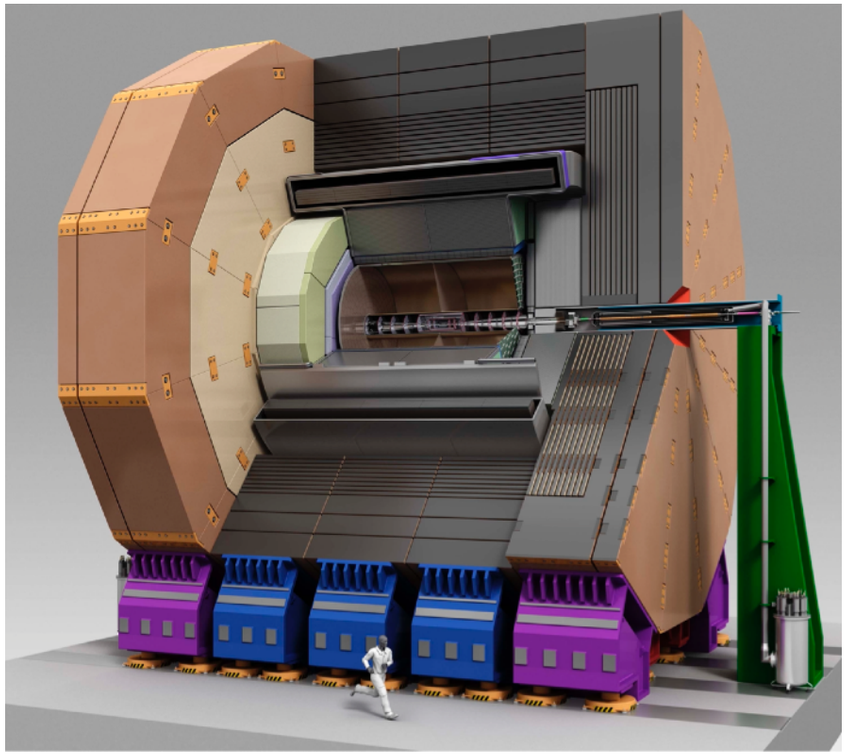


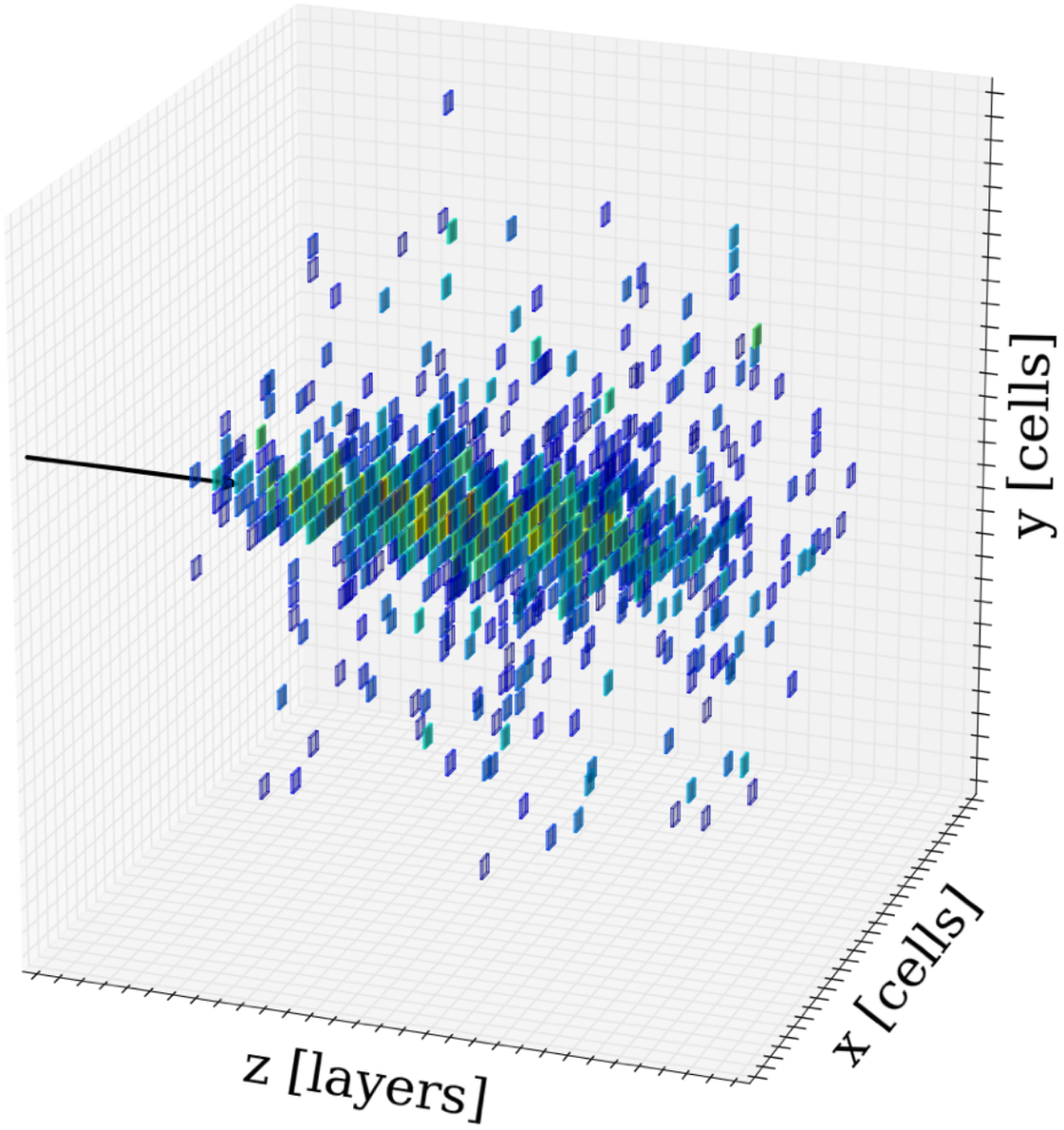


Concrete Problem

Describe photon showers in high granularity calorimeter prototype

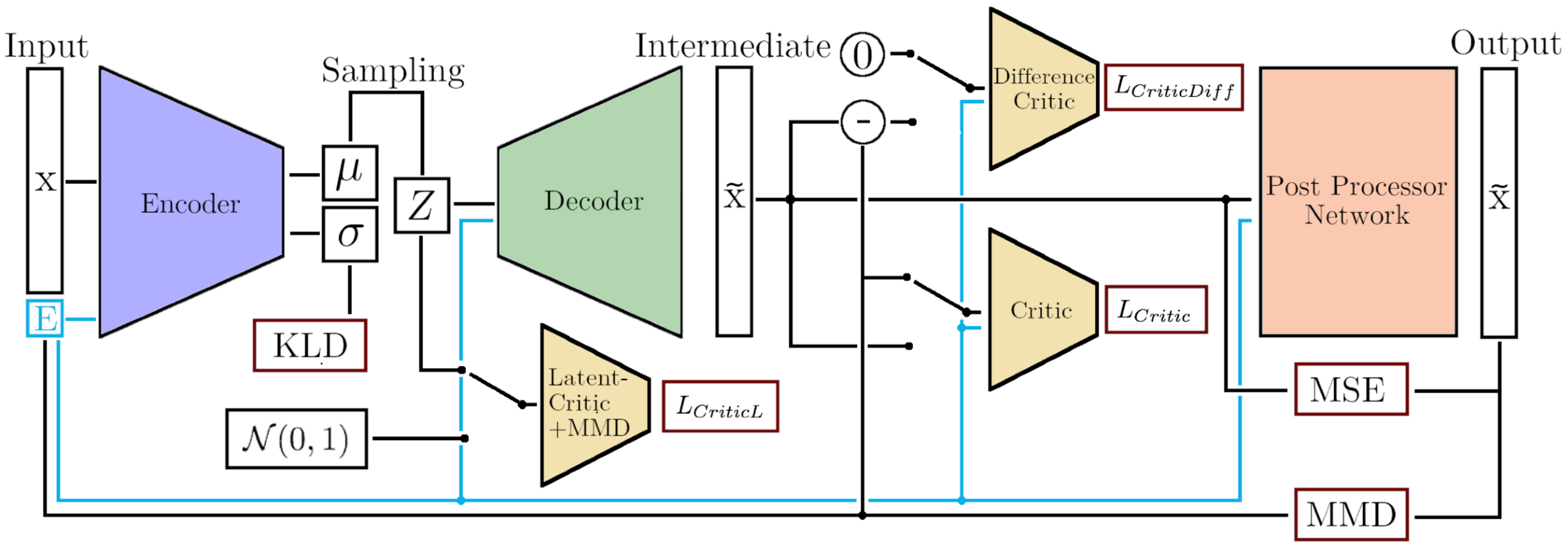
- 30x30x30 cells (Si-W)
- Photon energies from 10 to 100 GeV
- Use 950k examples (uniform in energy) created with GEANT4 to train





- Not only model individual images but also **differential distributions**

Architecture

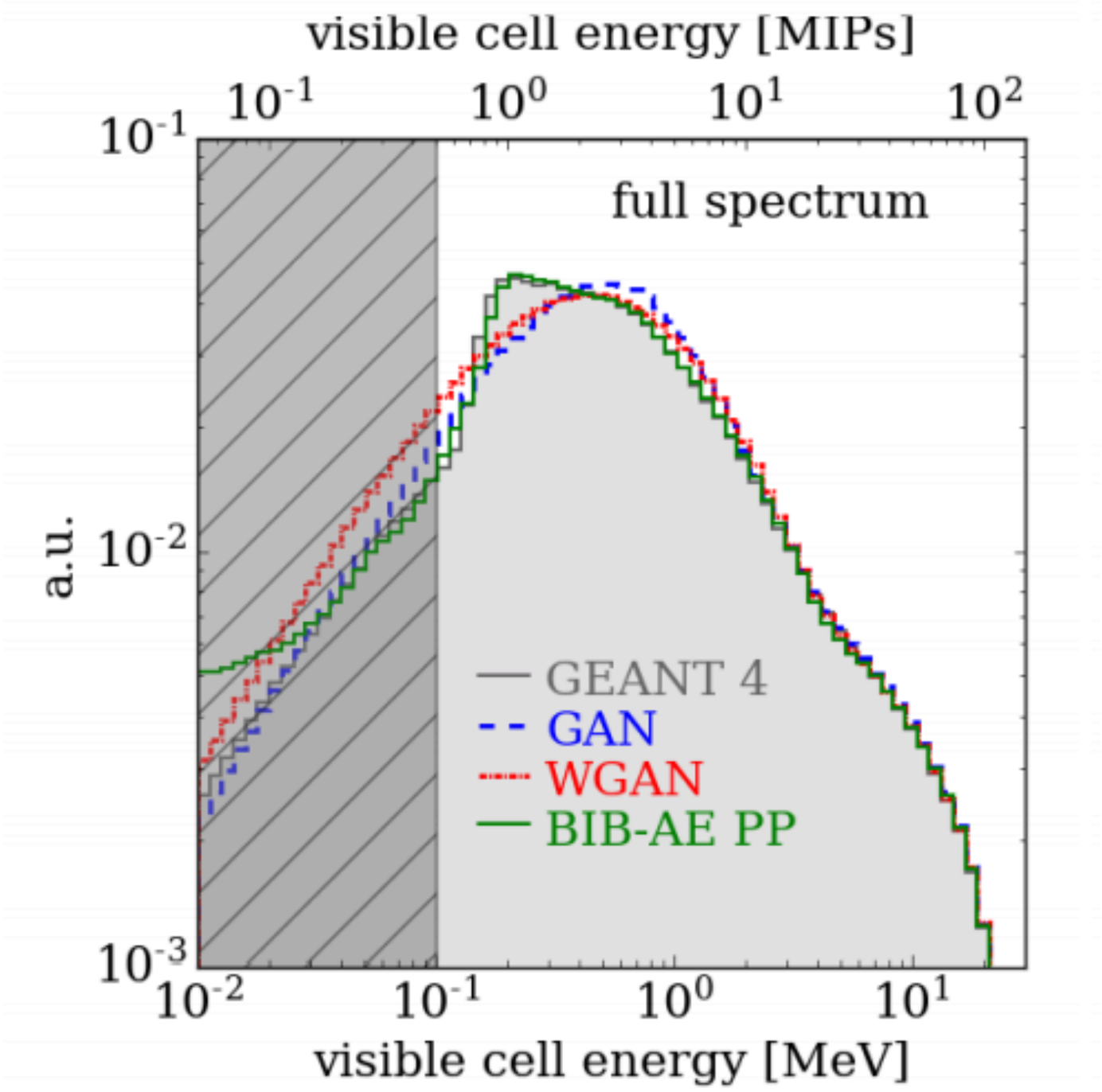


- BIB-AE (based on 1912.00830) with added post-processing
- Unifies features of GAN and VAE
- 71M trainable parameters

$$
\begin{aligned}
L_{\text{BIB-AE}} = & -\beta_{C_L} \cdot \mathbb{E}[C_L(E(x))] \\
& -\beta_{C} \cdot \mathbb{E}[C(D(E(x)))] \\
& -\beta_{C_D} \cdot \mathbb{E}[C_D(D(E(x)) - x)] \\
& +\beta_{\text{KLD}} \cdot \text{KLD}(E(x)) \\
& +\beta_{\text{MMD}} \cdot \text{MMD}(E(x), \mathcal{N}(0,1))).
\end{aligned}
$$

2005.05334

Results


visible cell energy [MIPs]
10^{-1}
10^{0}
10^{1}
10^{2}
full spectrum
a.u.
10^{-2}
10^{-3}
GEANT 4
GAN
WGAN
BIB-AE PP
10^{-2}
10^{-1}
10^{0}
10^{1}
visible cell energy [MeV]

Additional Challenges

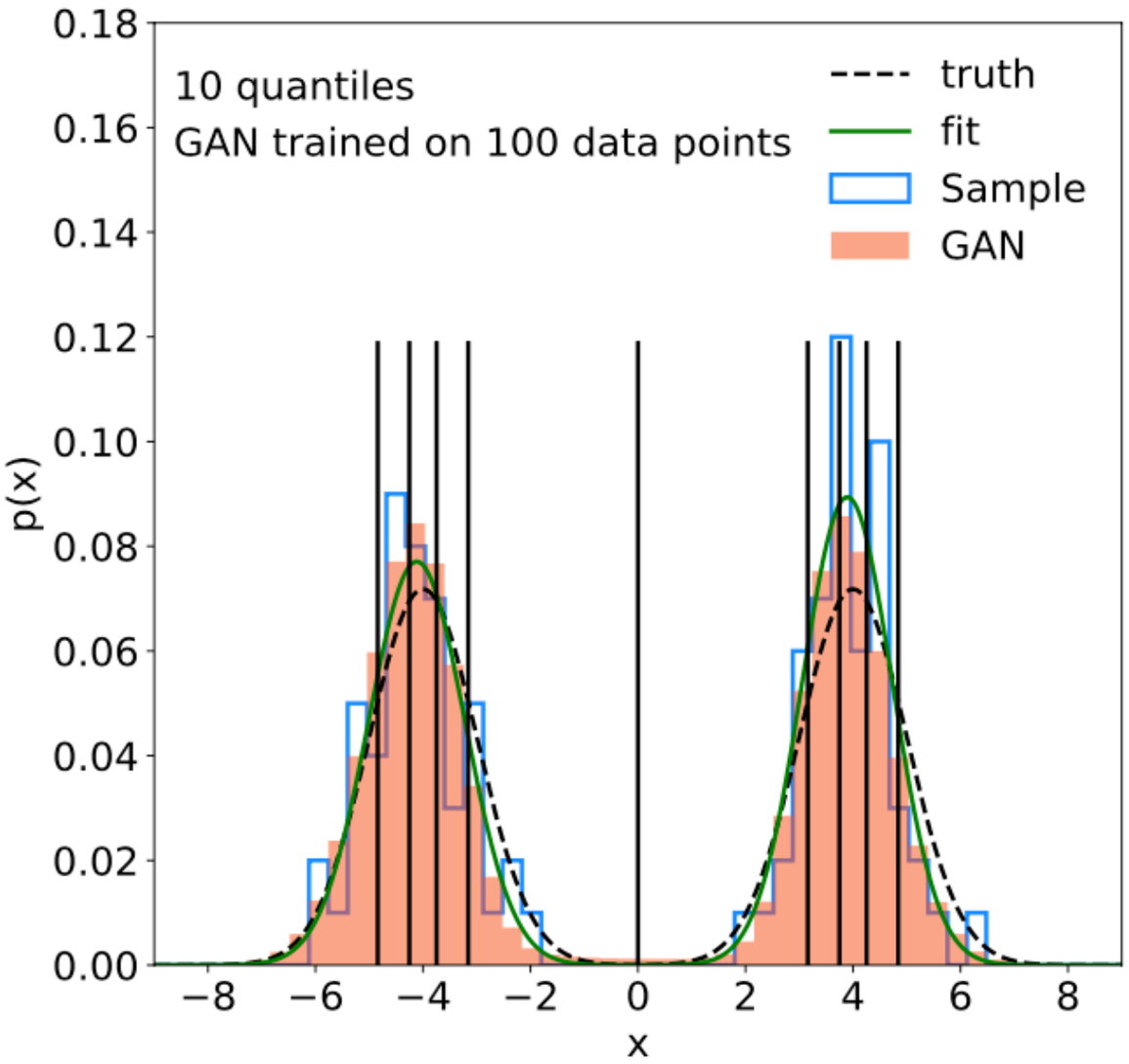
- How to evaluate convergence of models?
  - e.g. classifier metric
  - e.g. Frechet-inception distance
- Correctly model many differential distributions
- Condition on a large number of quantities (energy, particle type, impact position, angle, …)

Limitations of Generative Models

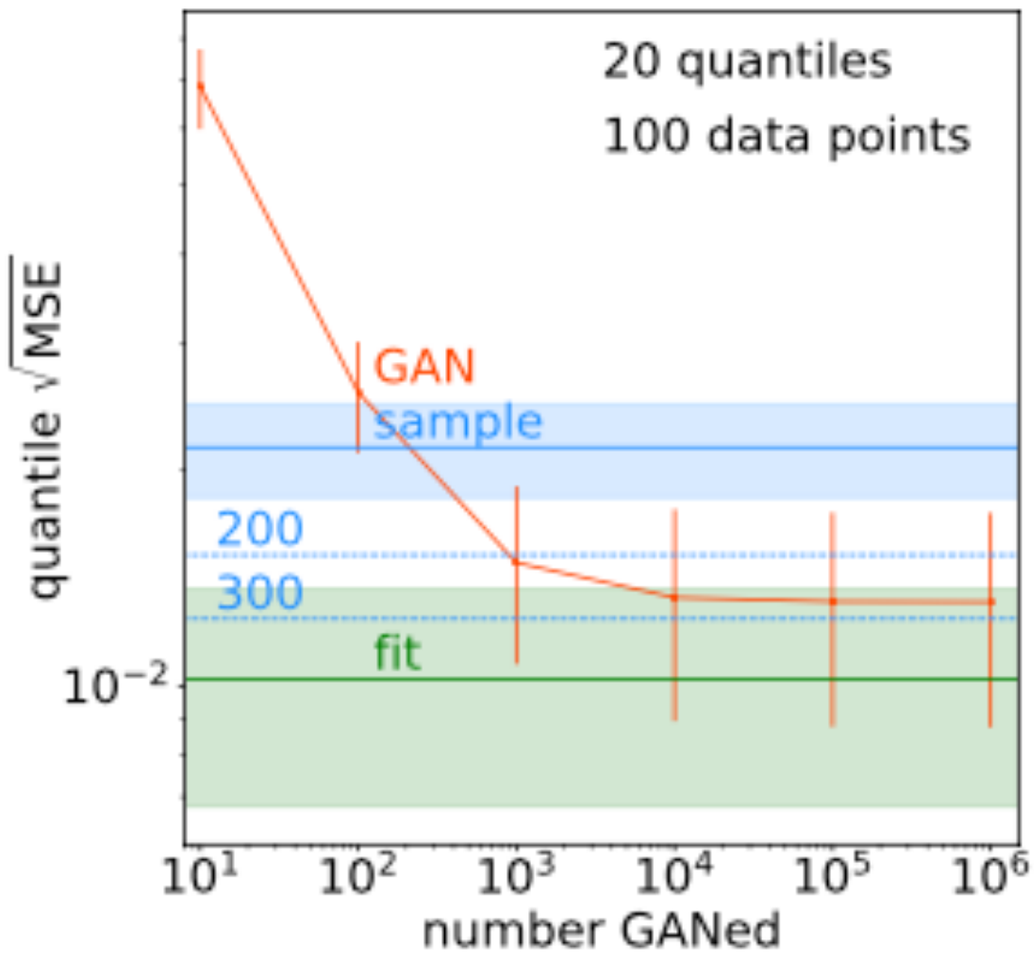
- Generative models are powerful in quickly producing more examples, still need training examples
- Machine learning is great at interpolation, but it cannot do magic
    - Be careful if training data covers desired application
- Expect to simulate typical examples, do not trust the tails of distributions without verification
    - Big question of uncertainty of generative models
- Can networks **amplify?**

Statistics

If we train a generator on N data points, and use it to produce M>>N examples, what is the statistical power of the M points?



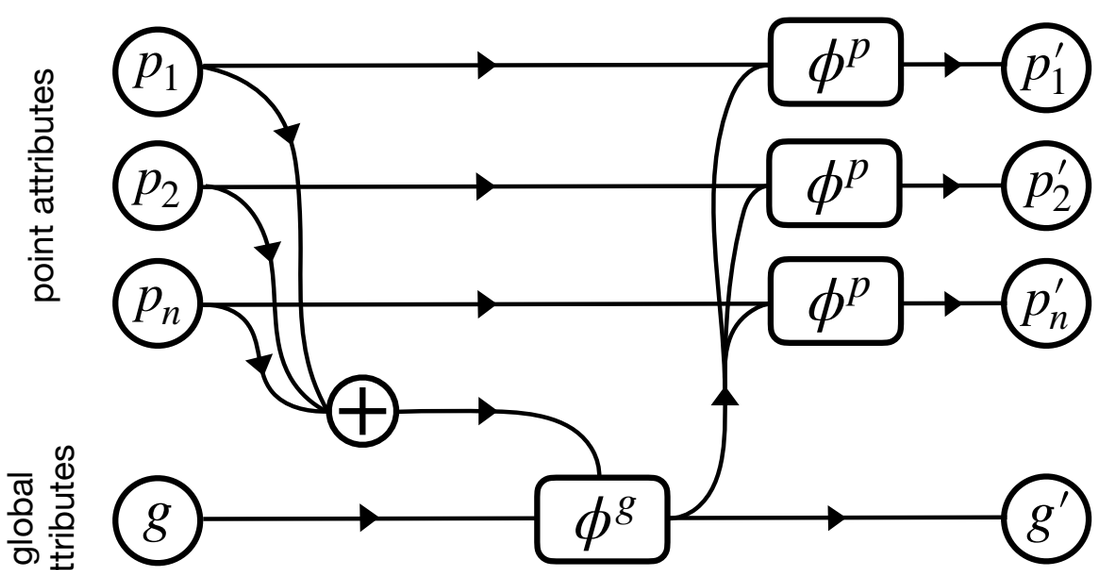
Quantile MSE: $\frac{1}{K_{\text{quant}}} \sum_{j=1}^{K_{\text{quant}}} \left(x_j - \frac{1}{K_{\text{quant}}} \right)^2$

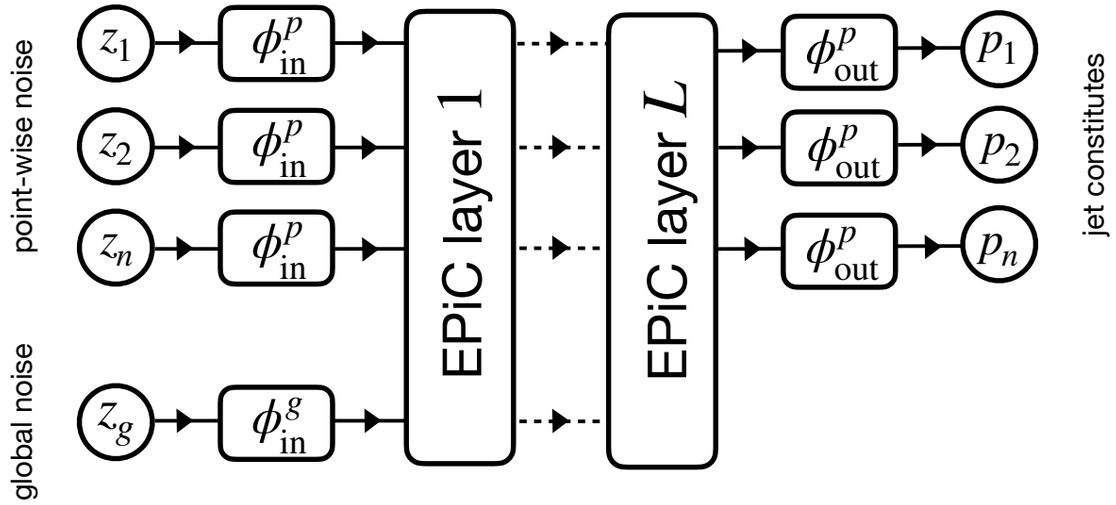


Butter, Diefenbacher, Kasieczka, Nachman, Plehn *GANplifying Event Samples, 2008.06545, also 2202.07352*

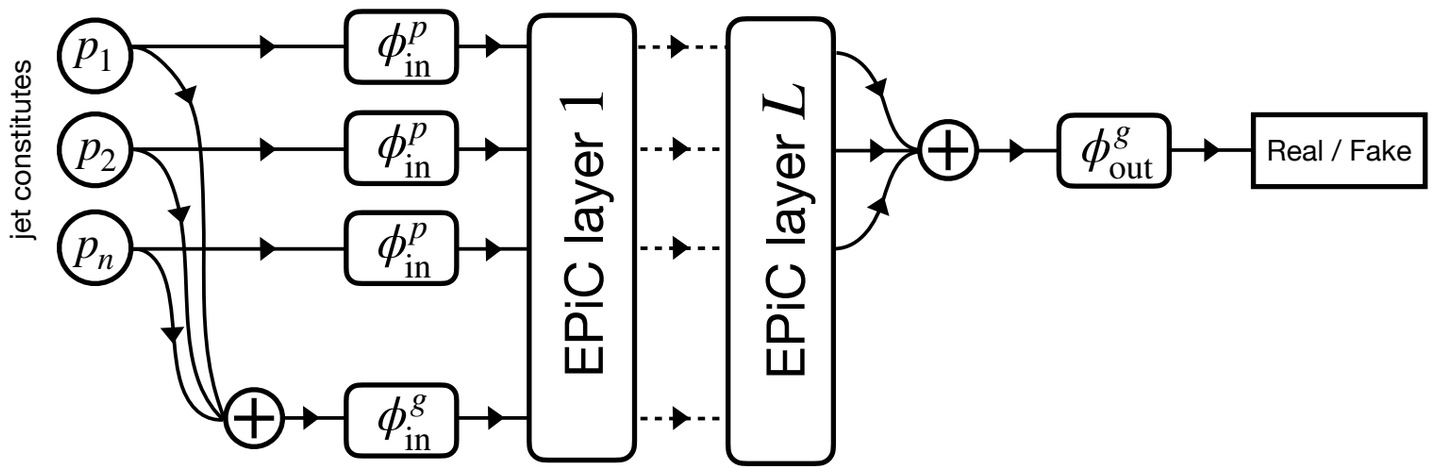
Generative Symmetries?

- Most generative work focuses on fixed structures
- Can we combine this with the topic of Monday morning?





(a) Generator



(b) Discriminator

2301.08128 EPiC - Buhmann et al; See also 2106.11535

Part IV: Anomaly Detection

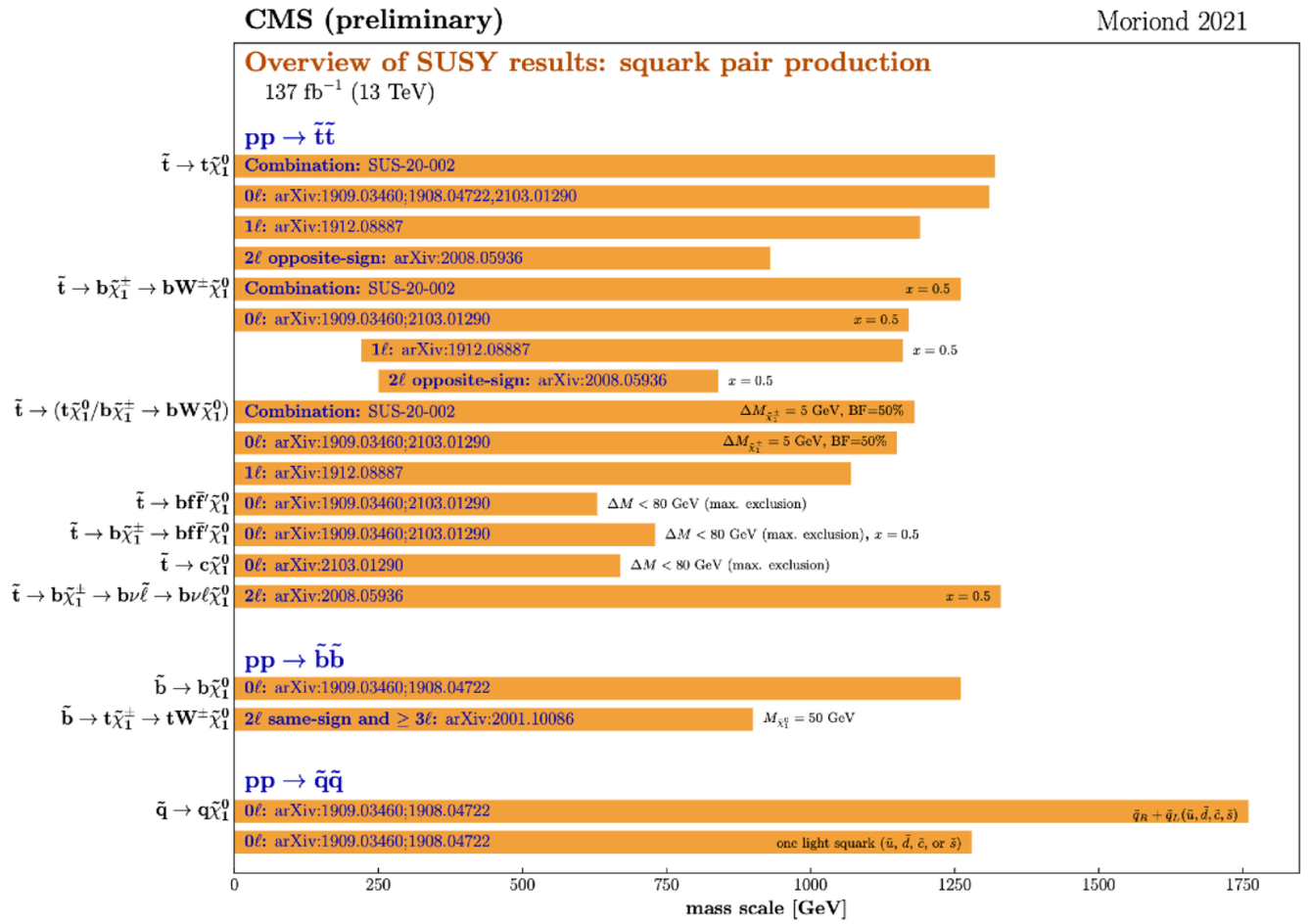
Anomaly Detection

Reminder

- Theoretical and experimental reasons to expect new physics beyond the Standard Model

- However, so far only negative results in direct (model driven) searches

- Make sure that we do not miss potential discoveries at the LHC
→**Anomaly detection**



What is an anomaly?







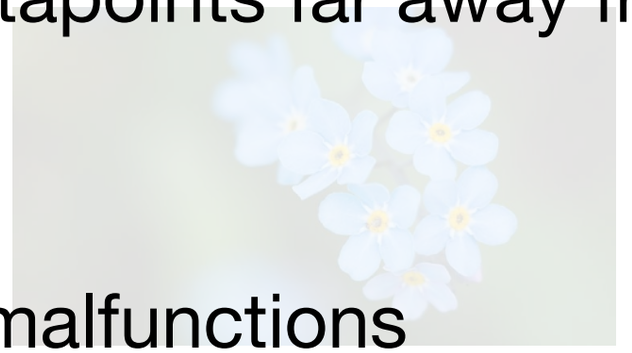






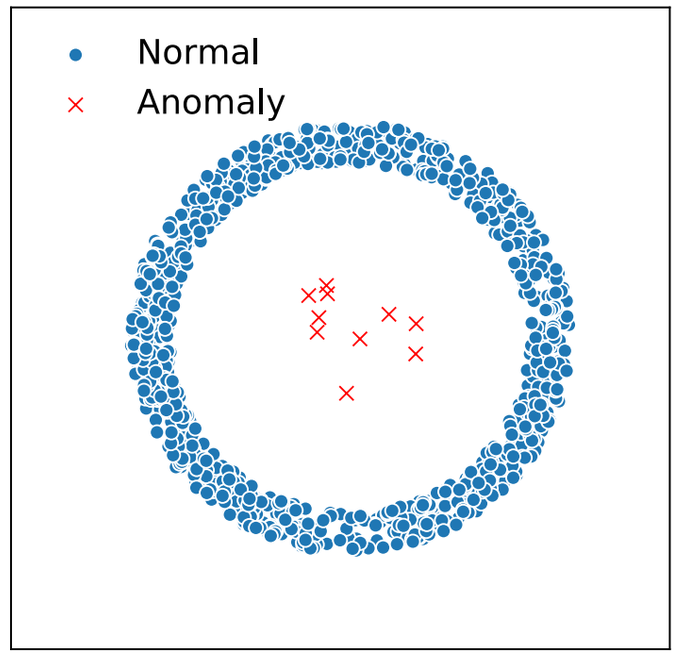
Point anomaly

- **Outliers**: Datapoints far away from regular distribution
- Examples:
  - Detector malfunctions
  - Background-free search







And now?





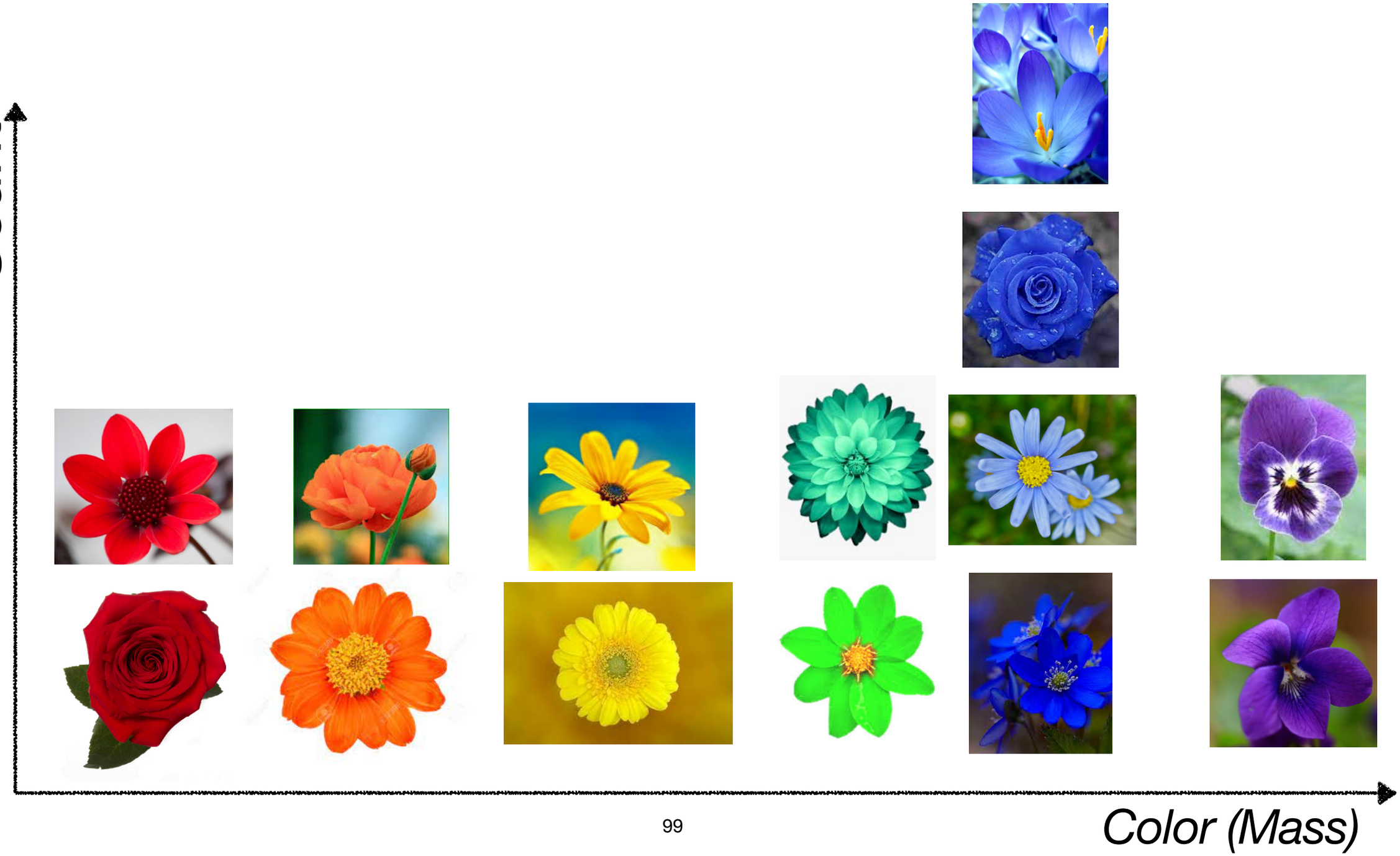






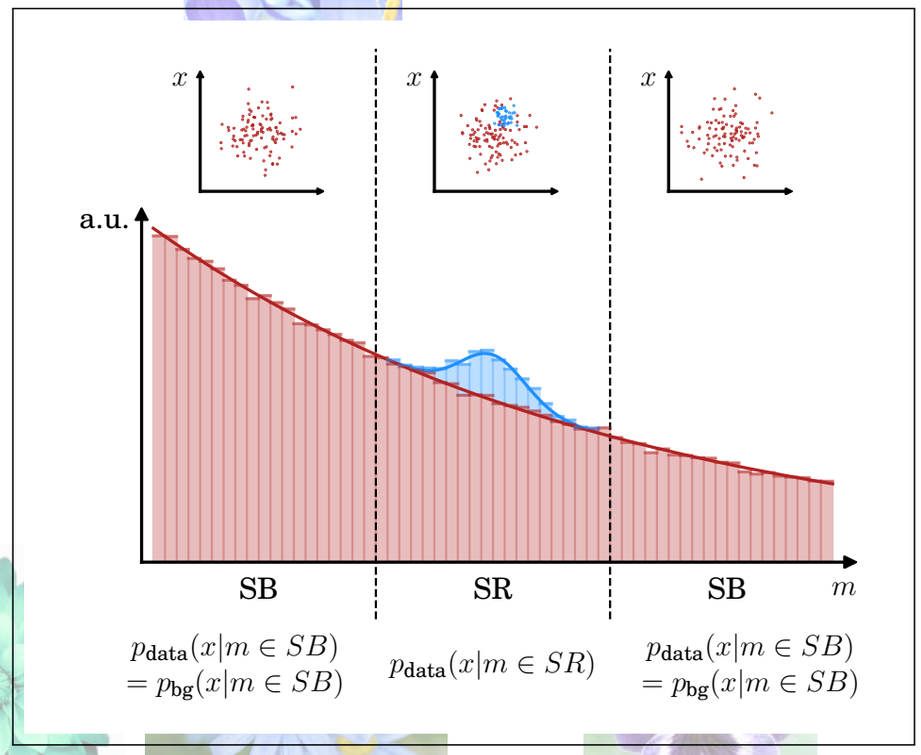


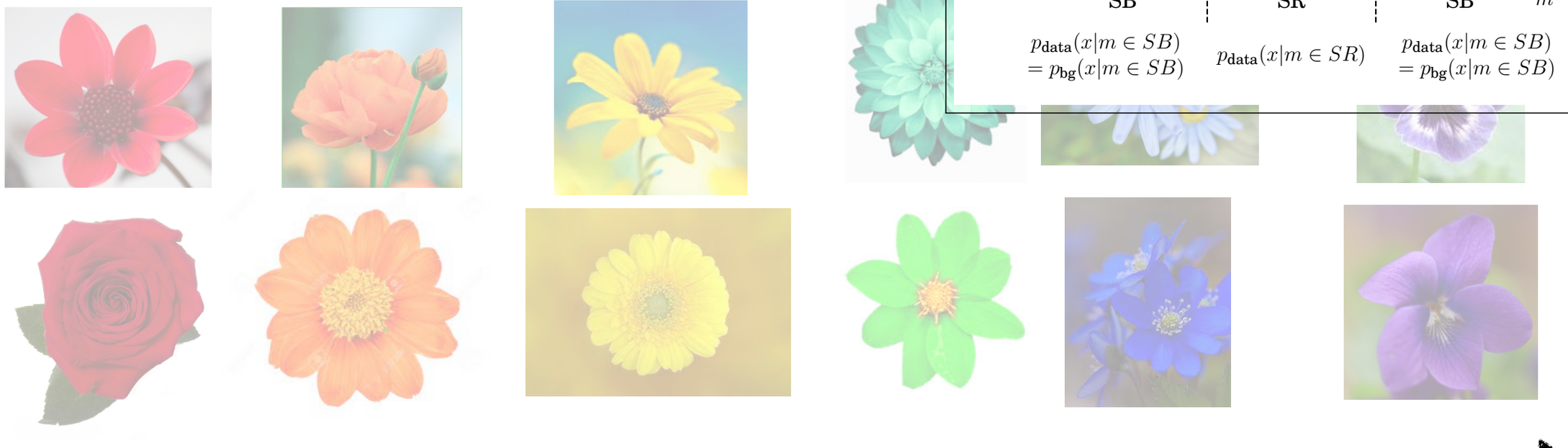

Group anomaly



Group anomaly

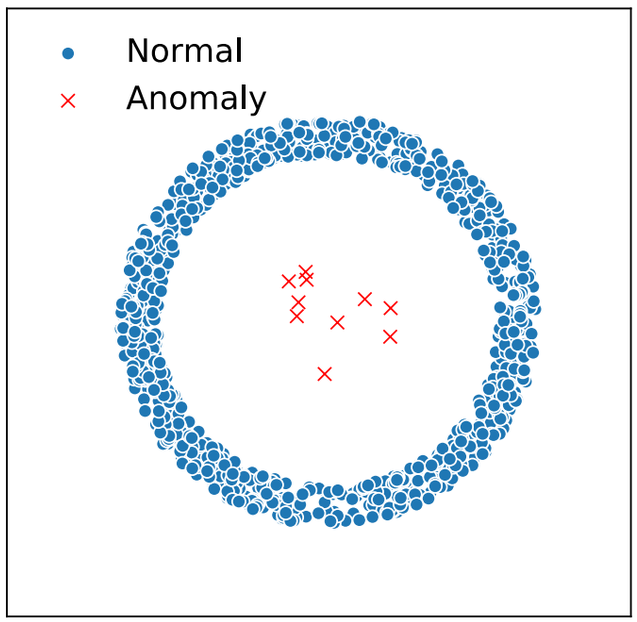
- Individual examples not anomalous, but interesting collective behaviour
- Examples:
  - New physics searches, e.g. resonances
  - Excess in time series



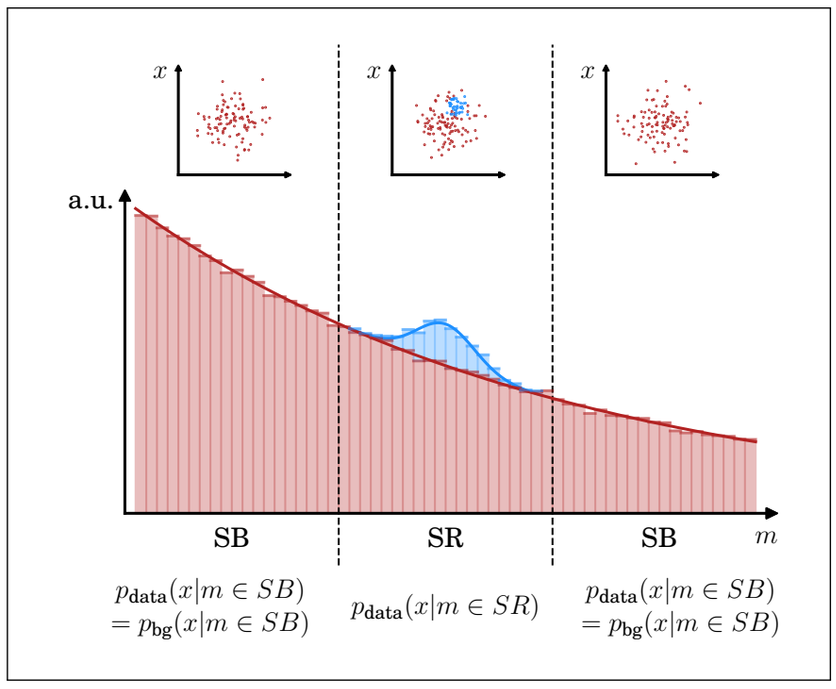


Types of anomalies

- **Outliers/Point anomalies**: Datapoints far away from regular distribution
- Examples:
  - Detector malfunctions
  - Background-free search



- **Group anomlies**: Individual examples not interesting,
  but signal is an overdensity with respect to background
- Examples:
  - Resonance searches
  - Transient signals in time series



Anomaly Searches

- Orthogonal strategy to model specific searches:
  - Discover new physics with making minimal assumptions
- Less sensitive to one specific model, broader coverage

ML-assisted global comparison

- Systematically compare simulation to recorded data, look for differences
- Con: Rely on imperfect simulation, maximally background model dependent
- Pro: Sensitive to all types of anomalies

Resonant anomaly detection / Enhanced bump hunts

- Estimate background in-situ from data
- Con: Need to make assumptions about signal shape
- Pro: Data-driven on background model

GK et al 2101.08320; Arrested et al 2105.14027; CMS 2010.02984; ATLAS 1807.07447

Anomaly Searches

- Orthogonal strategy to model specific searches:
  - Discover new physics with making minimal assumptions
- Less sensitive to one specific model, broader coverage

ML-assisted global comparison

- Systematically compare simulation to recorded data, look for differences
- Con: Rely on imperfect simulation, maximally background model dependent
- Pro: Sensitive to all types of anomalies

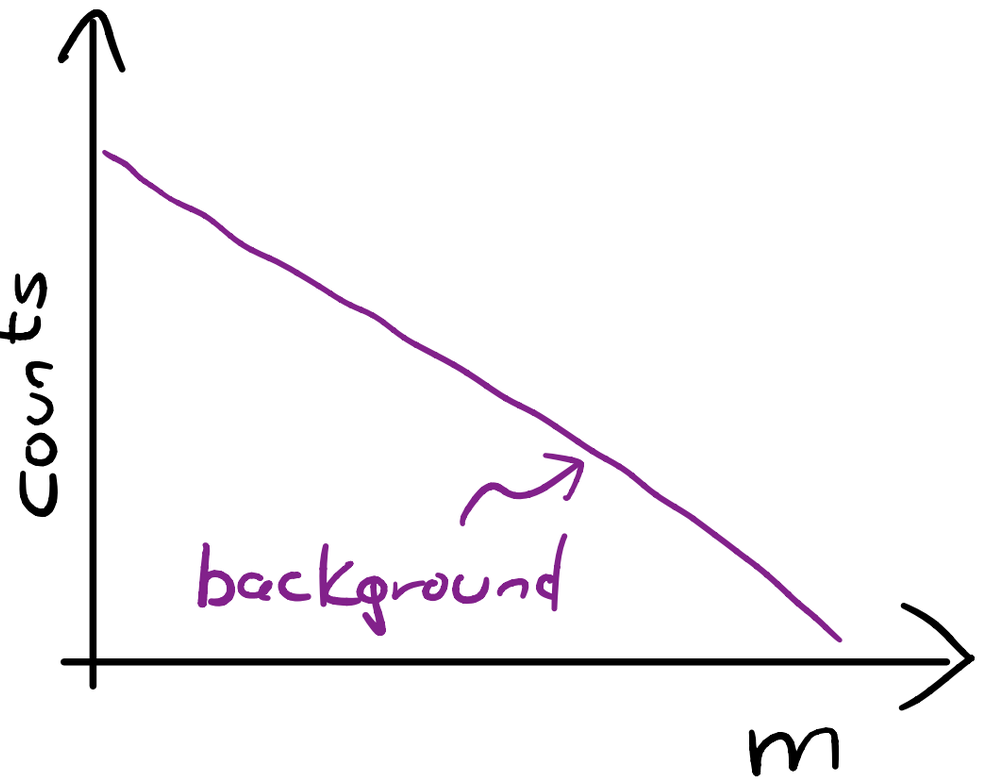
Resonant anomaly detection / Enhanced bump hunts

- Estimate background in-situ from data
- Con: Need to make assumptions about signal shape
- Pro: Data-driven on background model

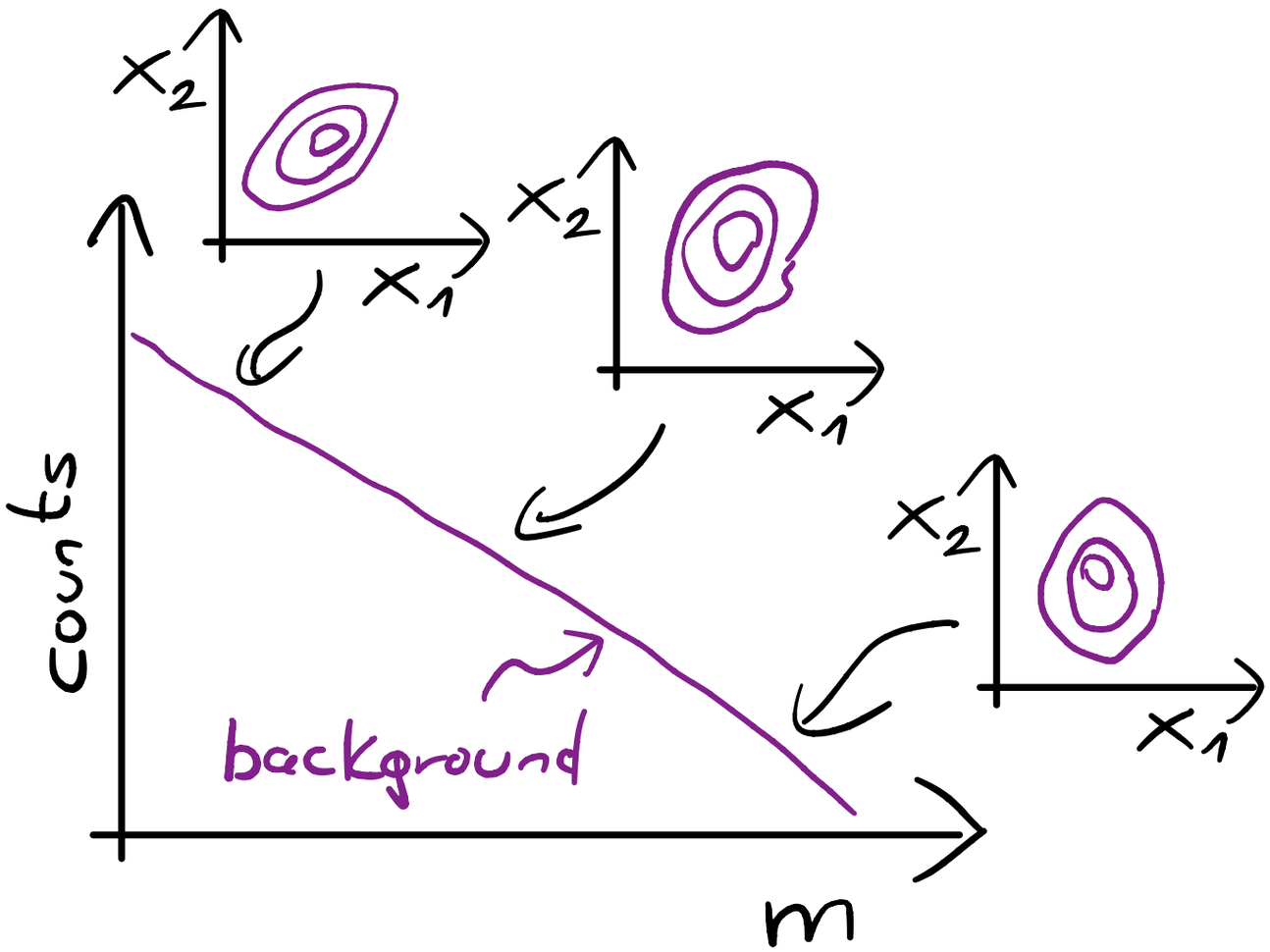
- Will focus on these for rest of the lecture

GK et al 2101.08320; Arrested et al 2105.14027; CMS 2010.02984; ATLAS 1807.07447

Resonant Anomaly Detection

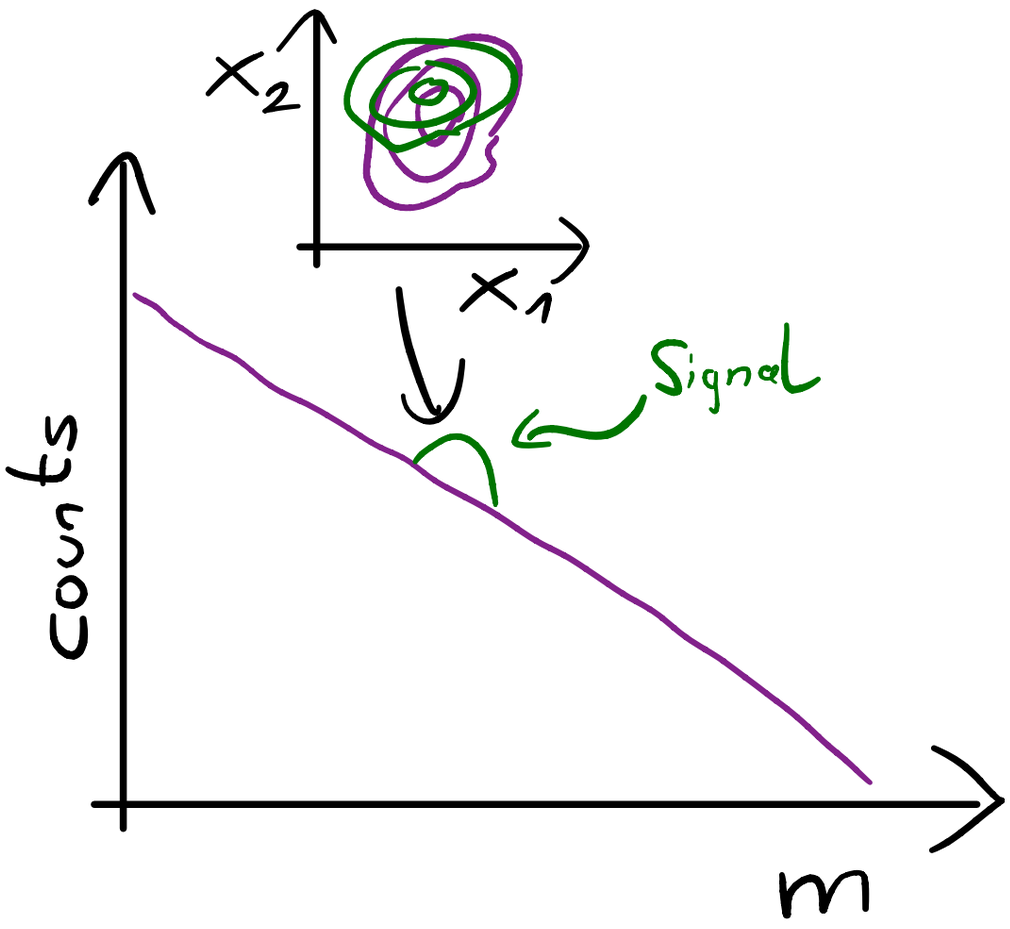


Resonant Anomaly Detection



+ additional dimensions
(in general correlated with m)

Resonant Anomaly Detection



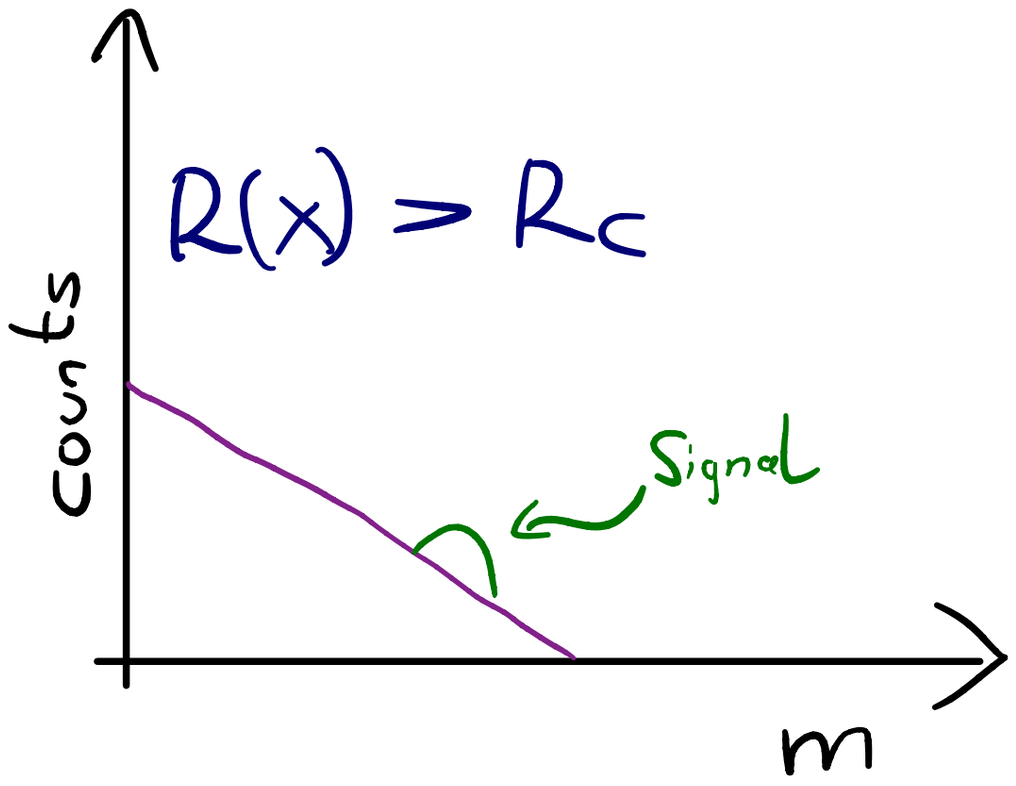
Look for a small signal, localised in m, and different shape in other features

Need to find a feature in which signal is resonant and background smooth.

No assumptions in other features.

Further generalisation as open issue.

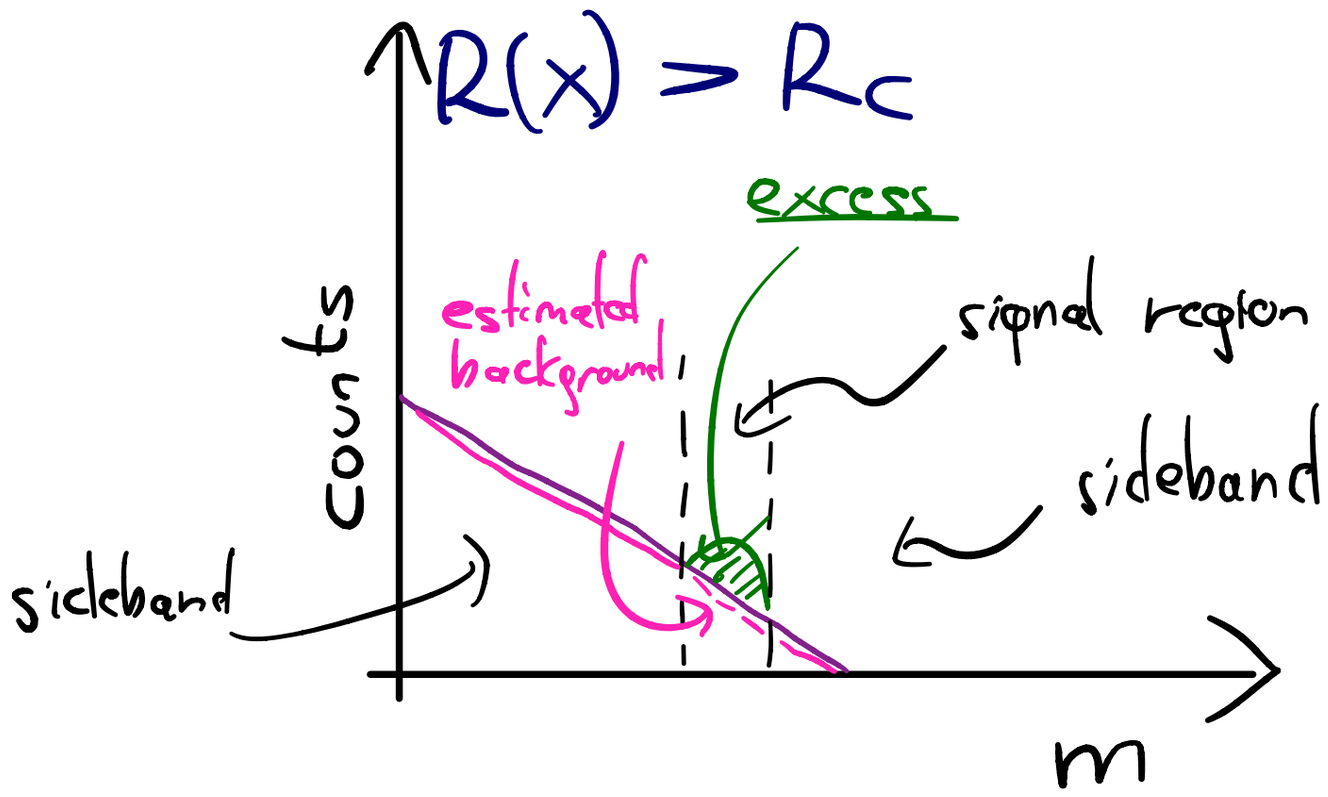
Resonant Anomaly Detection



Enhanced bump-hunt: Use ML to build classifier $R(x)$ so that selecting $R(x) > c$ enhances signal fraction

No worry, will come back to HOW this is done is a moment

Resonant Anomaly Detection



Enhanced bump-hunt: Then fit background from sidebands, compare to data in signal region

Overall Strategy

- Define an anomaly score **a**
  - Should be high for anomalous (signal-like) and low for background-like data

- Use a selection **a** to create an anomaly-enriched dataset

- Estimate background from data

- Compare predicted background to number of observed with high **a** (+potentially other selection criteria)

  - Statistically safe: Compare null-hypothesis (data is background only) to alternate hypothesis (data is not backround)
  - If no observations: Define exclusion limits by injecting potential signals

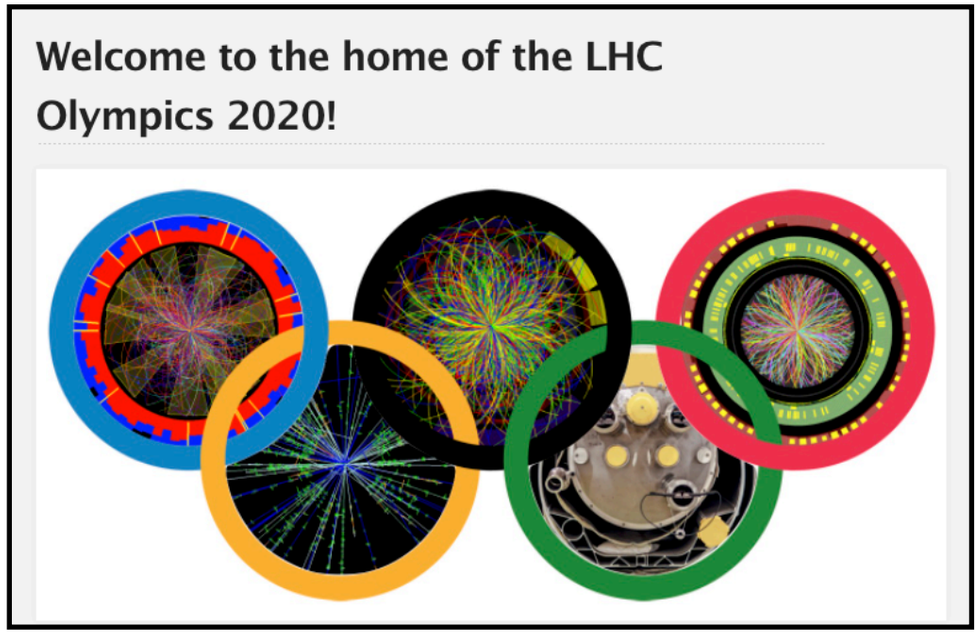
Introducing: LHC Olympics

- Encourage development and comparison of model-agnostic search strategies
    - Focus on group anomalies, data-driven searches
    - Use for a convenient overview of space of techniques
    - Complementary to 2105.14027

- Provide a complete package, balance details vs accessiblity

- Datasets:
    - One R&D dataset for algorithm development
    - Three black box datasets (BB1-BB3)
        - Unblinded over time

- Timeline:
    - Spring 2019: Release R&D dataset (link)
    - Autumn 2019: Release BB datasets (link)
    - January 2020: Winter Olympics as part of ML4Jets, unblinding of BB1 (link)
    - July 2020: (Virtual) Summer Olympics, unblinding of BB2 and BB3 (link)
    - LHC Olympics paper (https://arxiv.org/abs/2101.08320) public

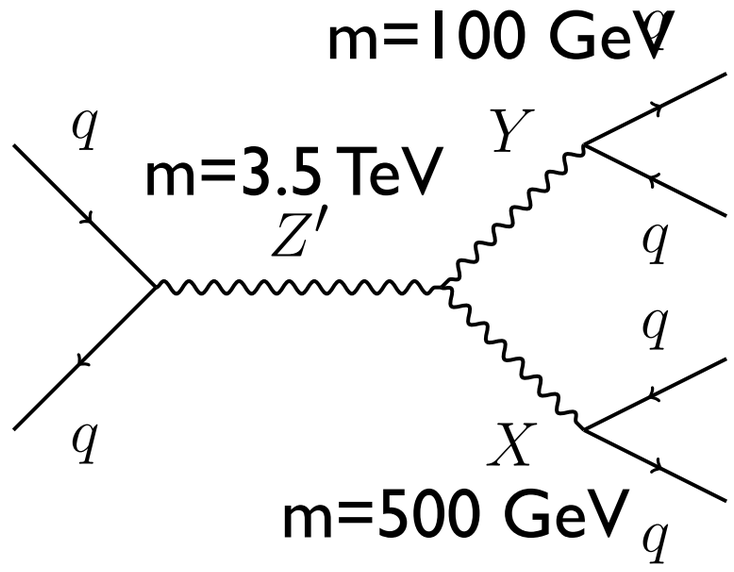


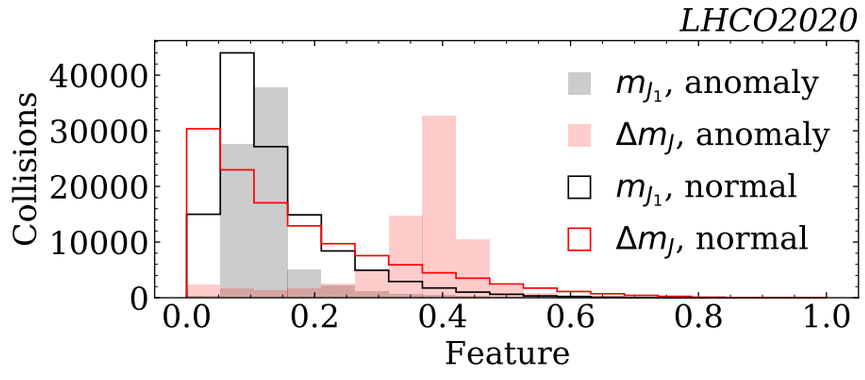
https://lhco2020.github.io/homepage/

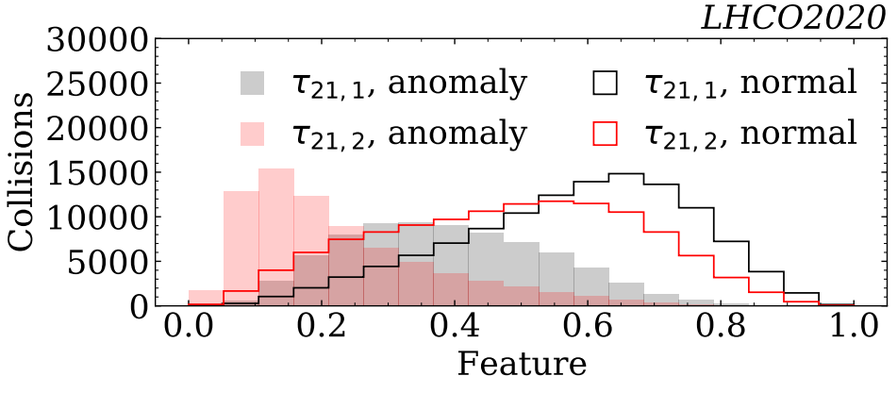
R&D dataset

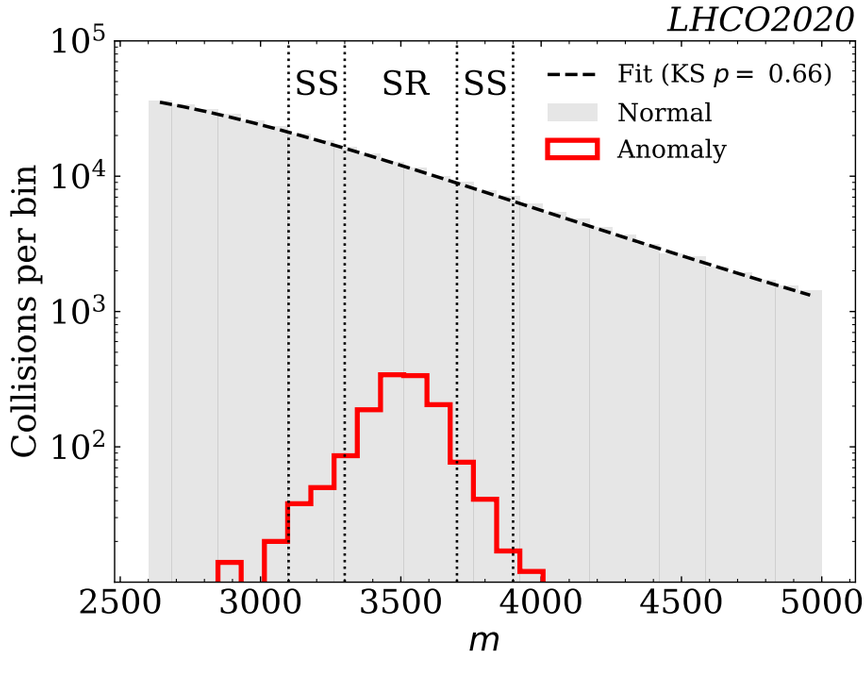
- For building and testing methods
- 1M background examples (Standard Model), 100k signal examples (signal, see Feynman diagram on the right)
- Labels provided

- Relatively simple signal
  - Known to differ in previously mentioned features from background distribution
- Unrealistically high S/B









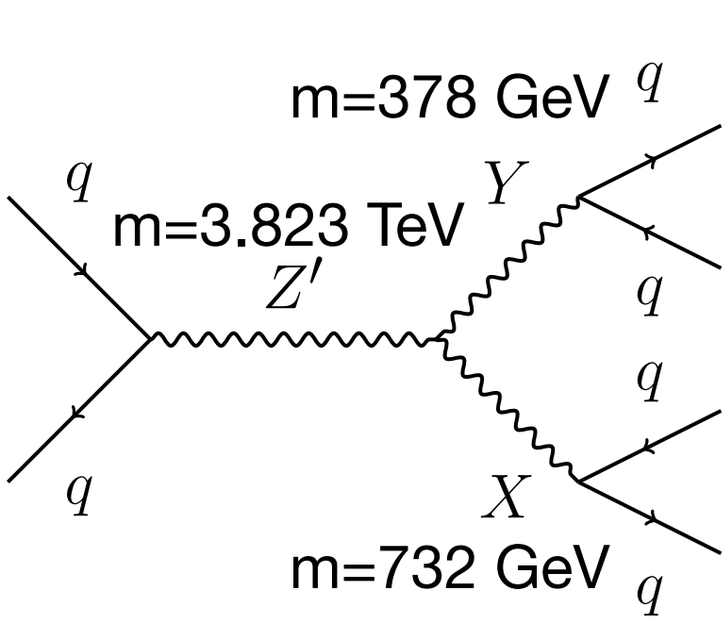
2107.02821

Challenge datasets

- All contain total of 1M examples; might contain signal; no labels provided during ‘content’ phase (labels available no)
- All used different simulation parameters for background (to avoid unrealistic exploits)

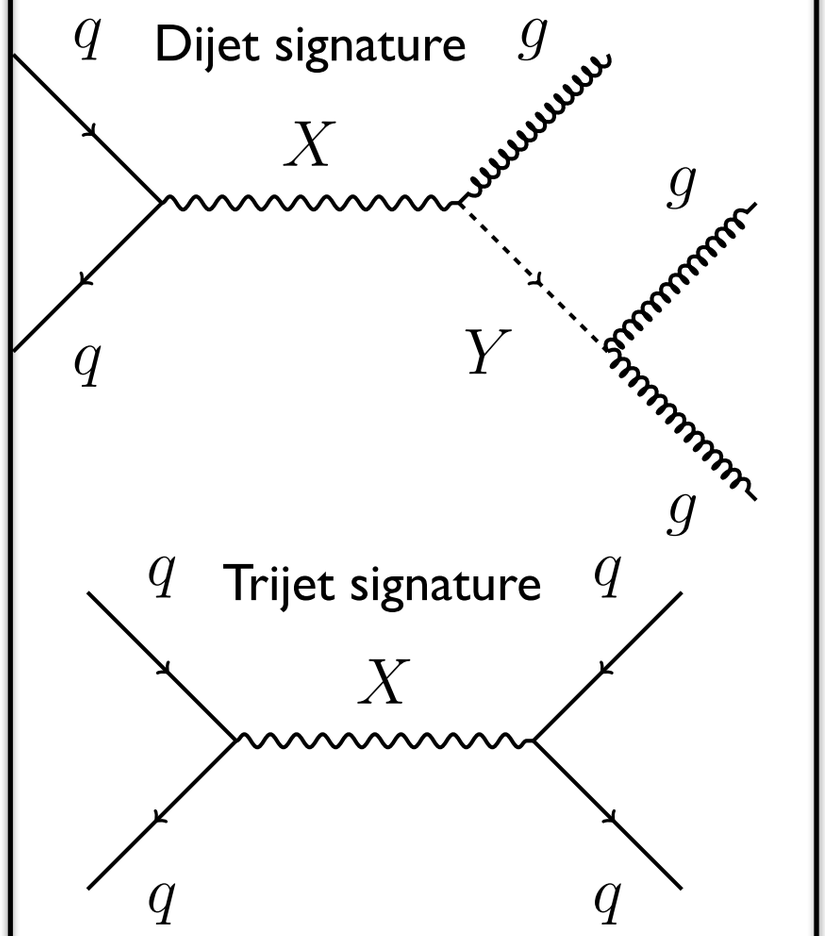
BB1: 834 signal examples
Same event topology as R&D dataset, different masses

might be easy?



BB2: empty

BB3:

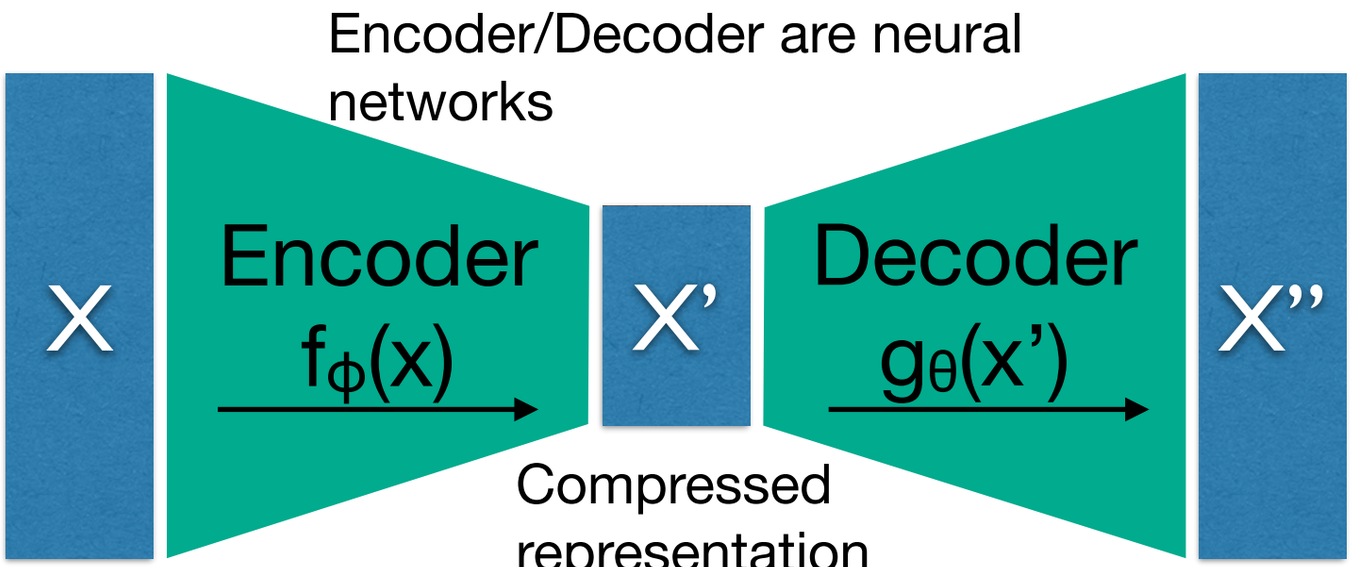


How to build anomaly score?

- Want to find anomaly score **a**
- Should be high for anomalous (signal-like) and low for background-like data
- Some options:
  - **a(x) = 1 / p(x|Background)**

Unsupervised - Autoencoders

- Several autoencoder-type learning approaches
- Underlying assumption is that an autoencoder trained on background dominated sample will have bad reconstruction performance for previously unseen signal

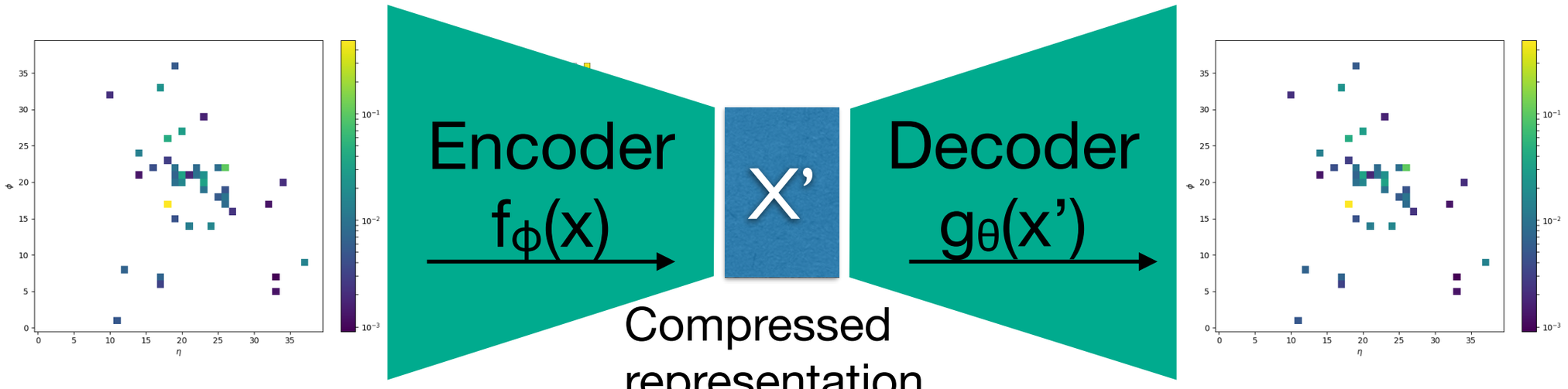


$$\mathcal{L}(x) = ||x - g_\theta(f_\phi(x))||_2$$

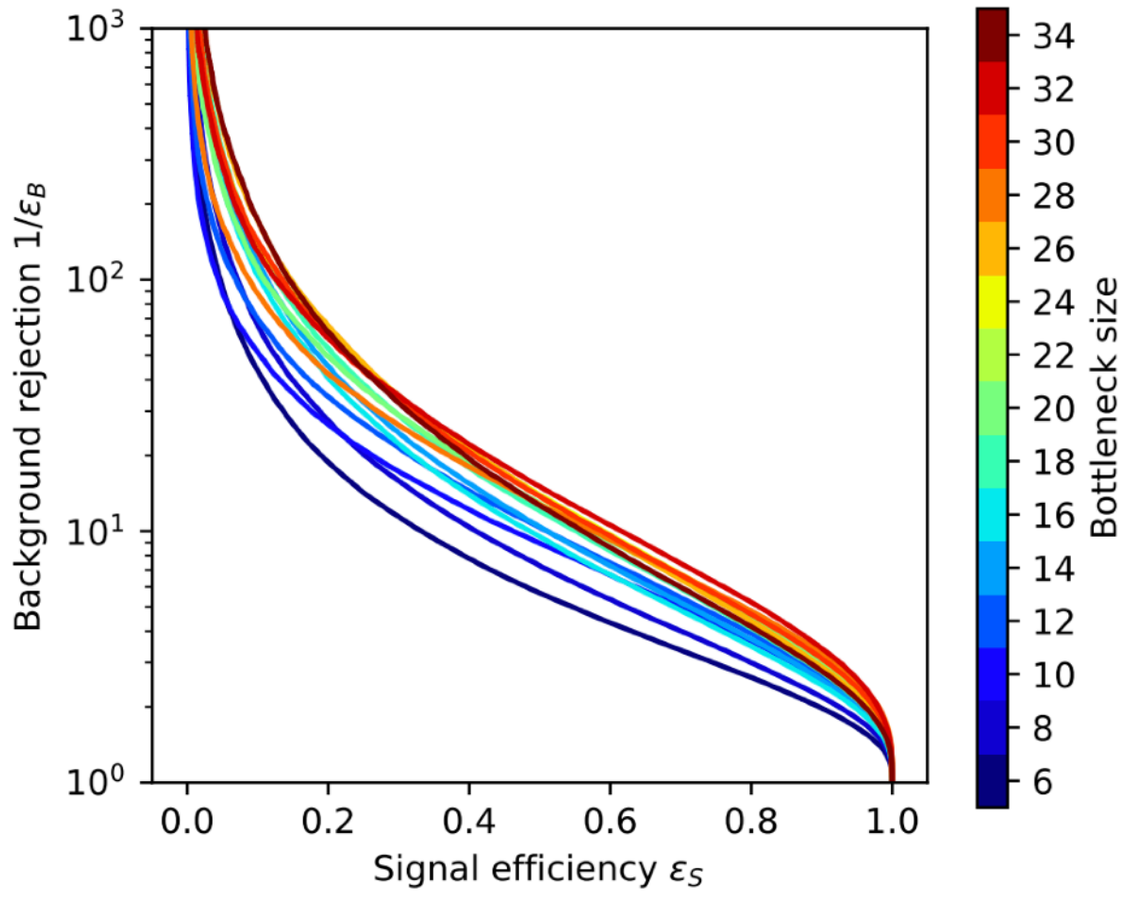
$$a(x) = \mathcal{L}(x)$$

- Differences in data representation
  Data space vs latent space anomaly detection
  Different latent space prior distributions

Physics Application



- Represent data as images
  - Boosted top vs QCD jets (~600 GeV)
    1 jet = 1 image (40x40 pixels, color=energy)
- Train QCD only sample
- Evaluate on mixed top/QCD jet sample



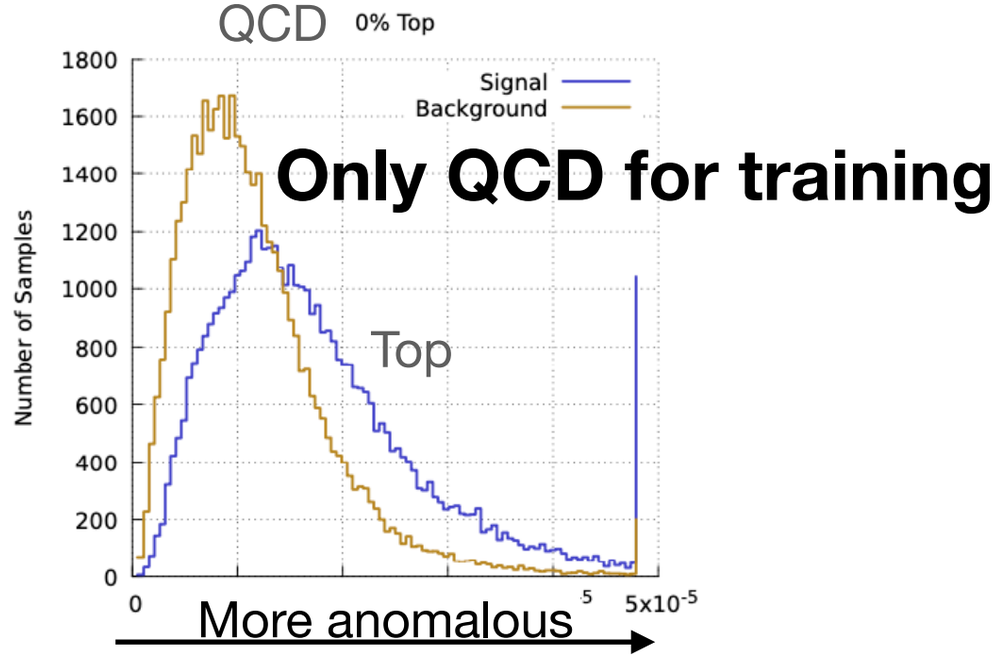
Heimel, GK, Plehn, Thompson, *QCD or What?*, 1808.08979
Farina, Nakai, Shih, *Searching for New Physics with Deep Autoencoders, 1808.08992*

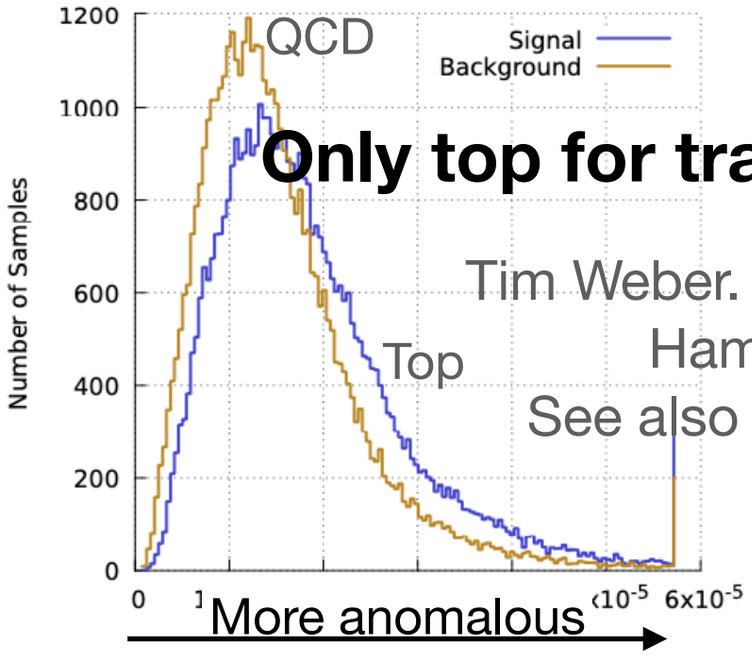
Limitations of AE

Complexity

Complexity

- If anomalies are much simpler (therefore easier to reconstruct):
  a(x) will still be lower, despite never encountered in training
- Observed with naive AE in QCD vs top
  - Train on tops only; top still considered anomaly wrt/ QCD



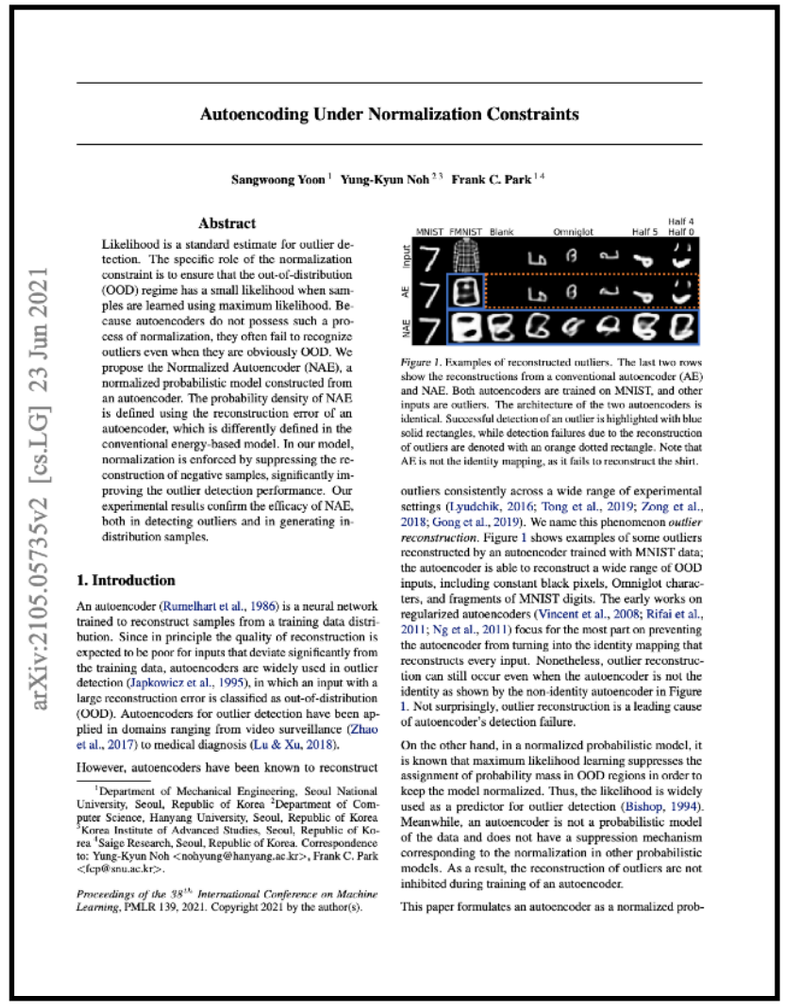


Tim Weber. MSc thesis. Hamburg, 2019;
See also 2104.09051

Complexity

Complexity

- If anomalies are much simpler (therefore easier to reconstruct):
  a(x) will still be lower, despite never encountered in training
- Observed with naive AE in QCD vs top
  - Train on tops only; top still considered anomaly wrt/ QCD
  - Can be overcome - e.g. by **normalised autoencoders** 2105.05735



Autoencoding Under Normalization Constraints

Sangwoong Yoon [1] Yung-Kyun Noh [2 3] Frank C. Park [1 4]

Abstract

Likelihood is a standard estimate for outlier detection. The specific role of the normalization constraint is to ensure that the out-of-distribution (OOD) regime has a small likelihood when samples are learned using maximum likelihood. Because autoencoders do not possess such a process of normalization, they often fail to recognize outliers even when they are obviously OOD. We propose the Normalized Autoencoder (NAE), a normalized probabilistic model constructed from an autoencoder. The probability density of NAE is defined using the reconstruction error of an autoencoder, which is differently defined in the conventional energy-based model. In our model, normalization is enforced by suppressing the reconstruction of negative samples, significantly improving the outlier detection performance. Our experimental results confirm the efficacy of NAE, both in detecting outliers and in generating in-distribution samples.

1. Introduction

An autoencoder (Rumelhart et al., 1986) is a neural network trained to reconstruct samples from a training data distribution. Since in principle the quality of reconstruction is expected to be poor for inputs that deviate significantly from the training data, autoencoders are widely used in outlier detection (Japkowicz et al., 1995), in which an input with a large reconstruction error is classified as out-of-distribution (OOD). Autoencoders for outlier detection have been applied in domains ranging from video surveillance (Zhao et al., 2017) to medical diagnosis (Lu & Xu, 2018).

However, autoencoders have been known to reconstruct outliers consistently across a wide range of experimental settings (Lyudchik, 2016; Tong et al., 2019; Zong et al., 2018; Gong et al., 2019). We name this phenomenon *outlier reconstruction*. Figure 1 shows examples of some outliers reconstructed by an autoencoder trained with MNIST data; the autoencoder is able to reconstruct a wide range of OOD inputs, including constant black pixels, Omniglot characters, and fragments of MNIST digits. The early works on regularized autoencoders (Vincent et al., 2008; Rifai et al., 2011; Ng et al., 2011) focus for the most part on preventing the autoencoder from turning into the identity mapping that reconstructs every input. Nonetheless, outlier reconstruction can still occur even when the autoencoder is not the identity as shown by the non-identity autoencoder in Figure 1. Not surprisingly, outlier reconstruction is a leading cause of autoencoder's detection failure.

Figure 1. Examples of reconstructed outliers. The last two rows show the reconstructions from a conventional autoencoder (AE) and NAE. Both autoencoders are trained on MNIST, and other inputs are outliers. The architecture of the two autoencoders is identical. Successful detection of an outlier is highlighted with blue solid rectangles, while detection failures due to the reconstruction of outliers are denoted with an orange dotted rectangle. Note that AE is not the identity mapping, as it fails to reconstruct the shirt.

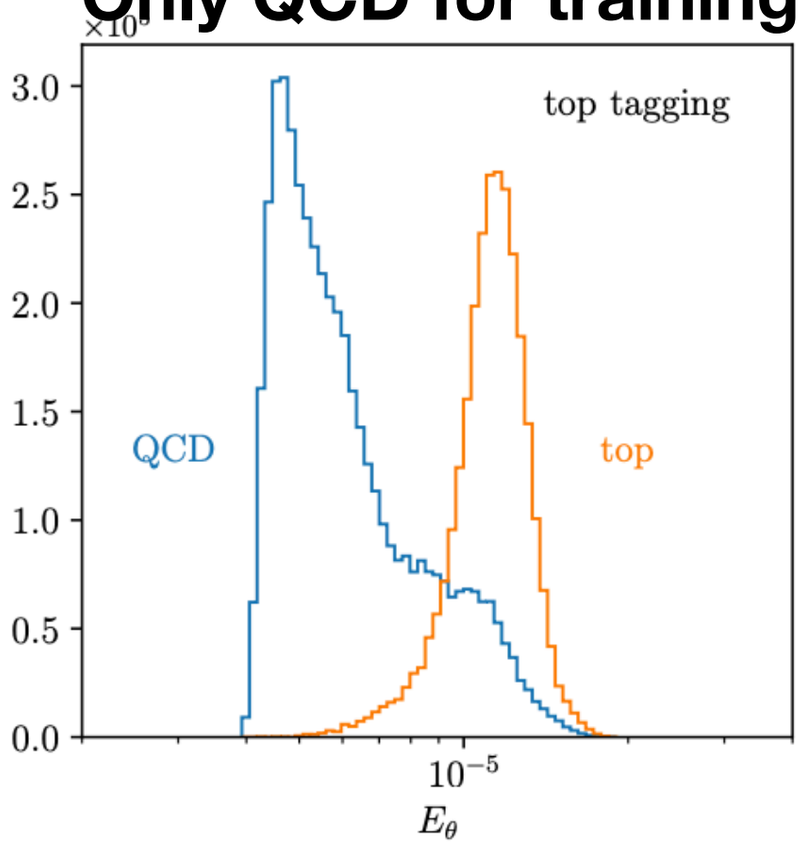
On the other hand, in a normalized probabilistic model, it is known that maximum likelihood learning suppresses the assignment of probability mass in OOD regions in order to keep the model normalized. Thus, the likelihood is widely used as a predictor for outlier detection (Bishop, 1994). Meanwhile, an autoencoder is not a probabilistic model of the data and does not have a suppression mechanism corresponding to the normalization in other probabilistic models. As a result, the reconstruction of outliers are not inhibited during training of an autoencoder.

This paper formulates an autoencoder as a normalized prob-

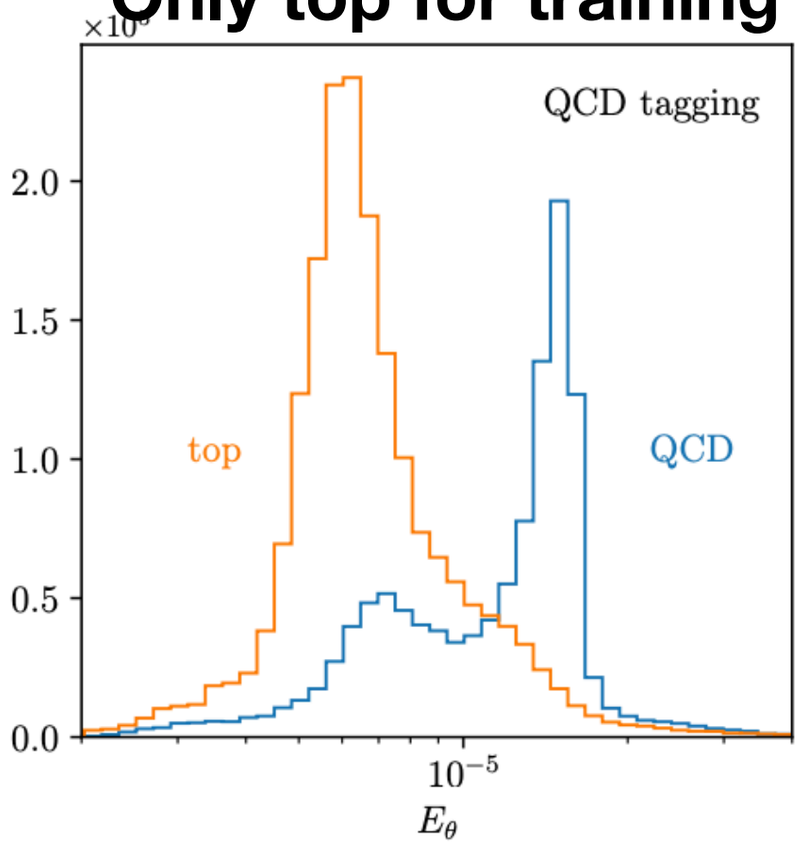
[1]Department of Mechanical Engineering, Seoul National University, Seoul, Republic of Korea [2]Department of Computer Science, Hanyang University, Seoul, Republic of Korea [3]Korea Institute of Advanced Studies, Seoul, Republic of Korea [4]Saige Research, Seoul, Republic of Korea. Correspondence to: Yung-Kyun Noh <nohyung@hanyang.ac.kr>, Frank C. Park <fcp@snu.ac.kr>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

Only QCD for training



Only top for training



2206.14225

Coordinate Dependence

- 2012.03808 (Le Lan, Dinh)

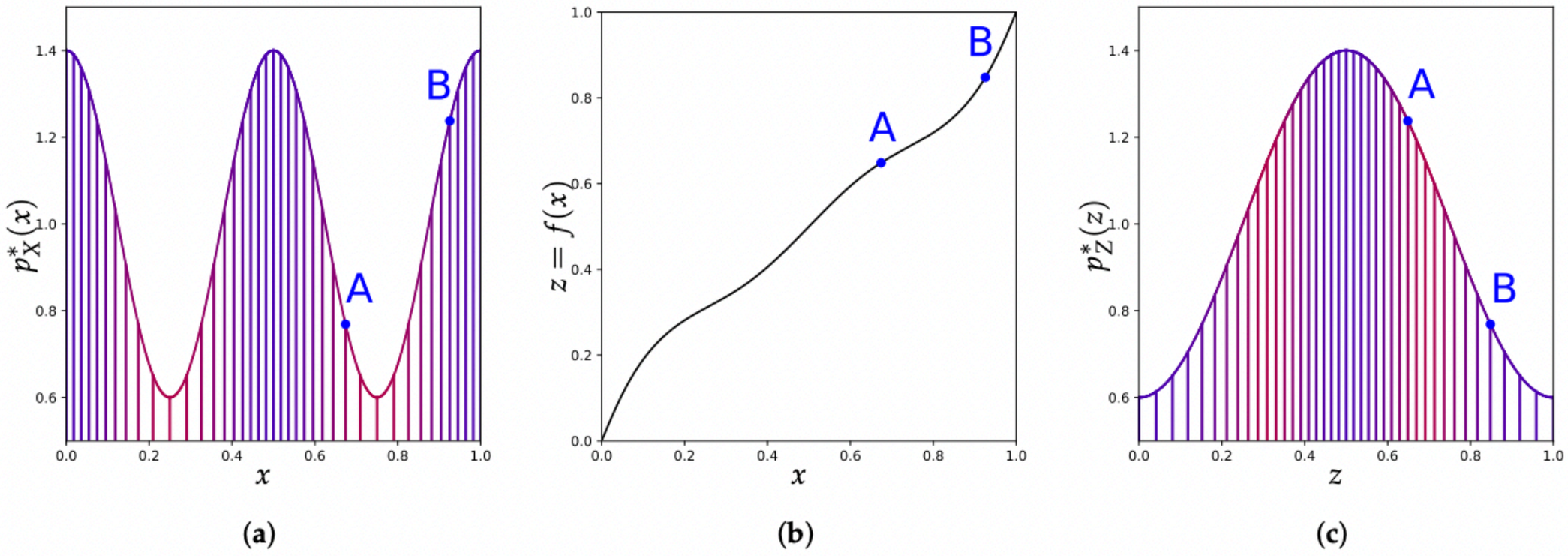


Figure 1. An invertible change of representation can affect the relative density between two points A and B, which has been interpreted as their relative regularity. (**a**) An example of a distribution density p_X^*. (**b**) Example of an invertible function f from $[0,1]$ to $[0,1]$. (**c**) Resulting density p_Z^* as a function of the new axis $z = f(x)$. In (**a**,**c**) points with high original density $p_X^*(x)$ are in blue and red for low original density.

Brief Recap

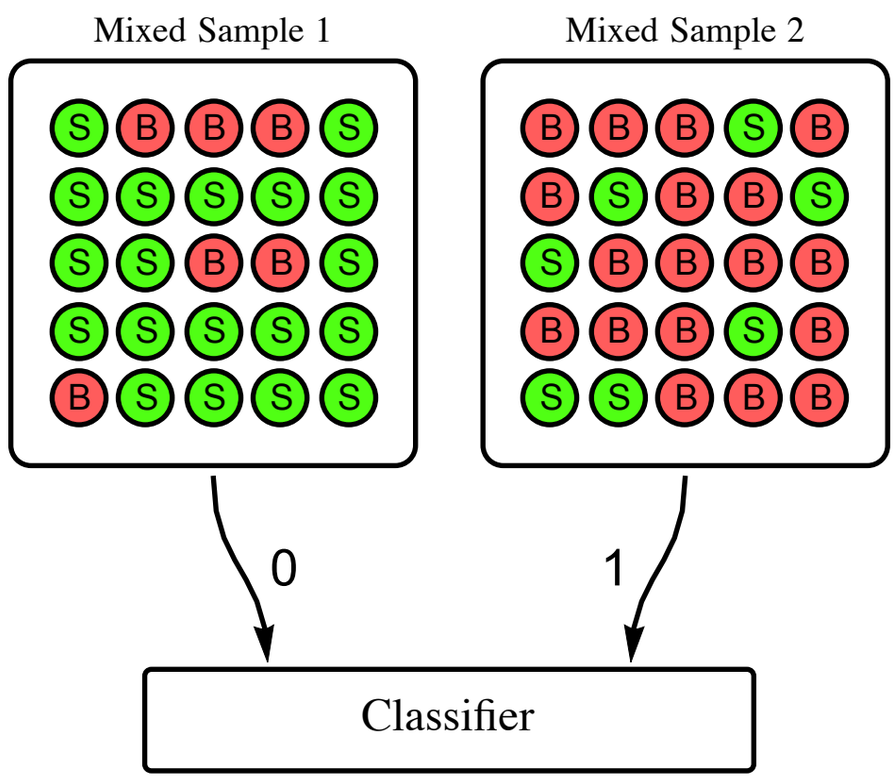
- Autoencoders (and VAEs) find anomalies by looking for regions of low background density

- Conceptually straightforward and computationally cheap but potential issues of coordinate dependence and complexity

How to build anomaly score?

- Want to find anomaly score **a**
- Should be high for anomalous (signal-like) and low for background-like data
- Some options:
  - a(x) = 1 / p(x|Background)
  - **a(x) = p(x|Signal) / p(x|Background)**

Example I: Label Noise aka CWoLA



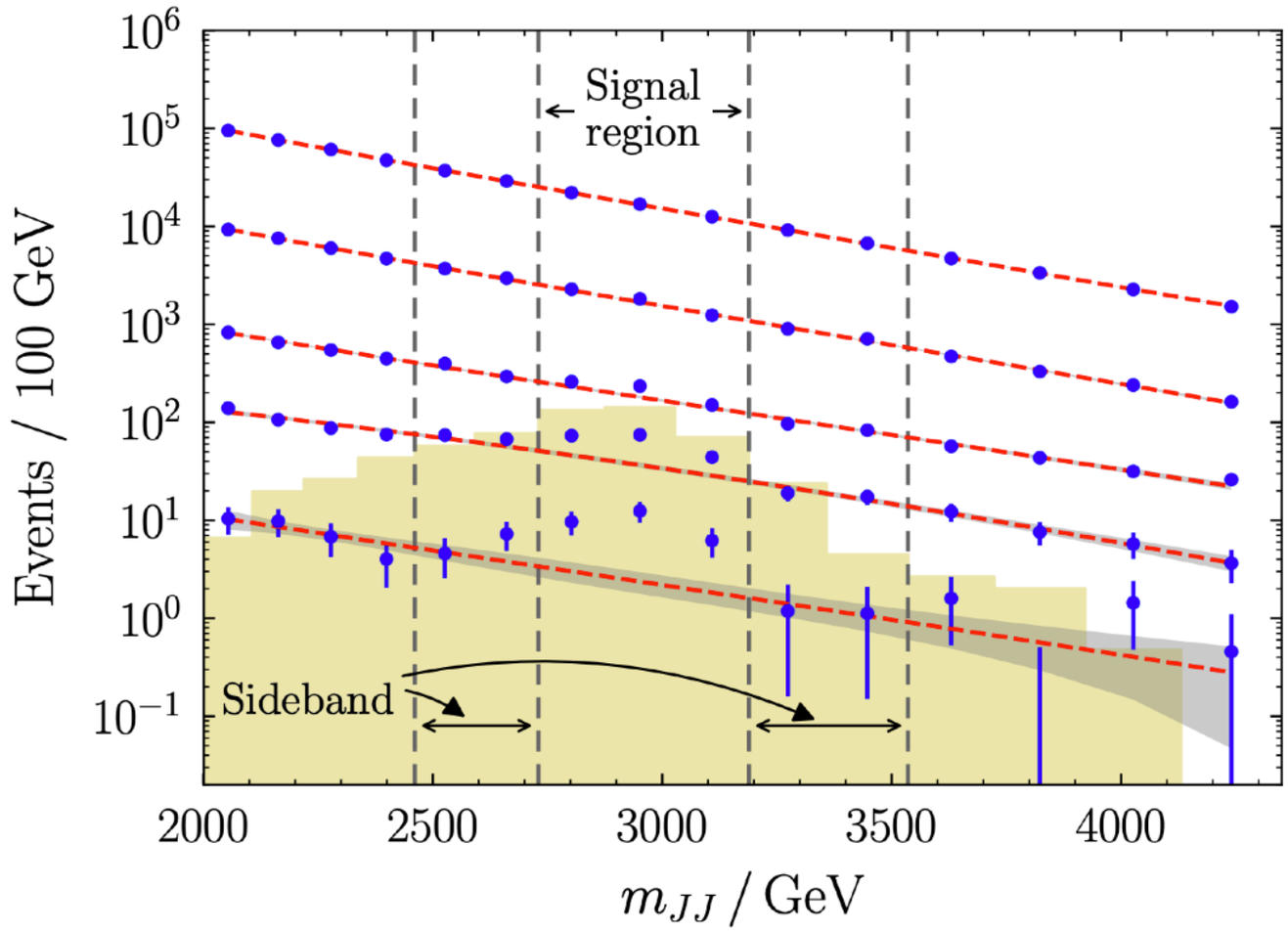
$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1-f_1)\, p_B}{f_2\, p_S + (1-f_2)\, p_B} = \frac{f_1\, L_{S/B} + (1-f_1)}{f_2\, L_{S/B} + (1-f_2)}$$

Important observation:

A classifier (i.e. a neural network) trained to distinguish two mixed samples learns to distinguish the components

Classification without labels: Learning from mixed samples in high energy physics, EM Metodiev, B Nachman, J Thaler, 1708.02949

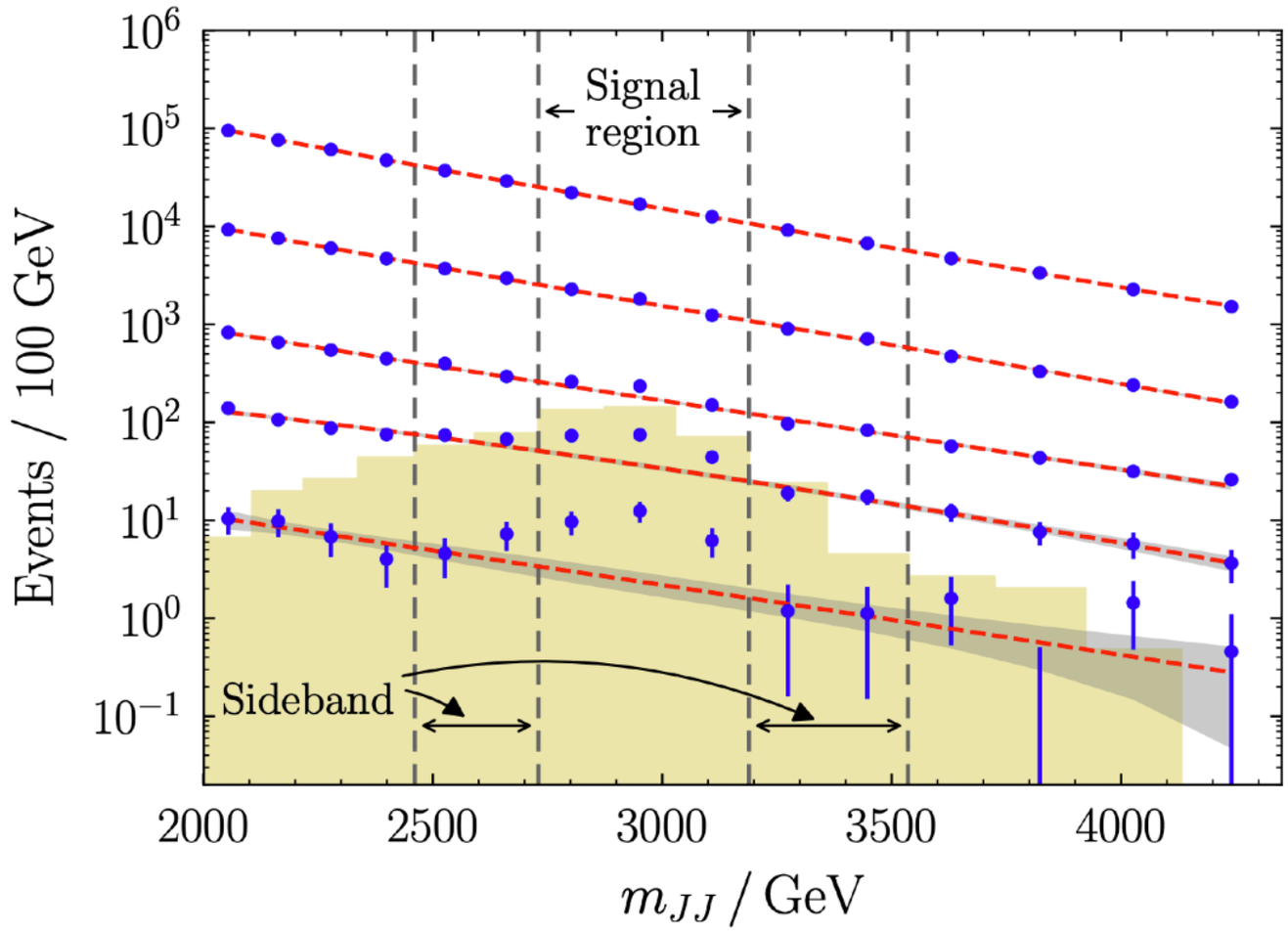
Example I: Label Noise



- Train a classifier to distinguish signal region (SR) and side-band (SB).
- If there is intrinsic difference (presence of signal) it will be able to distinguish, otherwise it will not.

Collins, Howe, Nachman, *Anomaly Detection for Resonant New Physics with Machine Learning*, 1805.02664

Example I: Label Noise



- Train a classifier to distinguish signal region (SR) and side-band (SB).
- If there is intrinsic difference (presence of signal) it will be able to distinguish, otherwise it will not.
- **Major downside: Additional features that correlate with the side-band variable**

Collins, Howe, Nachman, *Anomaly Detection for Resonant New Physics with Machine Learning*, 1805.02664

Example II: Density Estimation

Per Neyman-Pearson: Likelihood-ratio is optimal test statistic
Unfortunatly, p(x|anomaly) is not available

$$L_{S/B} = \frac{p(x|\text{anomaly})}{p(x|\text{normal})}$$

Build data/background ratio:

$$L_{D/B} = \frac{p(x)}{p(x|\text{normal})}$$

Approximate background density using control measurement (e.g. sideband)

$$L_{D/B} \approx \frac{p(x)}{\tilde{p}(x|\text{normal})}$$

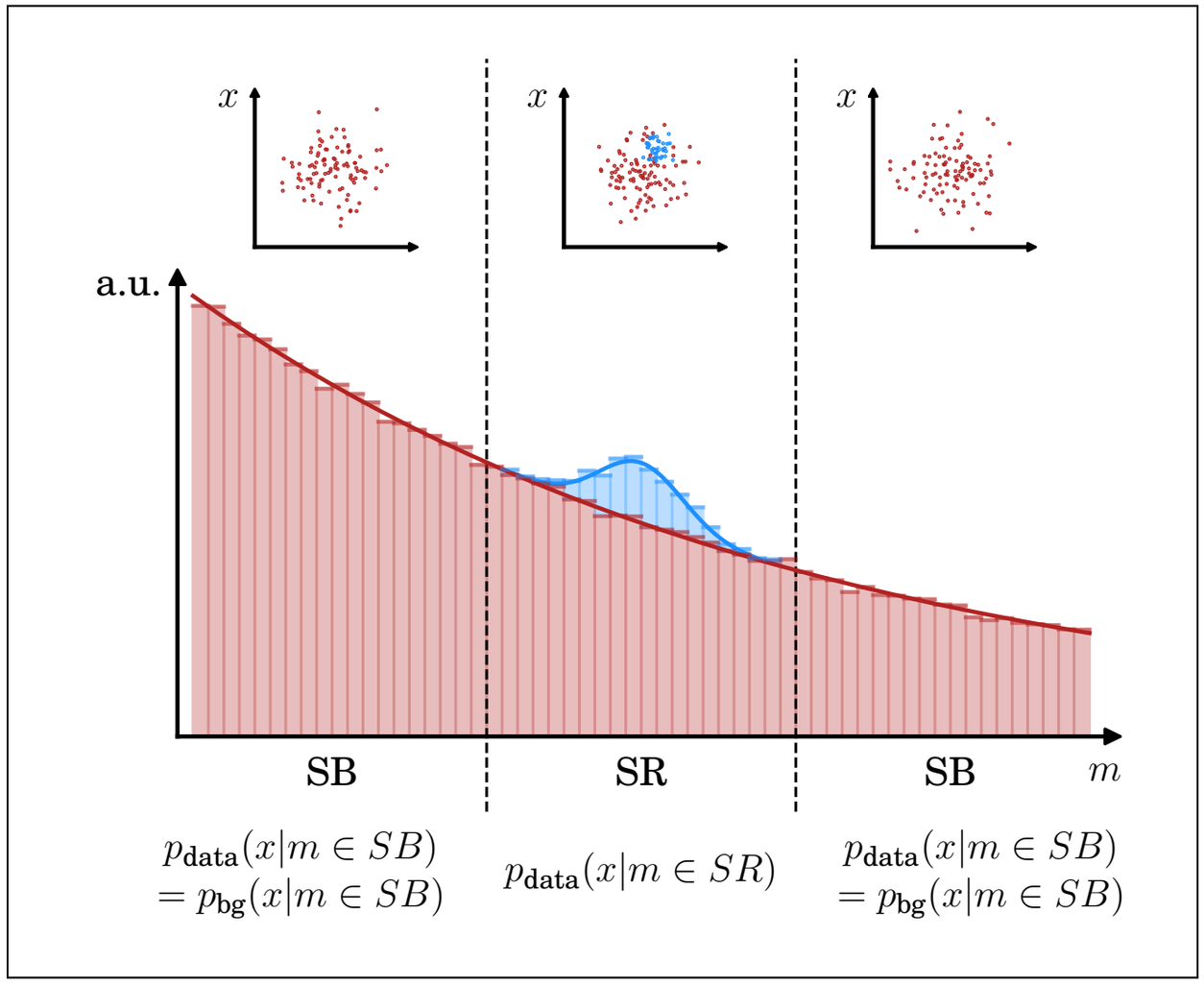
Expand

$$p(x) = f_{\text{normal}}\, p(x|\text{normal}) + f_{\text{anomaly}}\, p(x|\text{anomaly})$$

And insert:

$$L_{D/B} \approx f_{\text{normal}} + f_{\text{anomaly}} \frac{p(x|\text{anomaly})}{\tilde{p}(x|\text{normal})}$$

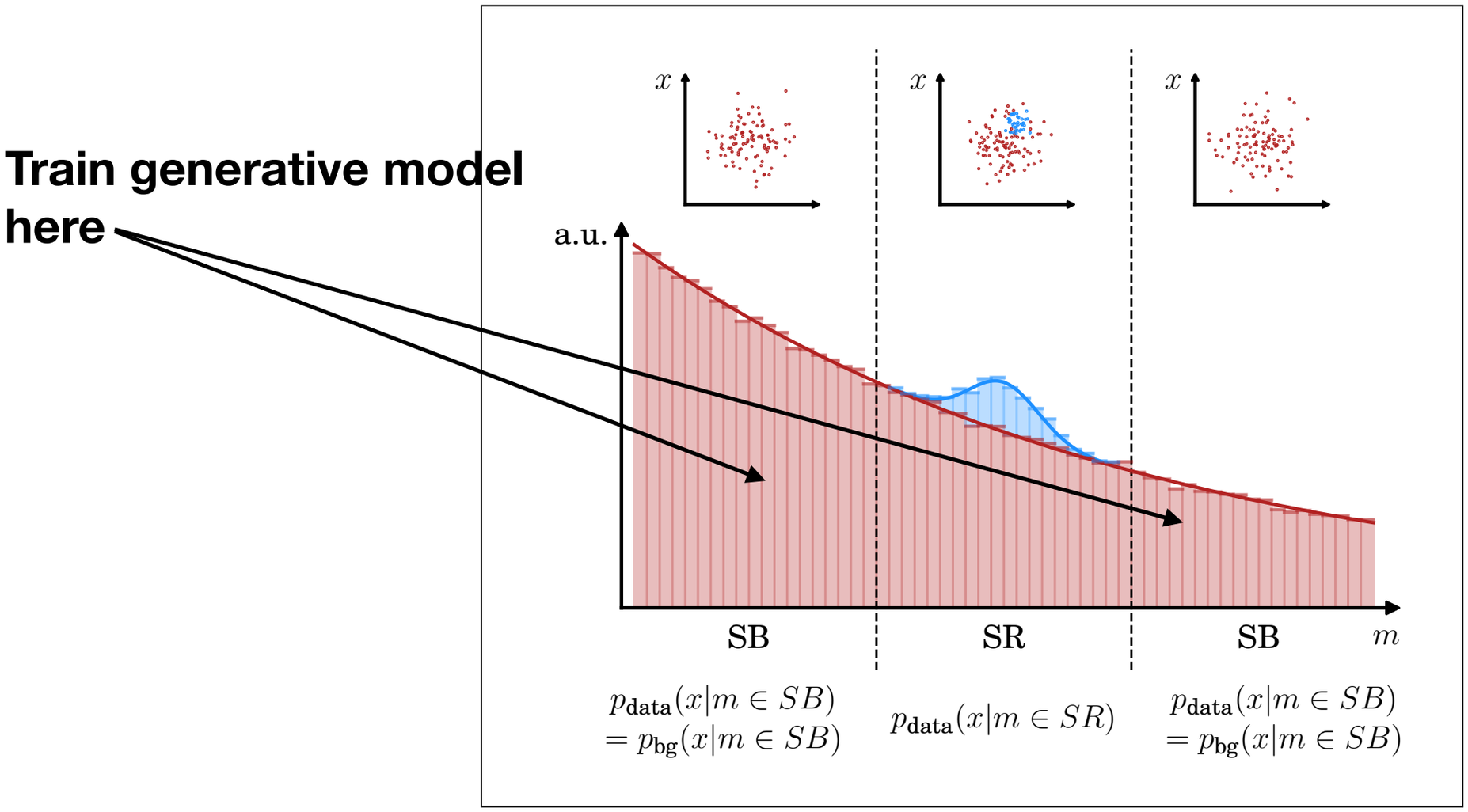
Generative models for anomaly detection



- 1): Choose one feature (m) in which to search for resonances

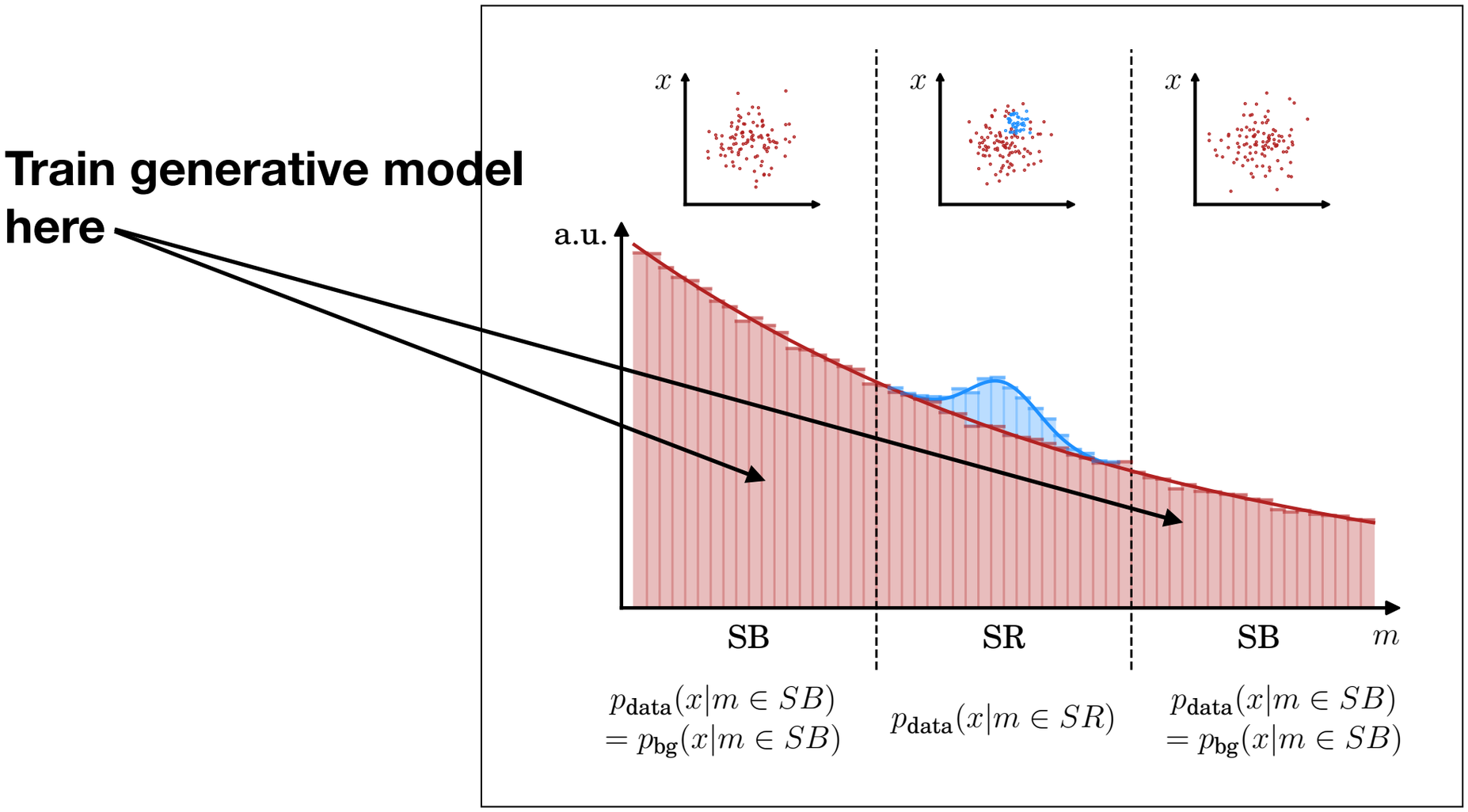
2109.00546

Generative models for anomaly detection



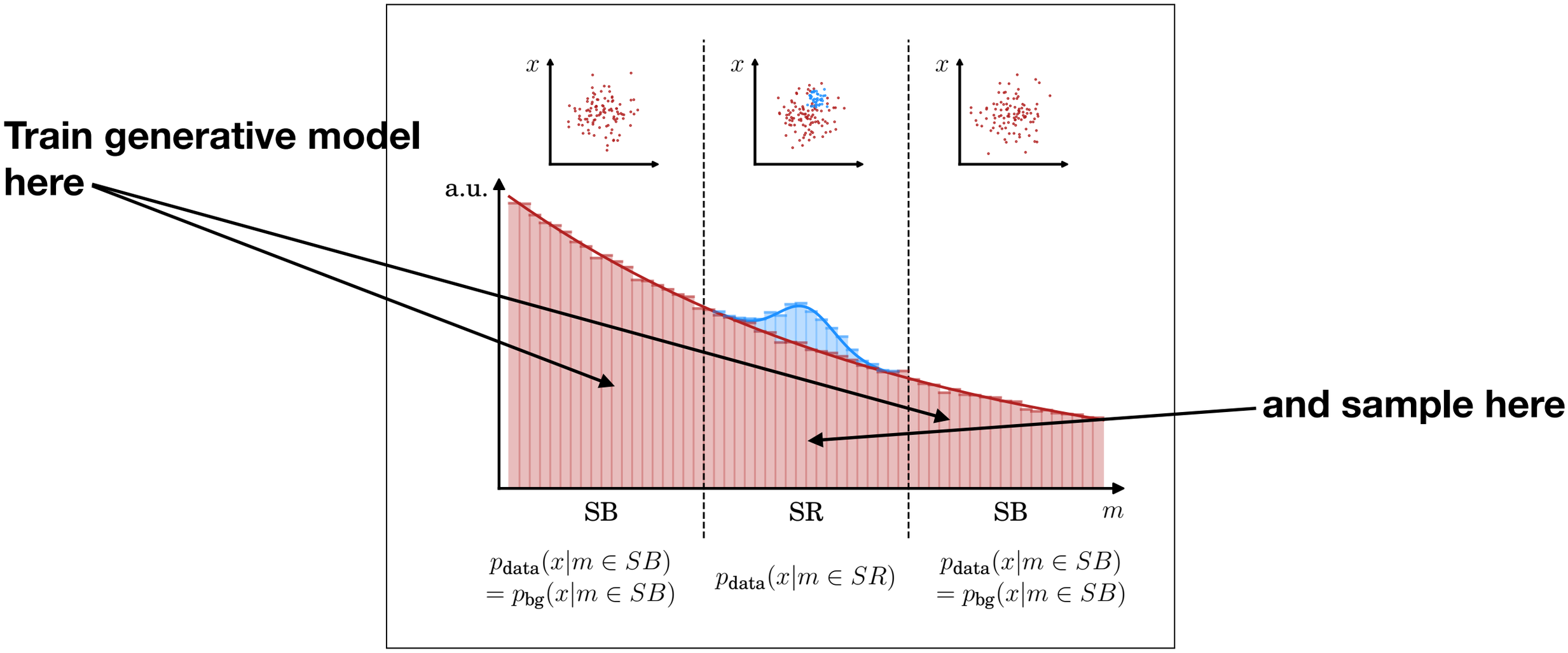
- 1): Choose one feature (m) in which to search for resonances
- 2): Use m divide spectrum into non-overlapping regions. Designate one as signal region (SR), others as sidebands (SB). Repeat the following for all choices of SR

Generative models for anomaly detection



- 3) Train a generative model p(x|m) on auxiliary features in SB (used MAF, other choices including GAN/VAE possible as well)

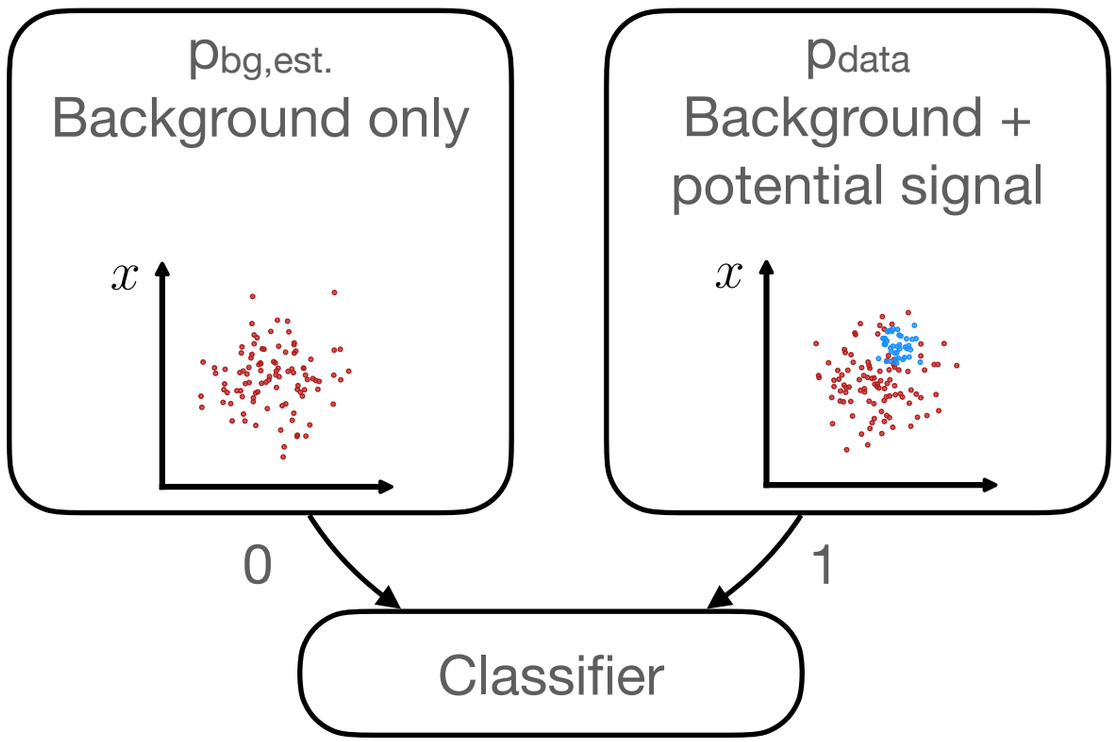
Generative models for anomaly detection



- 3) Train a generative model p(x|m) on auxiliary features in SB
- 4) Sample from p(x|m) in SR. Designate as $p_{\text{bg,est.}}$

2109.00546

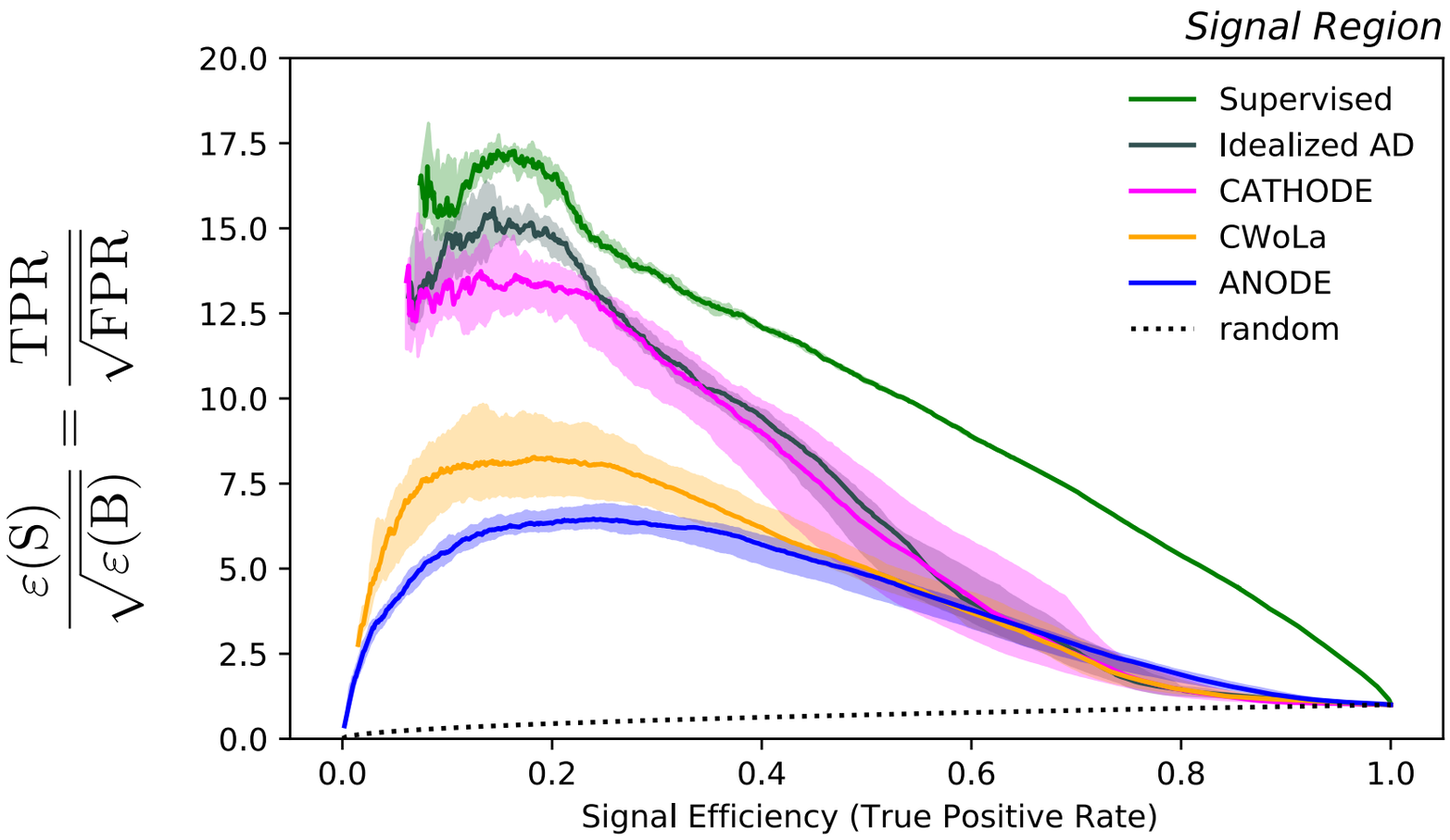
Generative models for anomaly detection



- 3) Train a generative model p(x|m) on auxiliary features in SB
- 4) Sample from p(x|m) in SR. Designate as $p_{bg,est}$
- 5) Train binary classifer between p_{data} and $p_{bg,est.}$

2109.00546

Generative models for anomaly detection

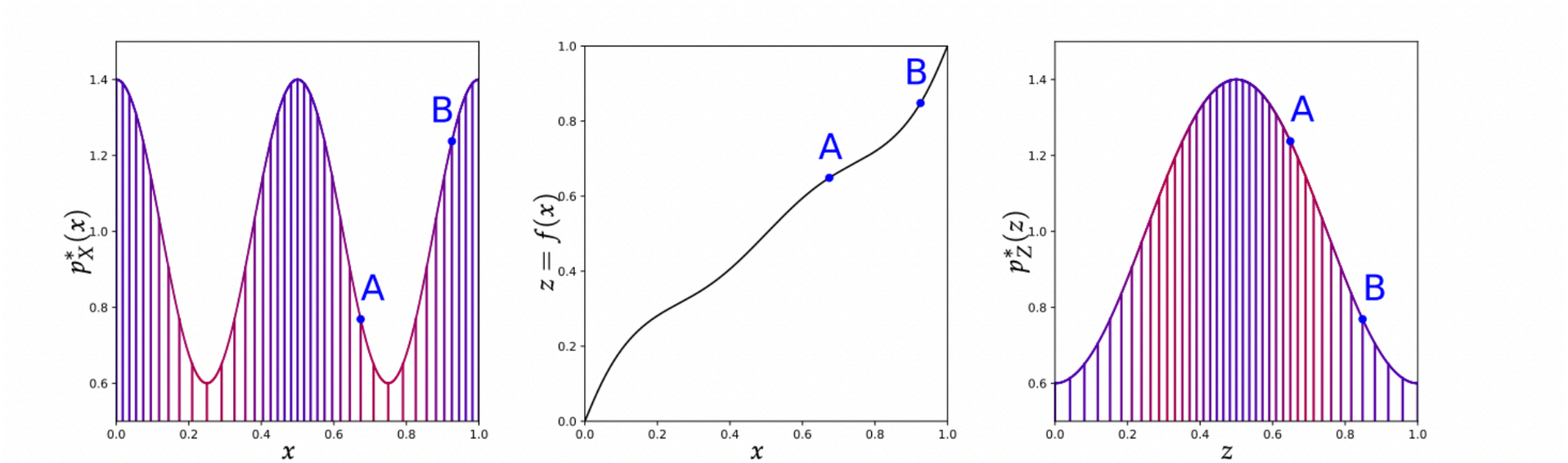


- 3) Train a generative model p(x|m) on auxiliary features in SB
- 4) Sample from p(x|m) in SR. Designate as $p_{bg,est.}$
- 5) Train binary classifer between p_{data} and $p_{bg,est.}$ (mixed sample classifer)
- 6) Cut on high classifier scores to enrich sample with anomalies
  (and perform statistical analysis)

2109.00546

Comments on anomaly detection

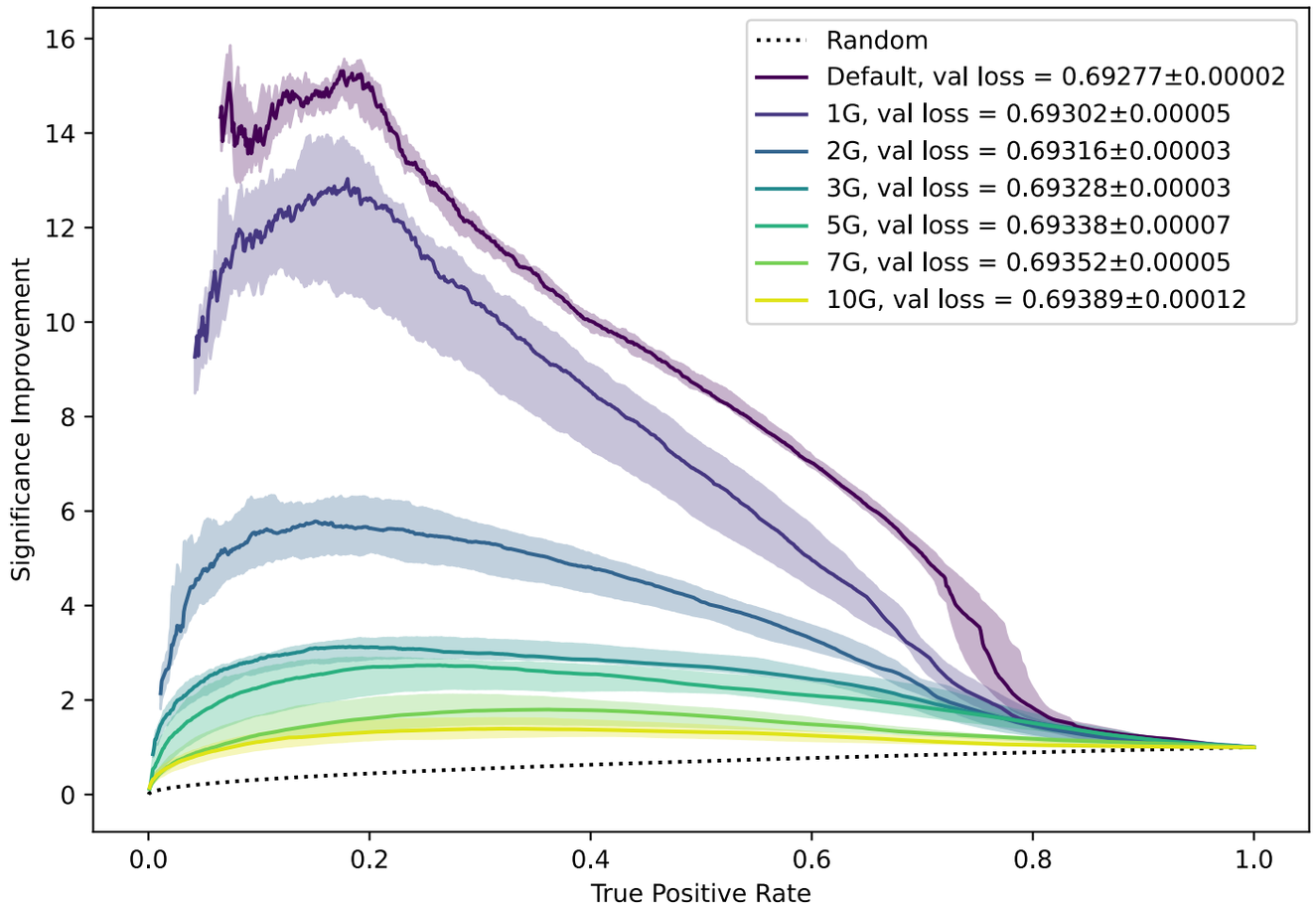
- As CATHODE approximates a likelihood ratio, it should be robust compared to methods that only use $p_{\text{Background}}$ (e.g. autoencoders)



2012.03808 (Lan & Dinh)

Comments on anomaly detection

- As CATHODE approximates a likelihood ratio, it should be robust compared to methods that only use $p_{Background}$ (e.g. autoencoders)
- However, still can be sensitive to choice of input features



Comments on anomaly detection

- As CATHODE approximates a likelihood ratio, it should be robust compared to methods that only use $p_{Background}$ (e.g. autoencoders)
- However, still can be sensitive to choice of input features
- Need also consider
  - Shaping of distributions under tighter anomaly detection cuts
  - Higher dimensional input data

Closing

- Deep Learning for particle physics is rapidly developing solutions to a wide range of problems
  - Classification
  - Anomaly detection
  - Robustness and uncertainties
  - Efficient generation
  - Fast processing
- Physics encounters challenges of complex data and large volumes with potential relevance to other domains
- **Contact**:
  - Email: gregor.kasieczka@uni-hamburg.de
  - Twitter: @GregorKasieczka
  - Webpage: https://www.physik.uni-hamburg.de/en/iexp/gruppe-kasieczka.html