Searching for new physics without knowing the signal model

- Machine Learning
- Automatization
- Data Derived Signal Regions

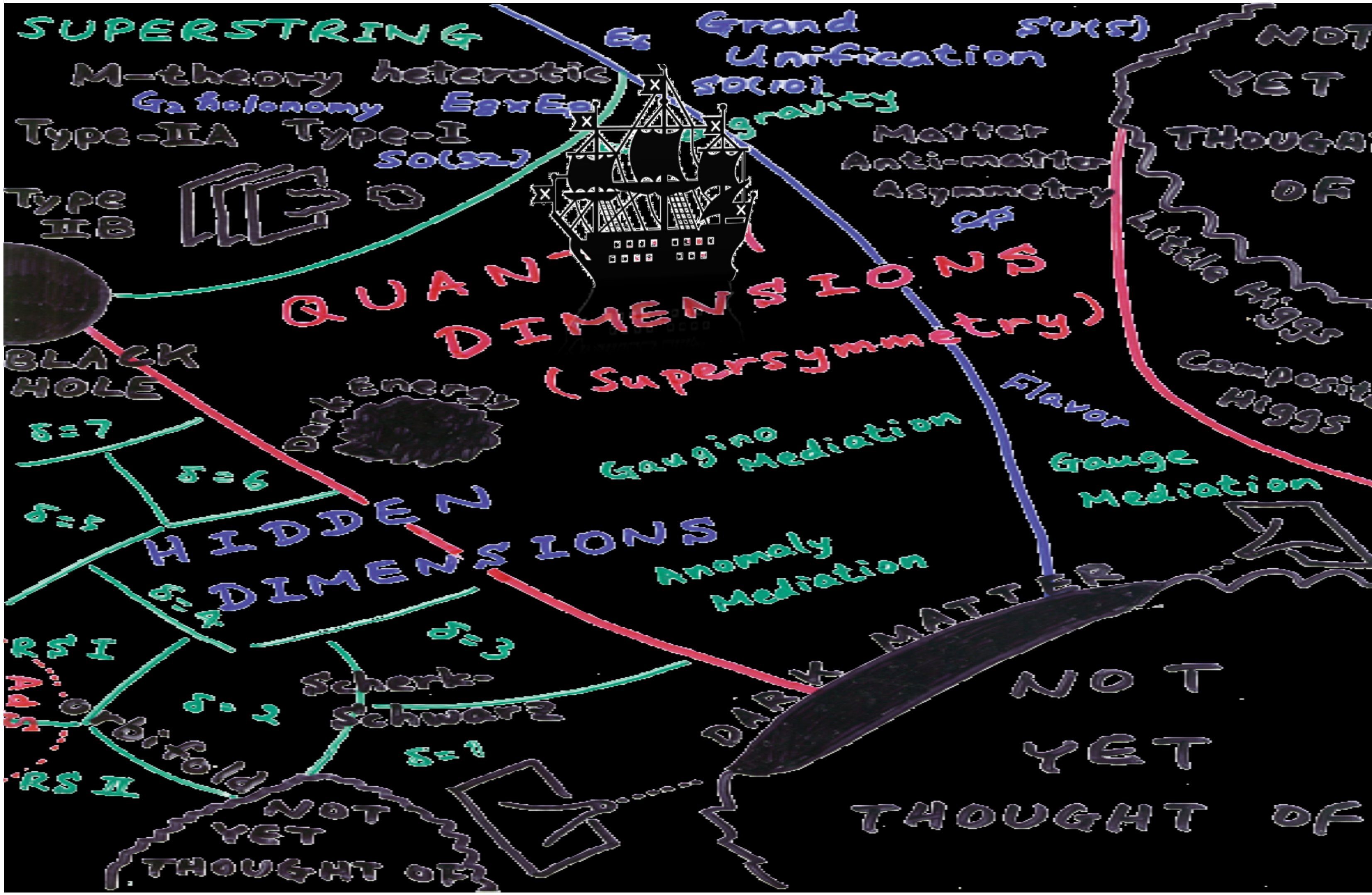
Sascha Caron
(Radboud University
and Nikhef)

Outline

- 1. Introduction:
- 2. Particle Physics (Particle DM) situation in 2019
- 3. DM searches with Deep Networks
- 4. Learning DM/HEP models
- 5. Unsupervised searches
- 6. Simulation without MC simulators

Very personal selection of topics/papers/etc.
No real overview talk…

What do we expect to find at the LHC?



One physicist's schematic view of particle physics in the 21st century
(Courtesy of Hitoshi Murayama)

2019: What has changed ?

- **We have not seen any signs of new physics (no SUSY and no convincing signal of anything else !)**

- **We got a “toolbox“ from the future**

2019: What has changed ?

- **We have not seen any signs of new physics (no SUSY and no convincing signal of anything else !)**

→ Was that expected ?
→ Maybe yes …

Higgs: You got a new toy, it‘s a playmobil castle with a size between 1-100 cm. Can you find it ?

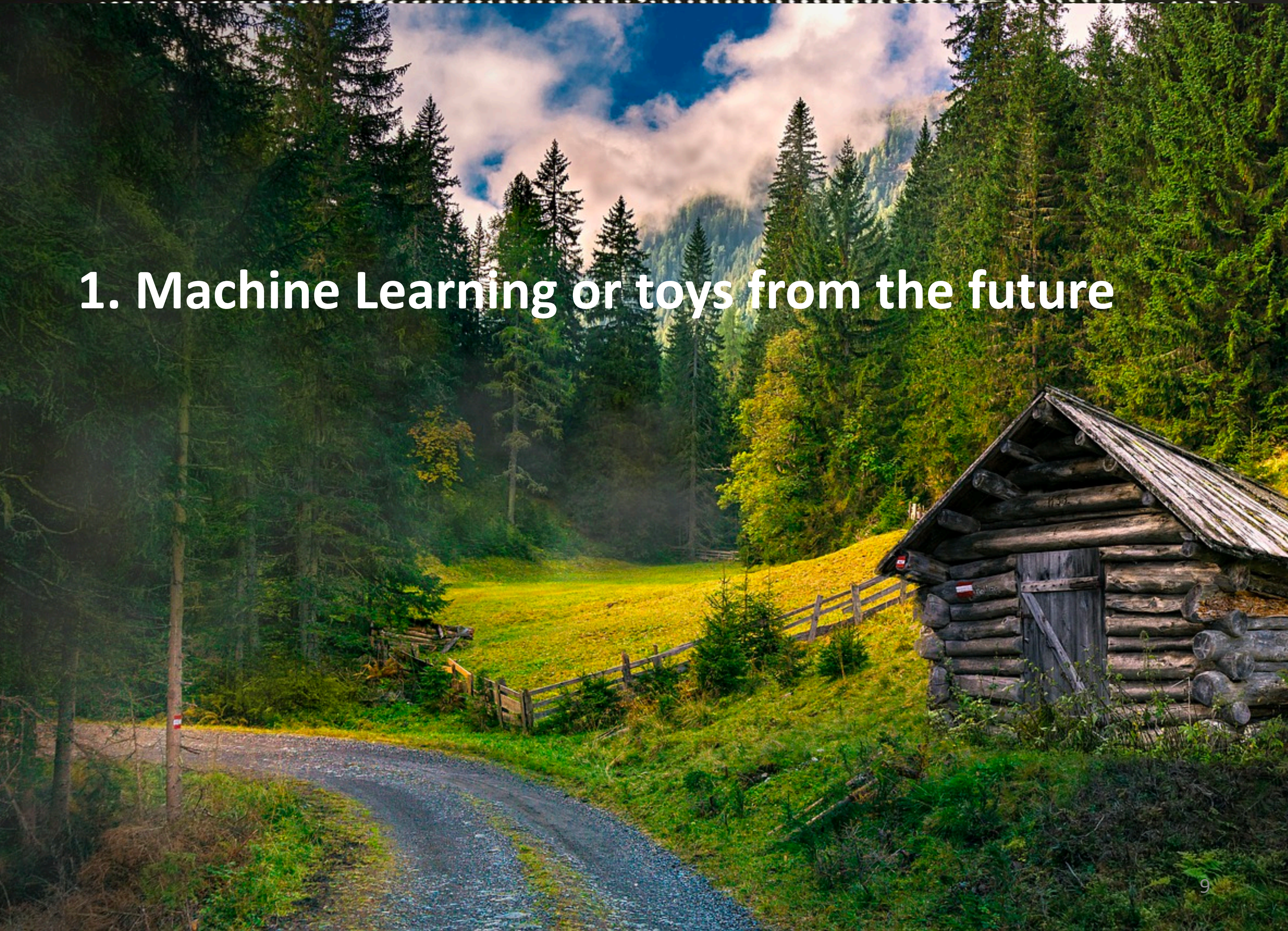
Today: I have a new toy for you, I put it somewhere in your room. The size is 0.1-100 cm. Can you find it ?

Could work to implement more of automatization for particle physics to „scan“ the full room for something interesting…

→This can help LHC, but might also work for astrophysics

→We can embed into this “scan“ our prejudice how new physics looks like, e.g. in this case it would be „toy“ detection software trained on all known toys…

1. Machine Learning or toys from the future



What is Machine Learning ?

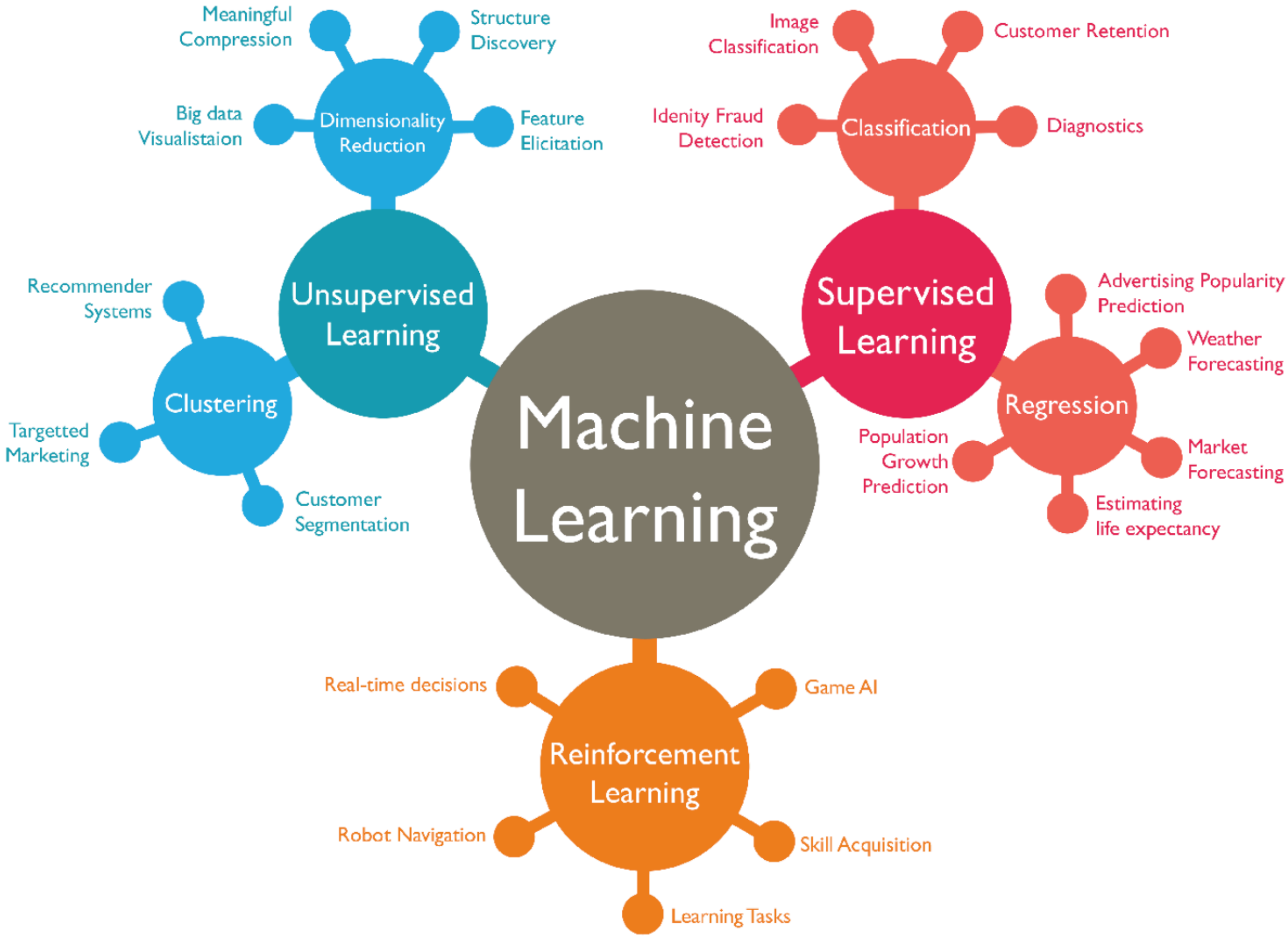


Image via Abdul Rahid

Most important example: Supervised Learning

Computer systems "learn" with data

Actually the computer "learn/derives and fits" a continuous estimator $\hat{\mathbf{f}}(\vec{x})$ for an unknown function $\mathbf{f}(\vec{x})$ from i discrete data points $\vec{x}_i$ with known function values $\mathbf{f}(\vec{x}_i)$.

i discrete data points $\vec{x}_i$ with known function values $\mathbf{f}(\vec{x}_i)$.

is called the "**training set**".

Determining $\hat{\mathbf{f}}(\vec{x})$ is called "**training**".

The axis values of the $\vec{x}_i$ values are called "**features**".

Neural Networks

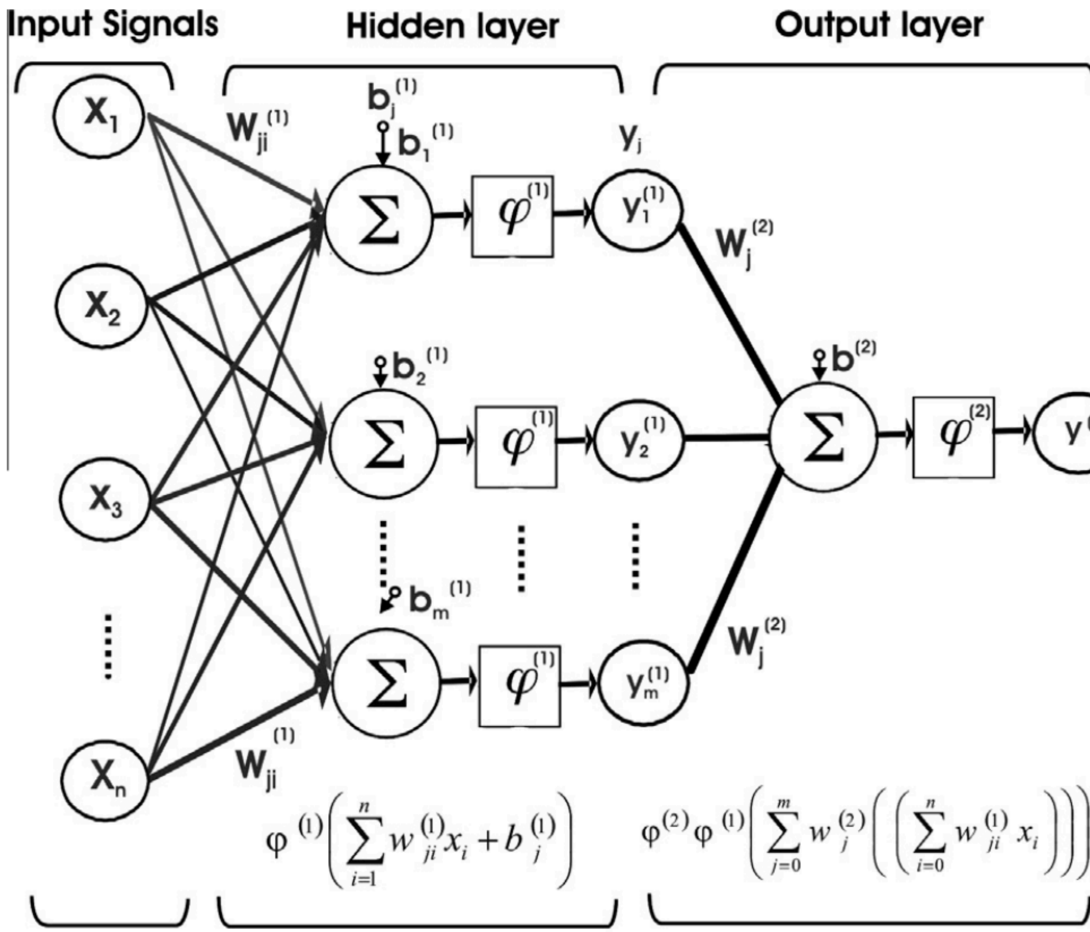
- Let us now make the basis function itself nonlinear combinations of its inputs

$$
\begin{aligned}
y^{(2)} &= \varphi^{(2)}\left(\sum_{j=1}^{m}\left(w_j^{(2)} y_j^{(1)} + b^{(2)}\right)\right) \\
&= \varphi^{(2)}\left(\sum_{j=1}^{m} w_j^{(2)} \varphi^{(1)}\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b^{(2)}\right)
\end{aligned}
$$

And phi is a non-linear activation function, b is called bias

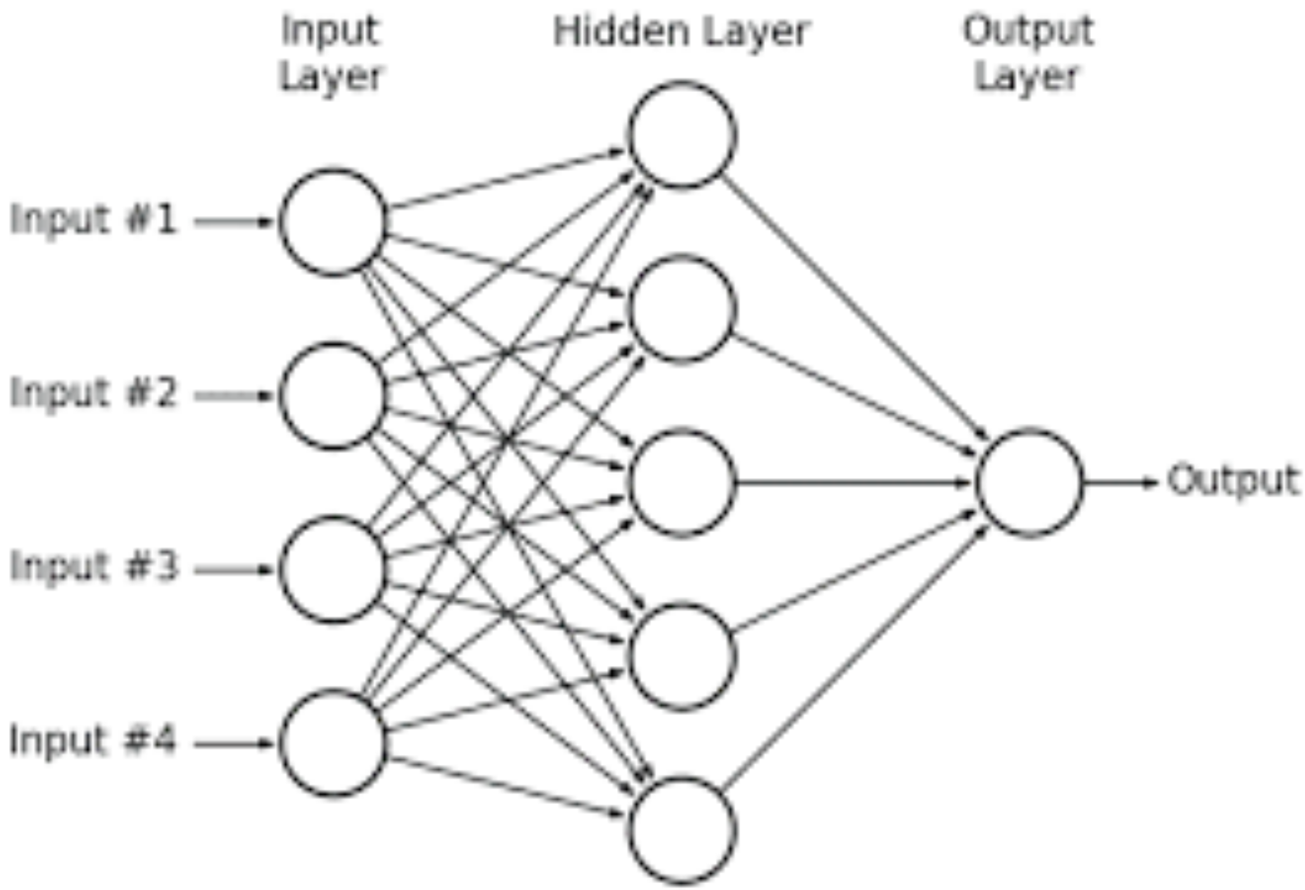
(bias allows to "shift" the activation function

$$
\begin{aligned}
y^{(2)} &= \varphi^{(2)}\left(\sum_{j=1}^{m}\left(w_j^{(2)} y_j^{(1)} + b^{(2)}\right)\right) \\
&= \varphi^{(2)}\left(\sum_{j=1}^{m} w_j^{(2)} \varphi^{(1)}\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b^{(2)}\right)
\end{aligned}
$$



y

$$
\begin{aligned}
y^{(2)} &= \varphi^{(2)}\left(\sum_{j=1}^{m}\left(w_j^{(2)} y_j^{(1)} + b^{(2)}\right)\right) \\
&= \varphi^{(2)}\left(\sum_{j=1}^{m} w_j^{(2)} \varphi^{(1)}\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b^{(2)}\right)
\end{aligned}
$$



This is a 3 layer (1 hidden layer) feedforward (multilayer) perceptron

→This is the “simplest network”

“Training”:

Finding the set of weights which minimize the error function

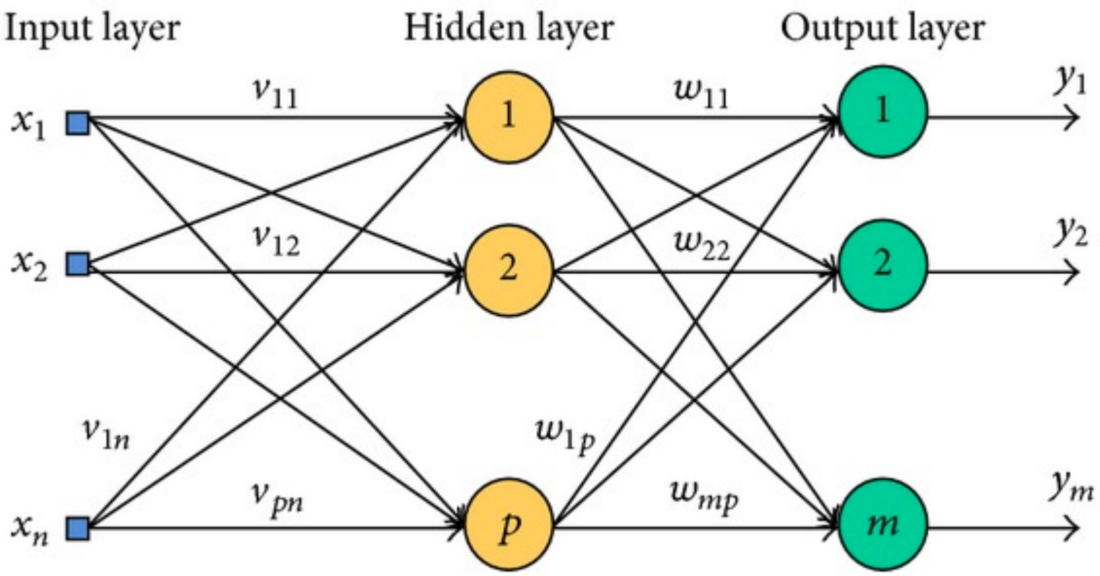
Example:
NNpdfs

Fitting pdfs without assuming the underlying function

10. **Parton distributions for the LHC Run II**
NNPDF Collaboration (Richard D. Ball (U. Edinburgh, Higgs Ctr. Theor. Phys. & CERN) *et al.*). Oct 31, 2014. 138 pp.
Published in **JHEP 1504 (2015) 040**
EDINBURGH-2014-15, IFUM-1034-FT, CERN-PH-TH-2013-253, OUTP-14-11P, CAVENDISH-HEP-14-11
DOI: 10.1007/JHEP04(2015)040
e-Print: **arXiv:1410.8849 [hep-ph]** | PDF
References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
CERN Document Server; ADS Abstract Service; Link to Article from SCOAP3
Detailed record - Cited by 1054 records 1000+

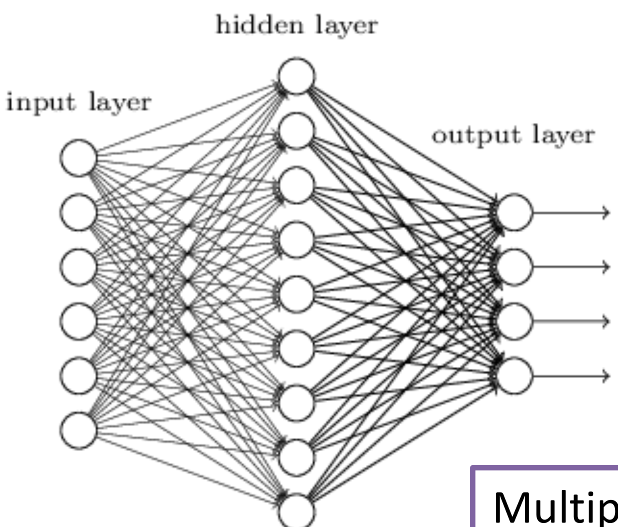
Neural Networks

- Of course we can have multiple output nodes



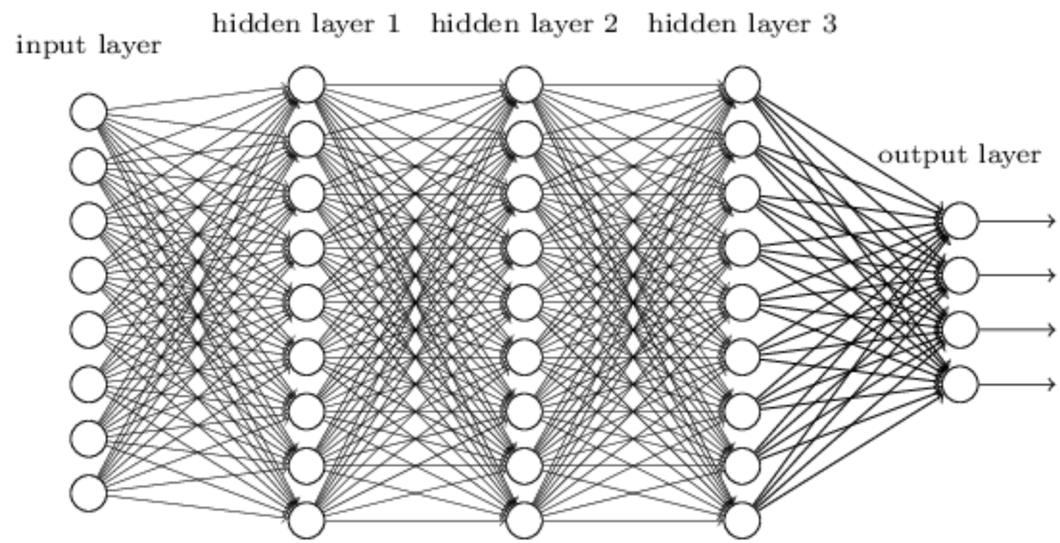
… or multiple hidden layers…

"Non-deep" feedforward neural network



Multiple Output Nodes !

Deep neural network



We often use 5-10 layers

2014 First deep network in HEP (begin 2018 we had 50 on arxiv)

Searching for Exotic Particles in High-Energy Physics with Deep Learning

P. Baldi,[1] P. Sadowski,[1] and D. Whiteson[2]

[1] *Dept. of Computer Science, UC Irvine, Irvine, CA 92617**

[2] *Dept. of Physics and Astronomy, UC Irvine, Irvine, CA 92617†*

Collisions at high-energy particle colliders are a traditionally fruitful source of exotic particle discoveries. Finding these rare particles requires solving difficult signal-versus-background classification problems, hence machine learning approaches are often used. Standard approaches have relied on 'shallow' machine learning models that have a limited capacity to learn complex non-linear functions of the inputs, and rely on a pain-staking search through manually constructed non-linear features. Progress on this problem has slowed, as a variety of techniques have shown equivalent performance. Recent advances in the field of deep learning make it possible to learn more complex functions and better discriminate between signal and background classes. Using benchmark datasets, we show that deep learning methods need no manually constructed inputs and yet improve the classification metric by as much as 8% over the best current approaches. This demonstrates that deep learning approaches can improve the power of collider searches for exotic particles.

AUC

Technique	Low-level	High-level	Complete
BDT	0.73 (0.01)	0.78 (0.01)	0.81 (0.01)
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 (< 0.001)	0.885 (0.002)

Discovery significance

Technique	Low-level	High-level	Complete
NN	2.5σ	3.1σ	3.7σ
DN	4.9σ	3.6σ	5.0σ

Important:
Input only 4 vectors !!!!
No knowledge about physics !!!!

Then ..hyped in QCD … jet algorithms.. … top taggers.. Showering and calorimeters

Machine Learning for Jet Physics

11 Dec 2017, 01:15 → 13 Dec 2017, 18:00 US/Pacific

2-100 (Lawrence Berkeley National Laboratory)

Benjamin Nachman , Kyle Cranmer , Matt Dolan , Timothy Cohen (Princeton/IAS)

Description There has been a recent surge of interest in developing and applying advanced machine learning techniques in HEP, and jet physics is a domain at the forefront of the excitement. The goal of this workshop is to gather experts and new-commers to discuss progress, new ideas, and common challenges. The workshop is open to the community; we invite contributions and will try to accommodate everyone within reason.

Slides

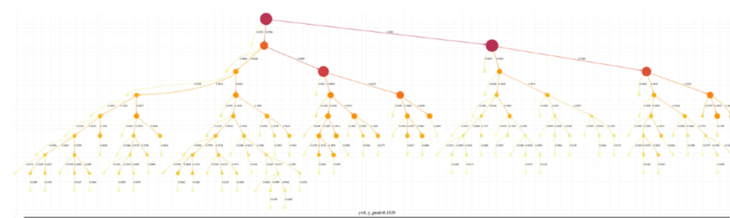
Participants Anders Andreassen, Andrew Larkoski, Aviv Cukierman, Benjamin Nachman, Bryan Ostdiek, Charilou Labitan, Christine McLean, Christopher Frye, Eric Metodiev, Felix Ringer, Francesco Rubbo, Frederic Dreyer

…CHINE LEARNING

…ays of injecting physics knowledge into …well…

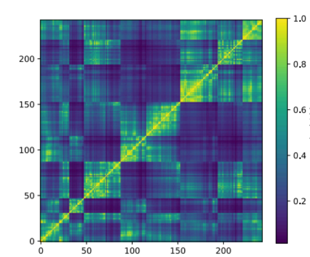
QCD-Aware recursive neural networks

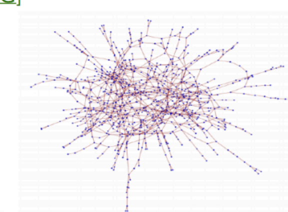
arXiv:1702.00748



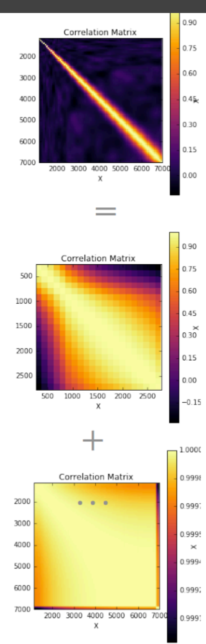
QCD-Aware graph convolutional neural networks

NIPS2017 workshop [http://bit.ly/2AkwYRG]





$$d^{\alpha}_{ii'} = \min(p_{ti}^{2\alpha}, p_{ti'}^{2\alpha})\frac{\Delta R^2_{ii'}}{R^2}$$



Final Kernel =

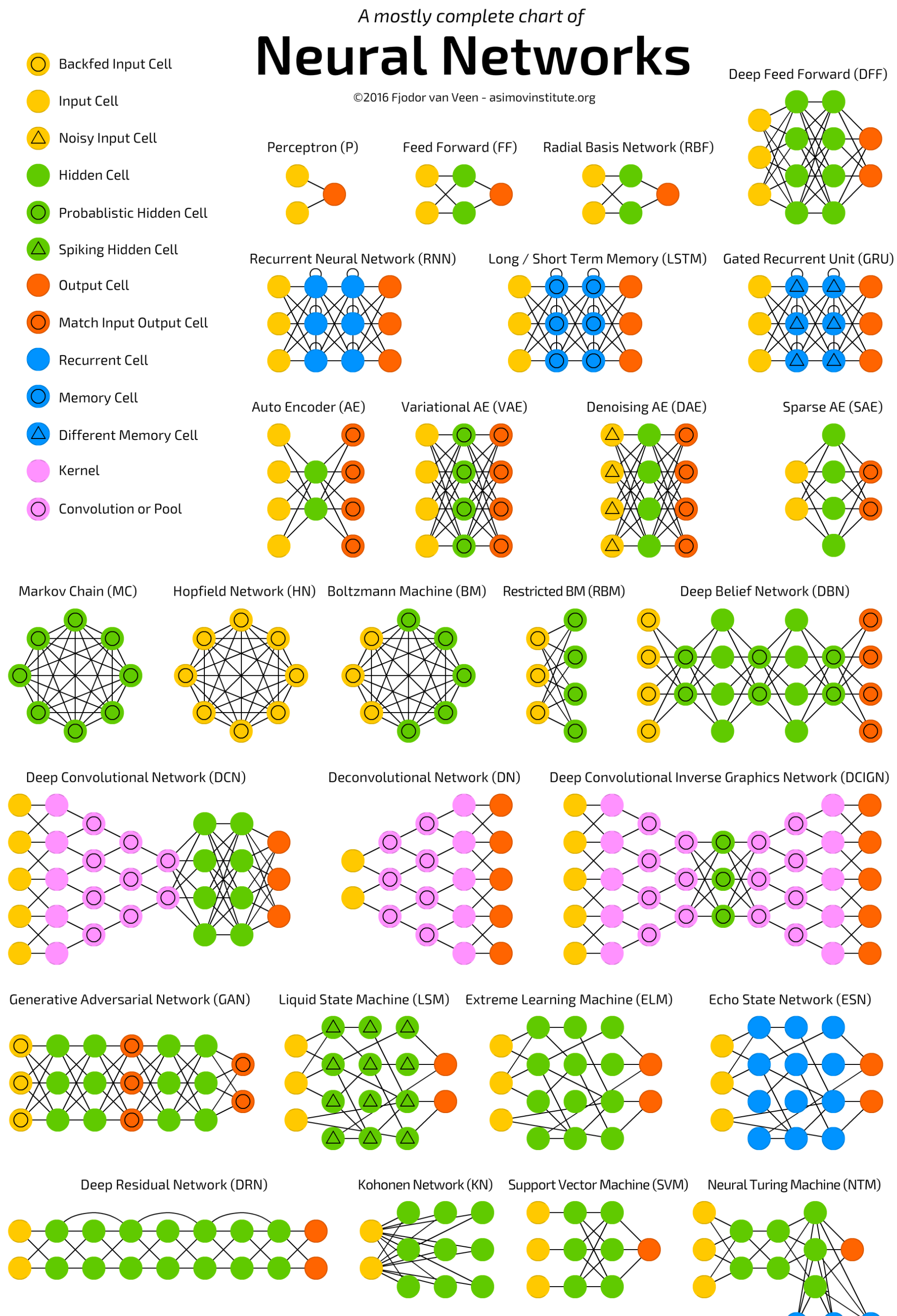
Poisson fluctuations

+ Mass Resolution

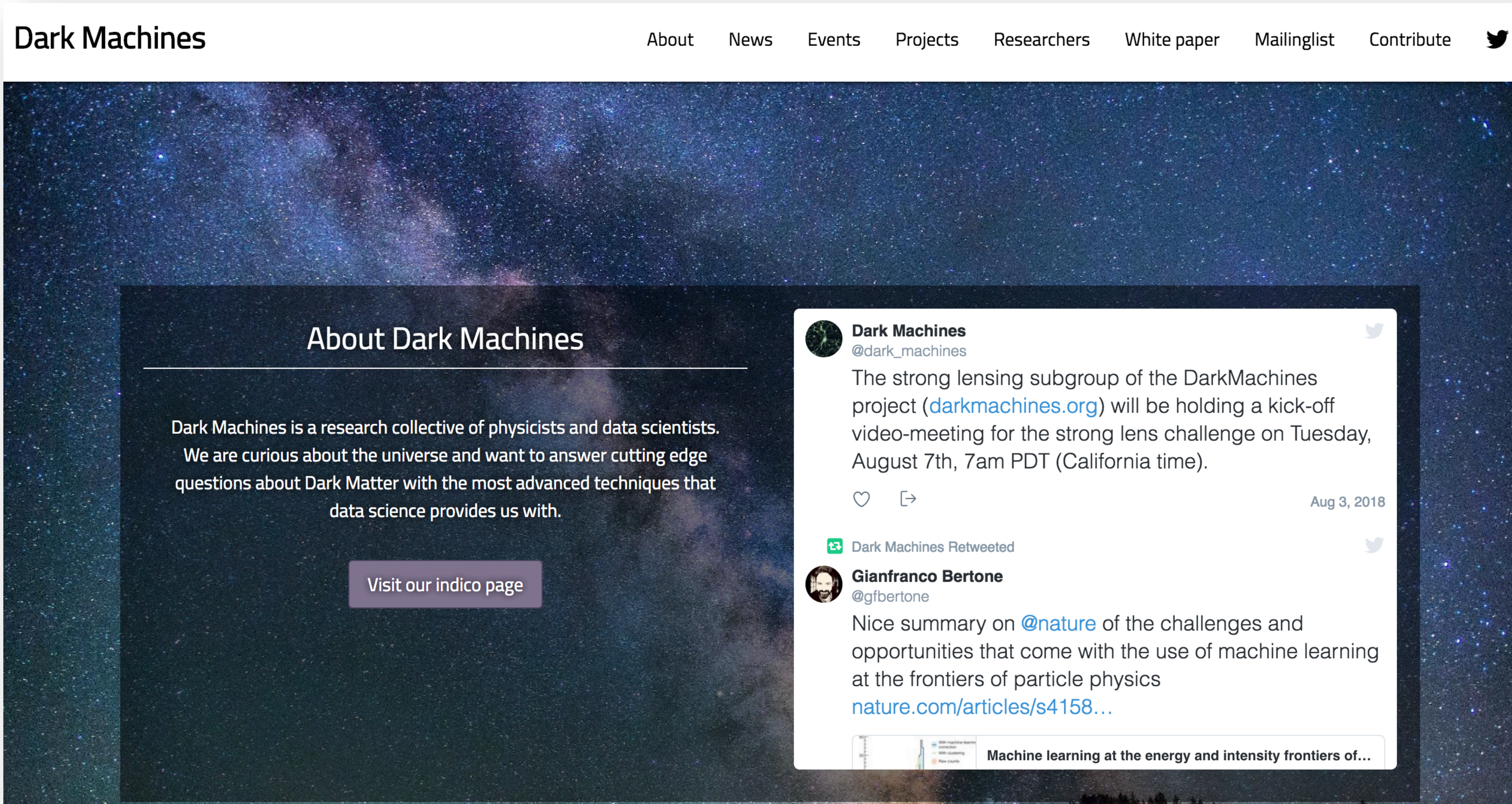
+ Parton Density Functions

+ Jet Energy Scale

Discussed in talk by Michael Kagan


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

2. Astroparticle DM searches with Machines


Dark Machines
About
News
Events
Projects
Researchers
White paper
Mailinglist
Contribute
About Dark Machines
Dark Machines is a research collective of physicists and data scientists. We are curious about the universe and want to answer cutting edge questions about Dark Matter with the most advanced techniques that data science provides us with.
Visit our indico page
Dark Machines
@dark_machines
The strong lensing subgroup of the DarkMachines project (darkmachines.org) will be holding a kick-off video-meeting for the strong lens challenge on Tuesday, August 7th, 7am PDT (California time).
Aug 3, 2018
Dark Machines Retweeted
Gianfranco Bertone
@gfbertone
Nice summary on @nature of the challenges and opportunities that come with the use of machine learning at the frontiers of particle physics nature.com/articles/s4158…
Machine learning at the energy and intensity frontiers of…

Deep Convolutional Networks

Actually Alpha-go used a deep convolutional network…
What is this ?

2015-2017: First deep learning attempts

We used deep convolutional networks to analyse gamma ray images for
Dark Matter

https://arxiv.org/abs/1708.06706

Convolutional Networks

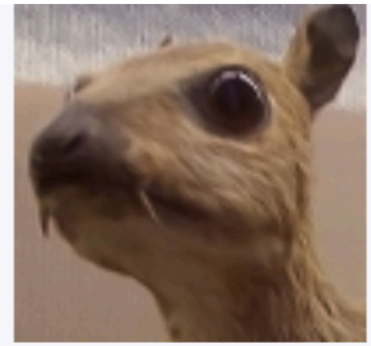
- Convolutional networks have convolution layers based on “filters”, a filter (a **matrix**) maps “a group of numbers” to “a number” reducing the data → CONV layers
- There are also layers which only do a downsampling (lower the dimensionality) POOL or “fully connected layers” to process the final numbers…

important paper:LeCun, Yann. "LeNet-5, convolutional neural networks". Retrieved 16 November 2013.

Filters (Matrix)

- Unity

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

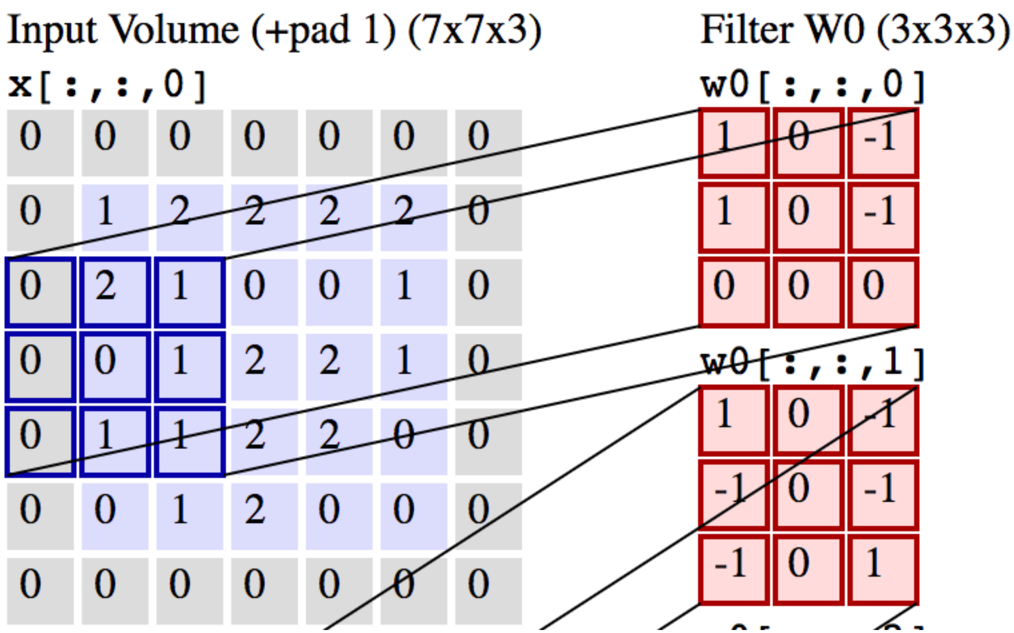


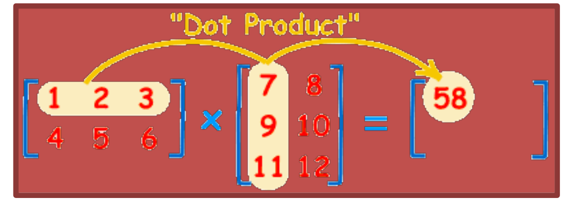
"Edge detector":

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

"Convolutional" Network use "invariances" (rotation, translation) in data (e.g. images)

CONV layer



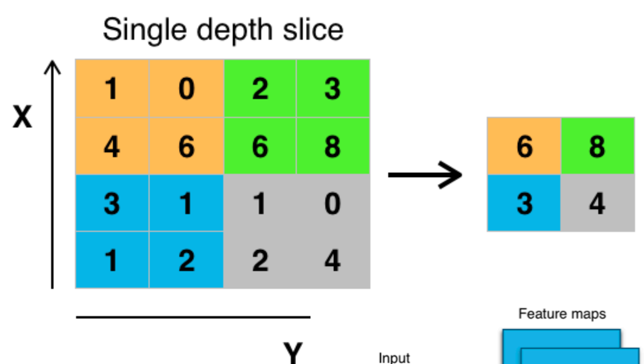


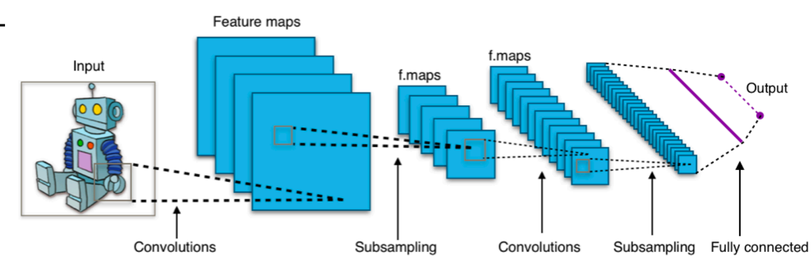
+ adding bias vector + applying e.g. RELU as non-linearity

Filter coefficients learned by backpropagation

Followed by a POOLING layer

(partition the input matrix into Submatrices, here max POOL, i.e. Storing the maximum)





(you can have Red, Green and Blue matrices)

Use case: gamma rays from galactic center

A method to investigate the origin of an excess emission of GeV γ rays in the direction of the Galactic Center
(reported by several groups by analyzing Fermi-LAT data)

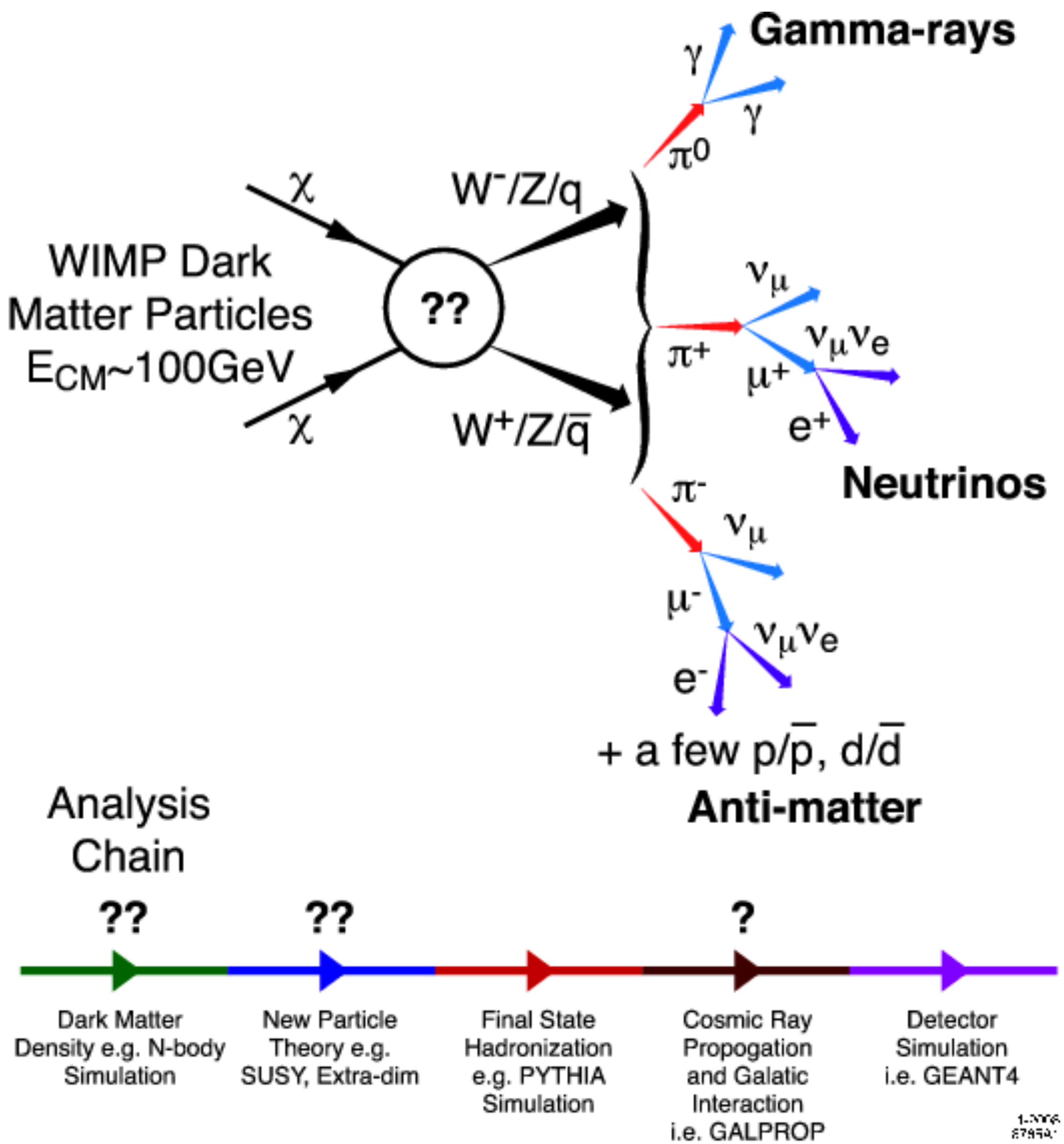
Interpretations of this excess include γ rays created **by the annihilation of dark matter particles** and γ rays originating from a **collection of unresolved point sources, such as millisecond pulsars.**

What have we done ?

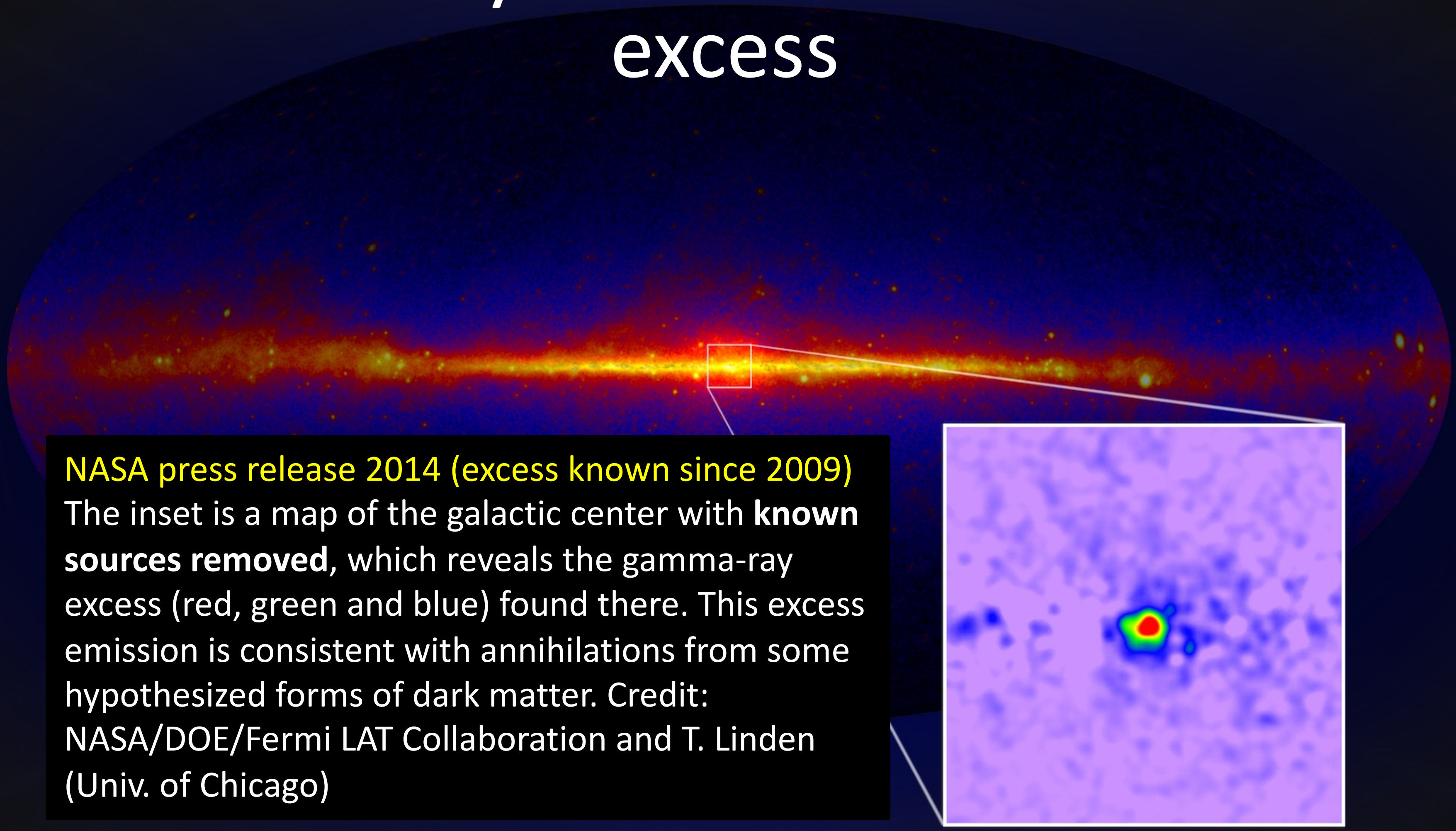
Simulated Fermi-LAT images based on point and diffuse emission models of the Galactic Center tuned to measured γ ray data

➔ Train and test convolutional Network on this

WIMP Astrophysics



Gamma rays & the Galactic Center excess



NASA press release 2014 (excess known since 2009)

The inset is a map of the galactic center with **known sources removed**, which reveals the gamma-ray excess (red, green and blue) found there. This excess emission is consistent with annihilations from some hypothesized forms of dark matter. Credit: NASA/DOE/Fermi LAT Collaboration and T. Linden (Univ. of Chicago)

Official paper in 2015

Fermi-LAT Observations of High-Energy Gamma-Ray Emission Toward the Galactic Center
Fermi-LAT Collaboration (M. Ajello (Clemson U.) *et al.*). Nov 9, 2015. 29 pp.
e-Print: **arXiv:1511.02938 [astro-ph.HE]** | PDF

Our convolutional network (convnet)

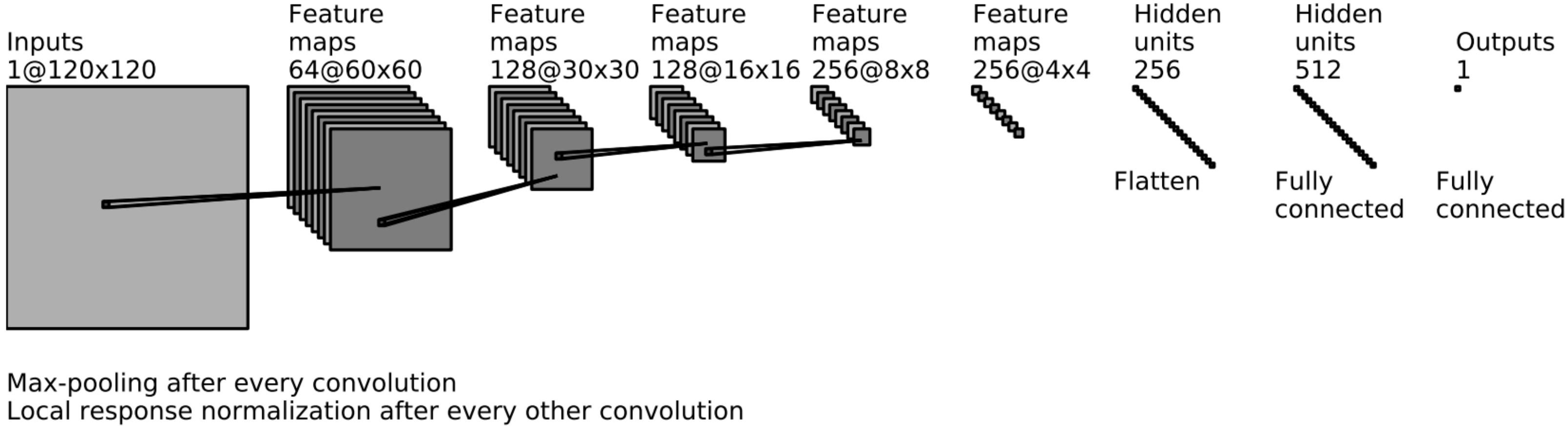


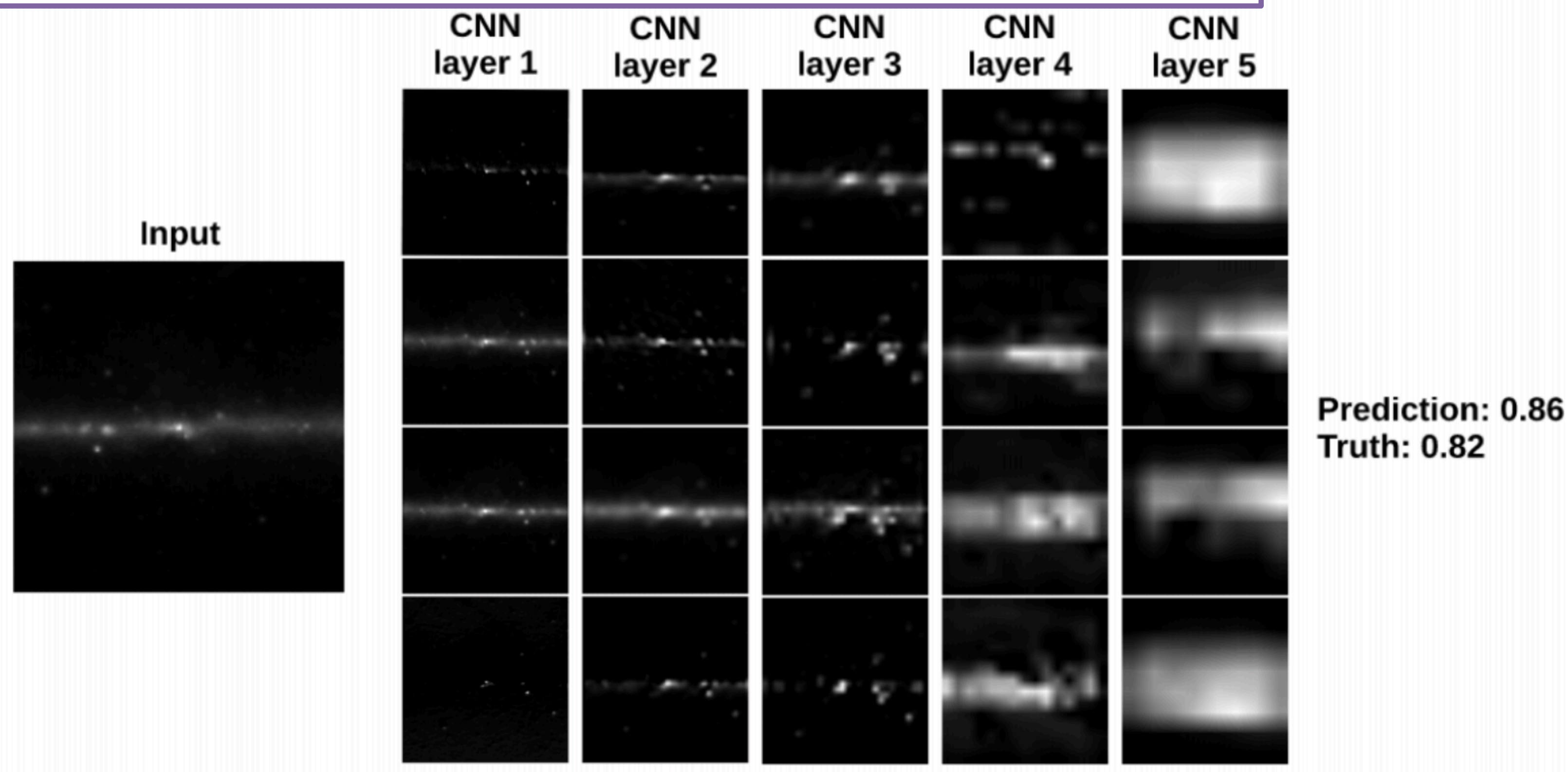
Figure 6: Visualization of the convolutional neural network. The network consists of an input layer, 5 convolutional + pooling layers, 2 fully connected layers and finally an output layer.

Hierarchical/compositional structure → smaller to larger structures
(reason: visible system is hierarchical as well…)

In comparison: GoogleLeNet has like 30 layers…

Isotropic or point sources: A Deep Convolutional Network approach

Output of the 5 convolutional layers can be “visualized” per event.



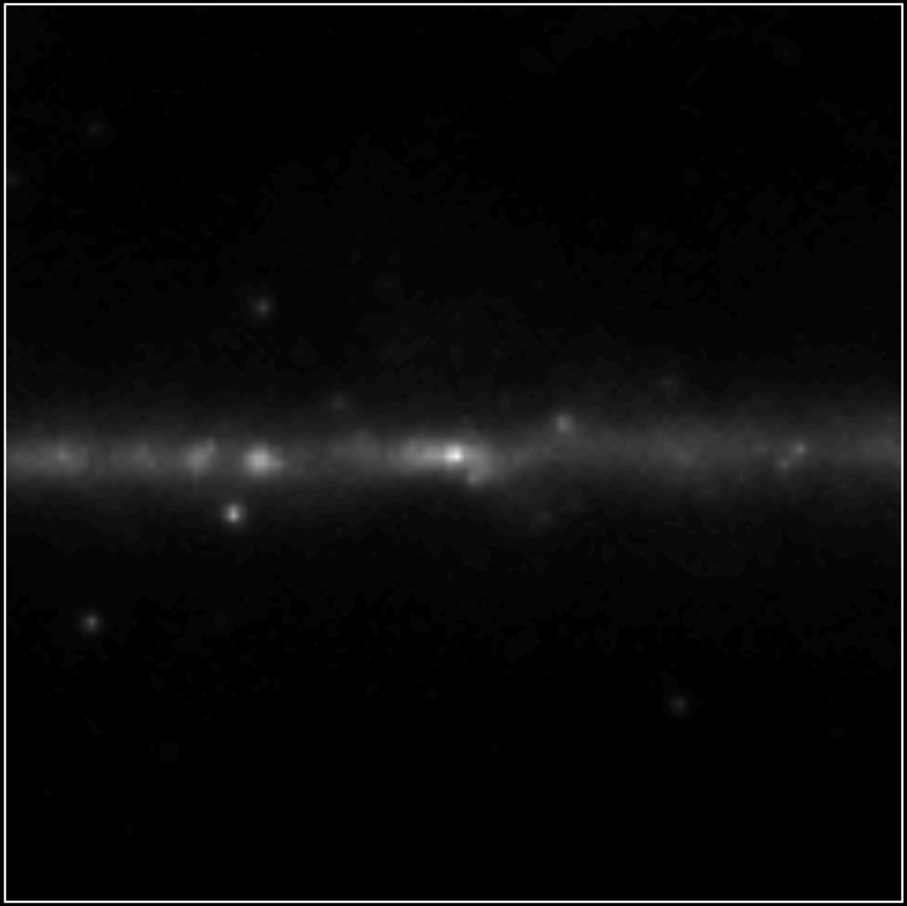
Activations of the network. Only four filters per layers are shown for clarity, between 256 and 65 filters are used for the different layers

arXiv:1708.06706

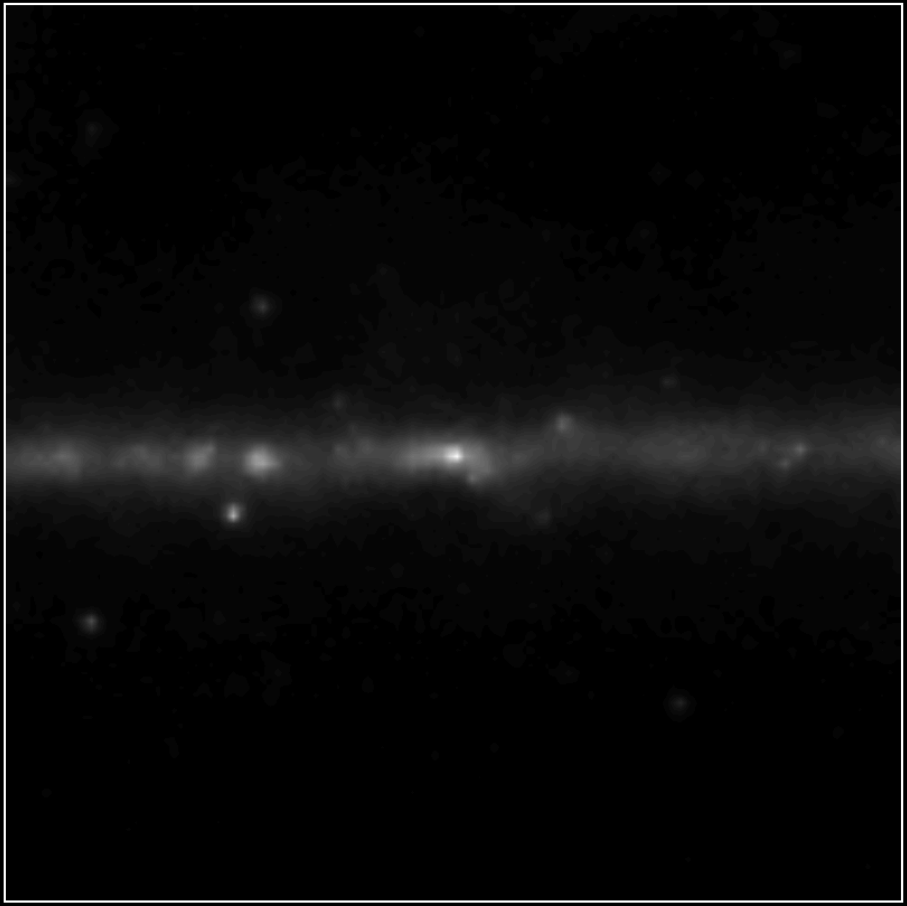
Guess the fraction of point sources

www.mydarkmachine.org

What is this fraction?



This is 0.5



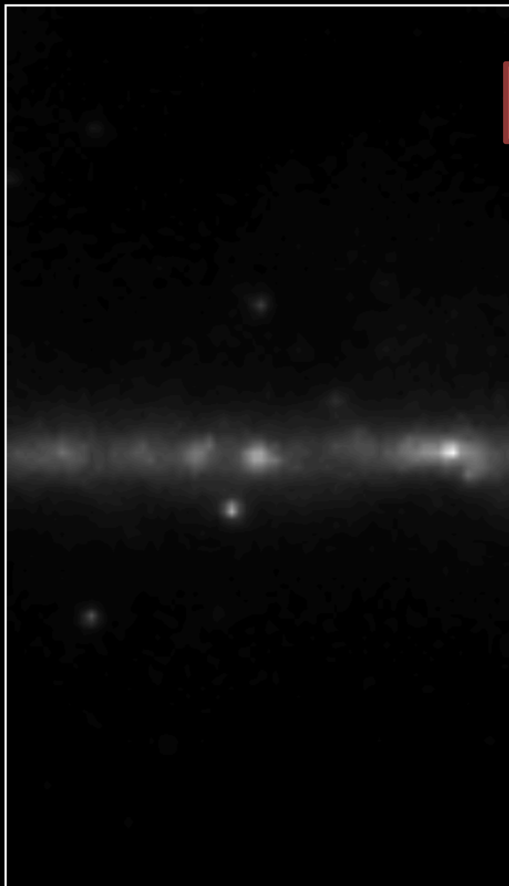
Your prediction:

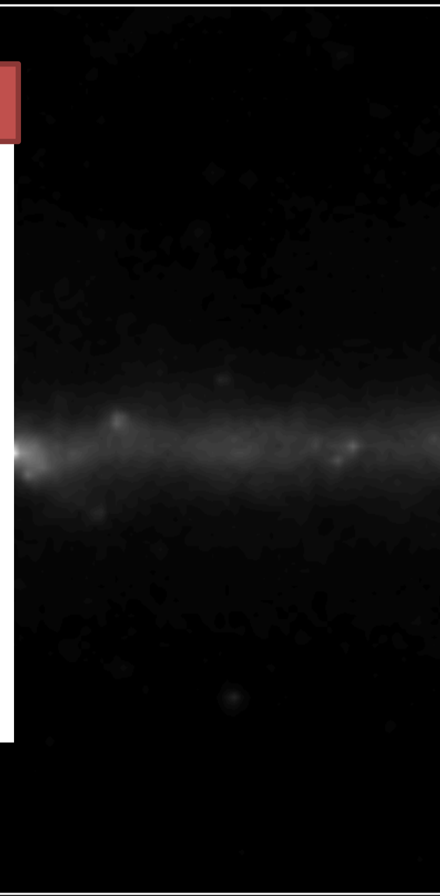
Invert image:

Guess

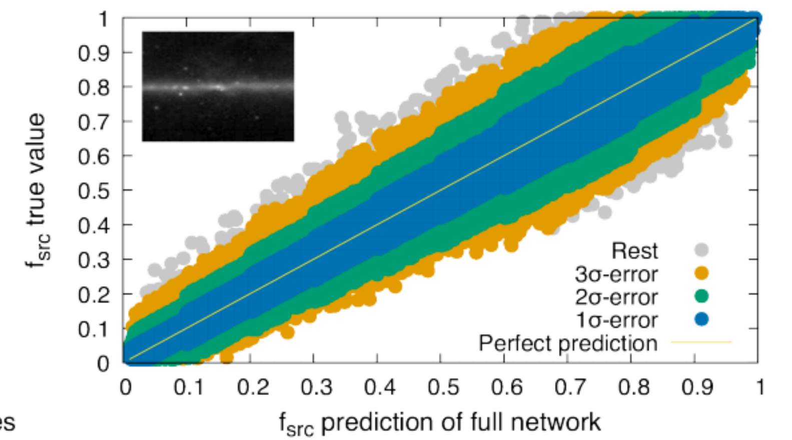
What is this fraction?

This is 0.5





Network can generalize over randomness



(b) Prediction of the full network versus true values.

Your prediction:

Invert image:

Truth:	0.052
Network:	0.1230
Your guess:	0.5

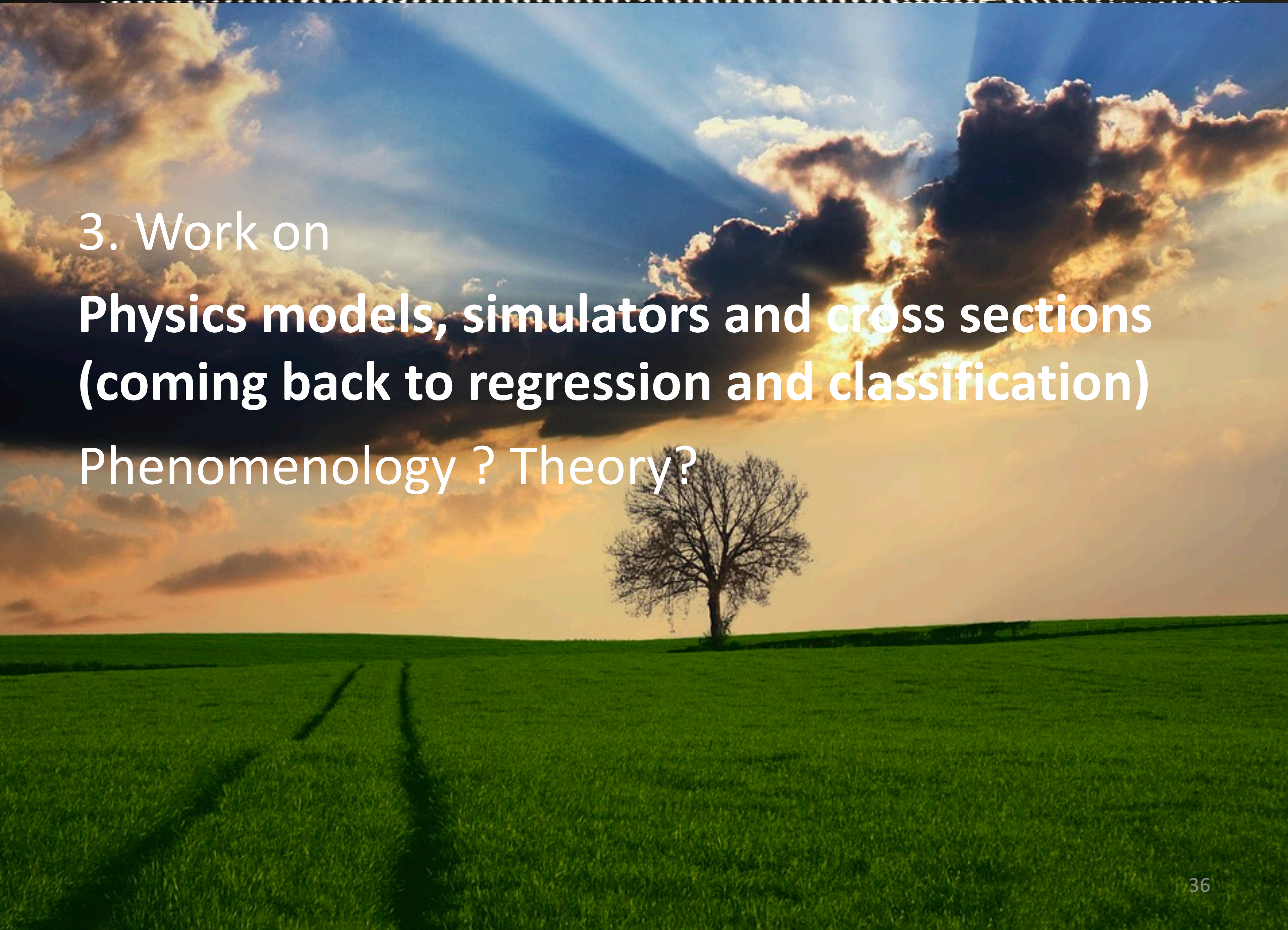
Who is better? The network

Interpretation here is frequentists and relies on the model to be correct (uncertainties from toy experiments, no p-value yet)

Next steps

- Categorize objects on the gamma-ray sky

Also point source detection now → see e.g. recent paper called "deepsource"
https://arxiv.org/abs/1807.02701

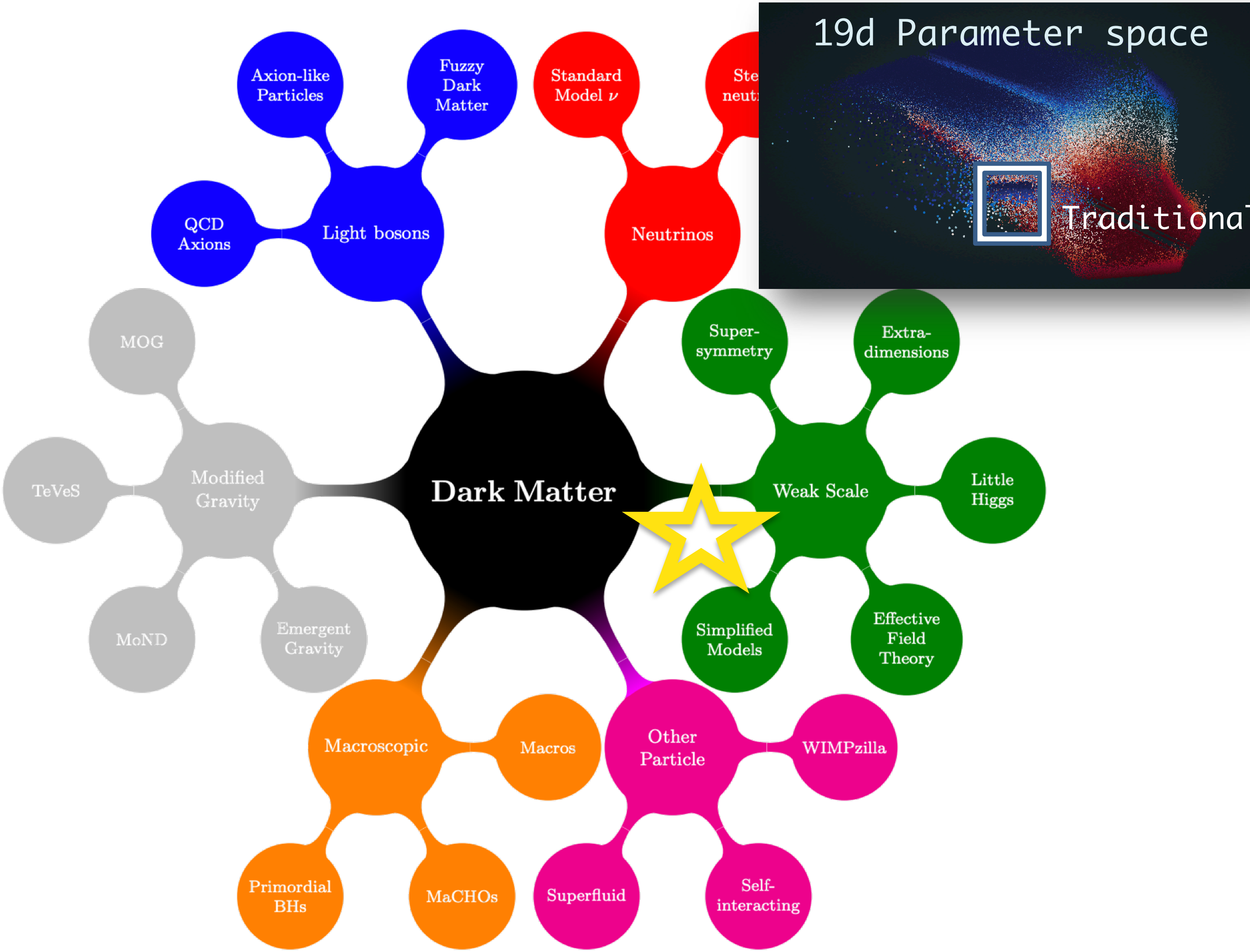


3. Work on

Physics models, simulators and cross sections (coming back to regression and classification)

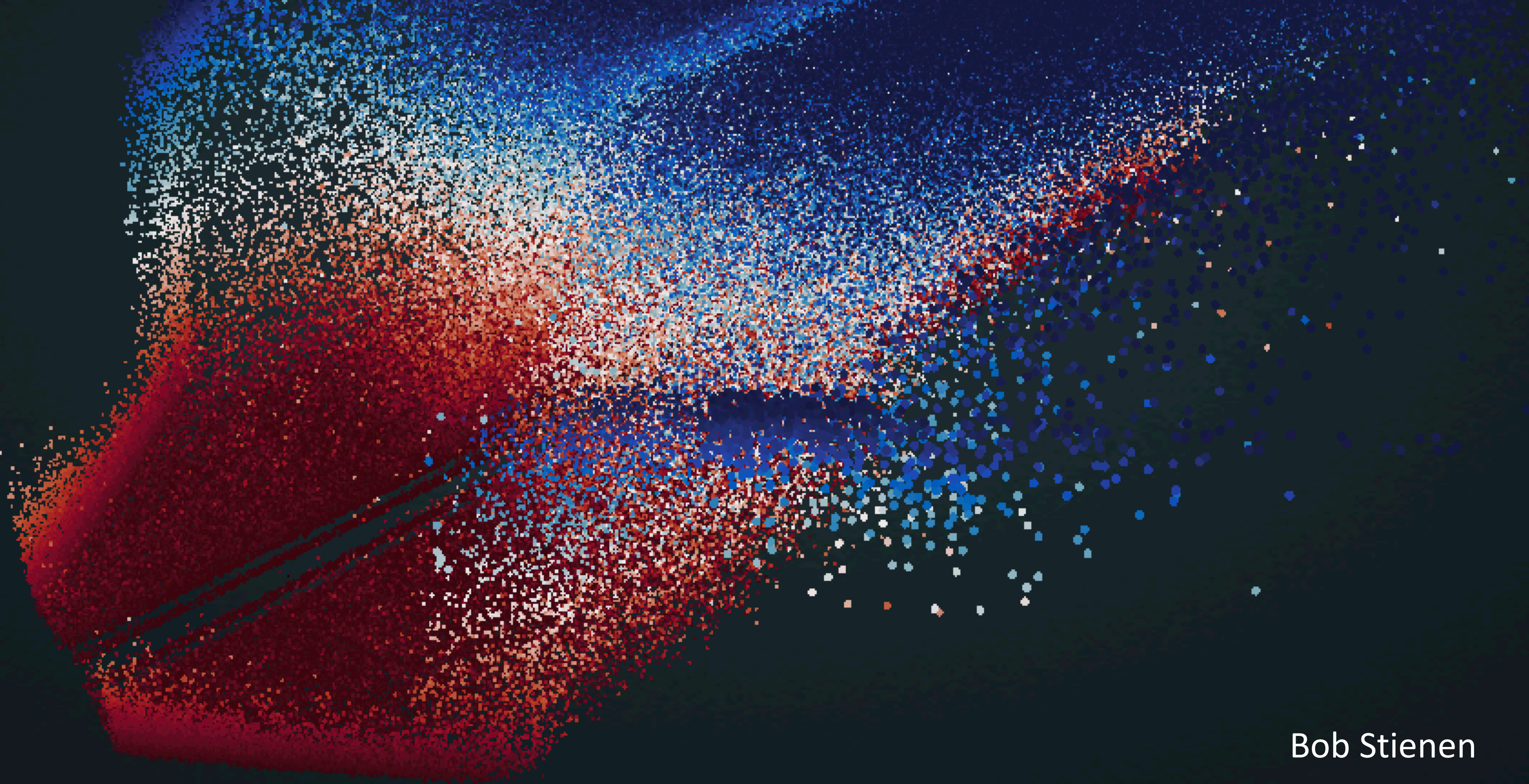
Phenomenology ? Theory?

What could it be? Dark Matter models



Models are in reality very very complicated
We humans simplify them

Ever seen such limit plots?
Can we broaden the search strategy ?
Can we also fine the model outside of the box ?



Bob Stienen

Accelerating searches

Inputs ➔ ***Long simulations + many programs*** ➔ Output

Train classification / regression tool to replace ***this*** by ML

Advantages:

- **Speed !**
- **Generality !**

Coupling Theory and Machine Learning part 1

"Learning a function" from datasets with known labels sounds boring and old-fashioned.

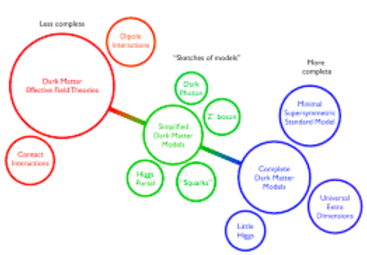
However we can couple it to simulators+ experiments + phenomenology ….

Coupling Theory and Machine Learning part 1

"Learning a function" from datasets with known labels sounds boring and old-fashioned.

However we can couple it to simulators+ experiments + phenomenology ….

Supersymmetry → Experimental Test → Machine Learning to generalize over Supersymmetry



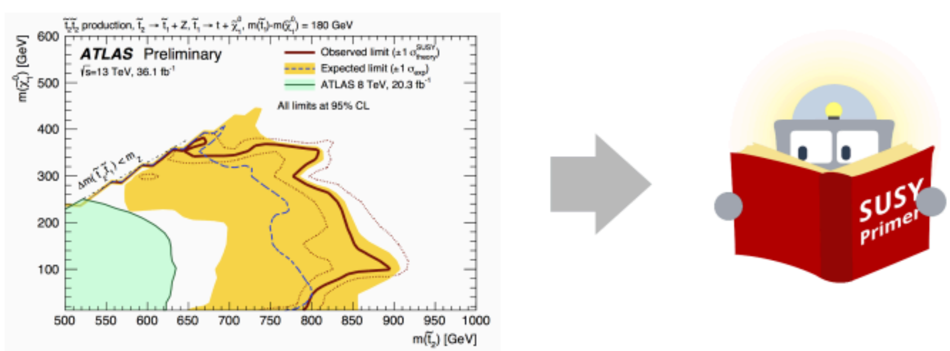
Coupling Theory and Machine Learning part 1

"Learning a function" from datasets with known labels sounds boring and old-fashioned.

However we can couple it to simulators+ experiments + phenomenology ….

SUSY-AI

- Exclusion determination in 19d pMSSM
- 310,324 model points with known exclusion as data input
- Algorithm: a collection of decision trees (Random Forest)
- **Idea: going from 2d slices to N-dim representations**



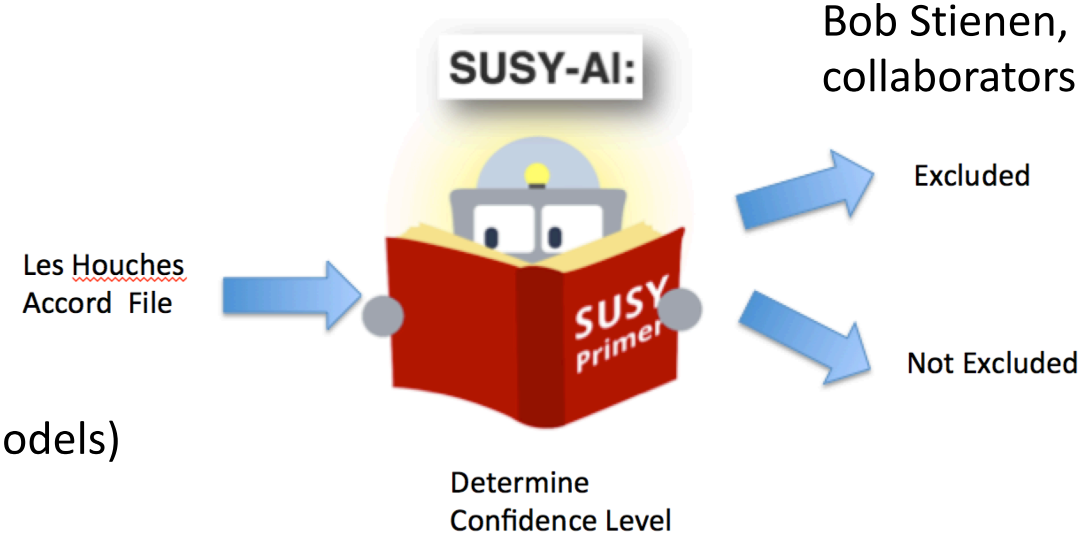
Prevent overfitting: *Boosting: many trees + not subset of all features for each tree*
*Bagging: random picking training data -> each tree of the forest sees only 0.68*data (see extra slides)*

SUSY-AI

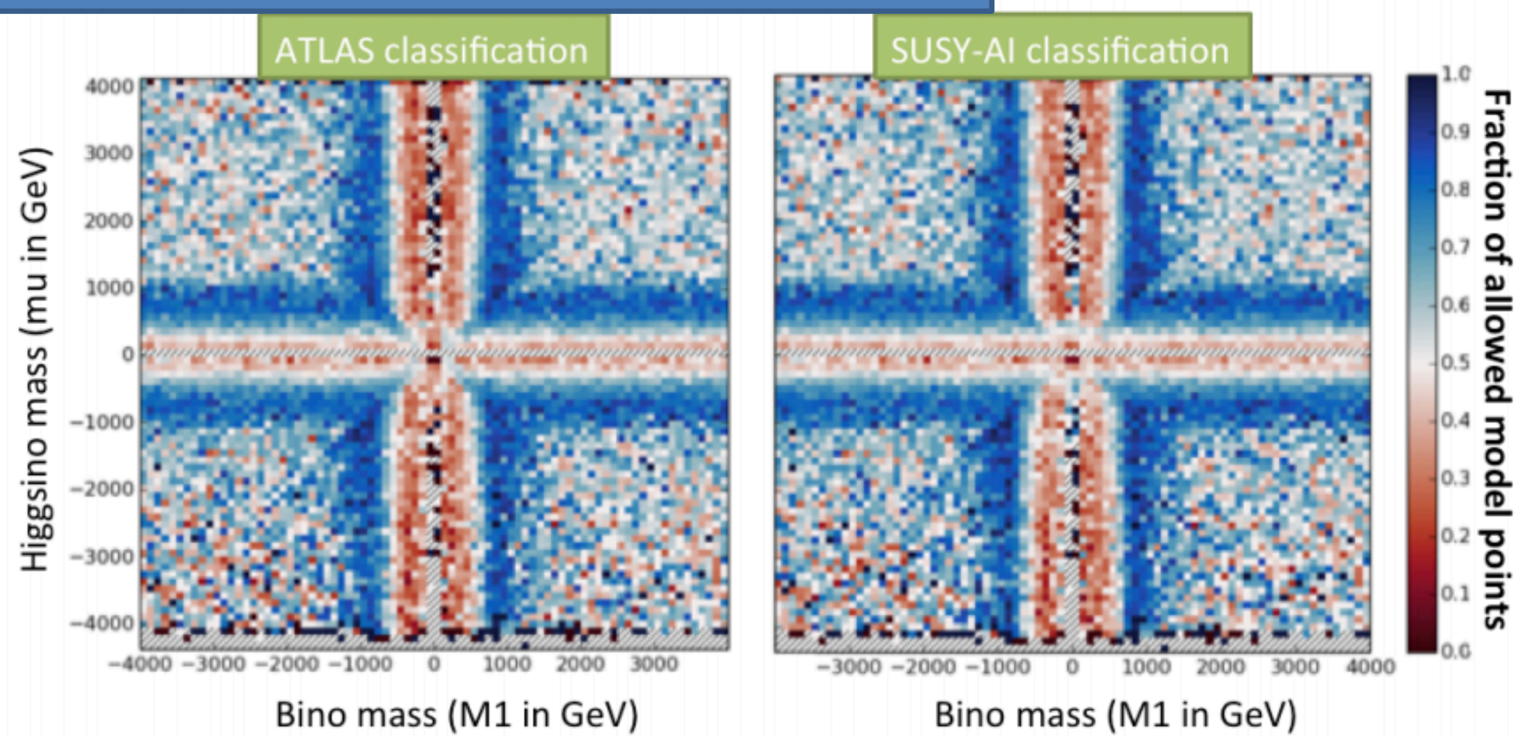
Bob Stienen, >10 collaborators

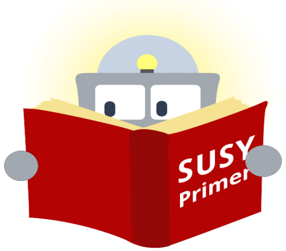
Encoding of model constraints with Machine Learning

Aim: Generic framework (**all** models)



Testing with out-of-bag estimation (remember 0.68!)



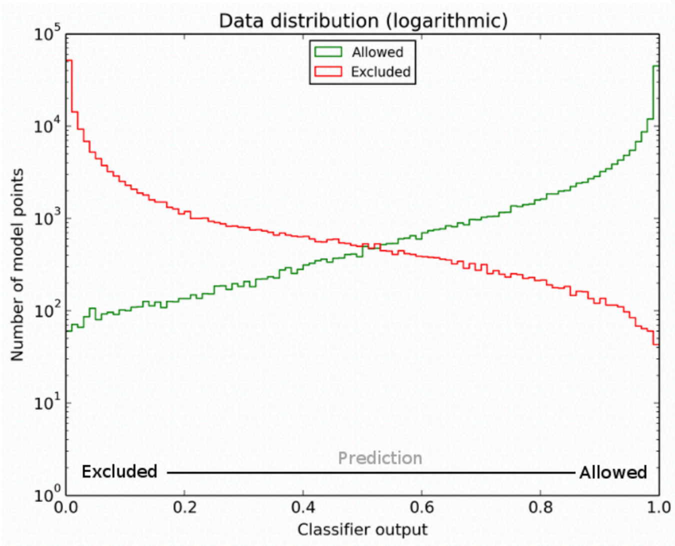


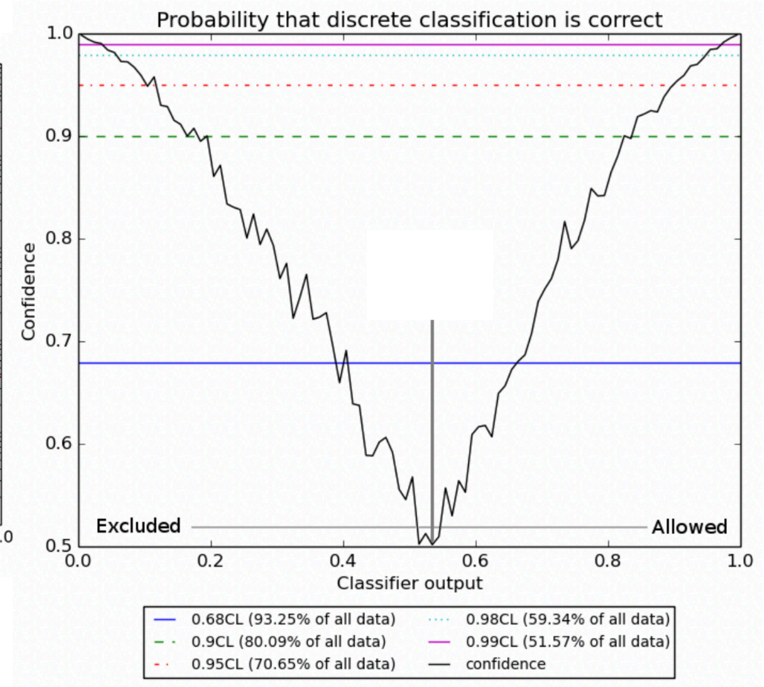
Used training data to learn classification

It determines a **confidence** level of its **classification** using the training data.

→ Need **more points** in regions of low certainty

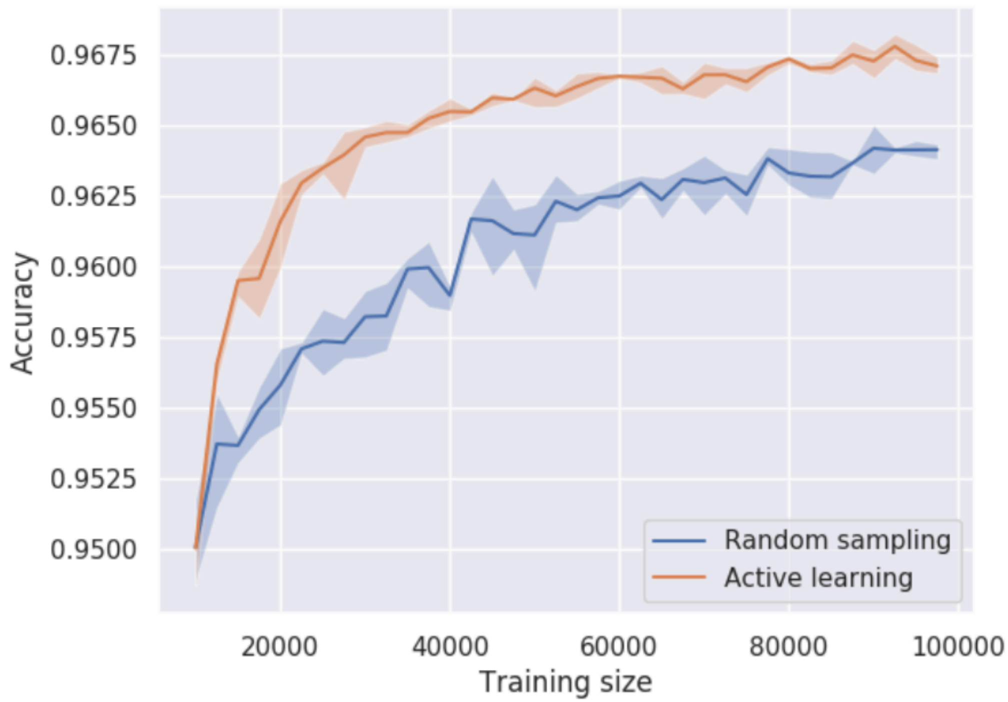
Ratio of majority class per bin





Active learning

arXiv:1905.08628 , mainly Bob Stienen



Query-by-Dropout-Committee

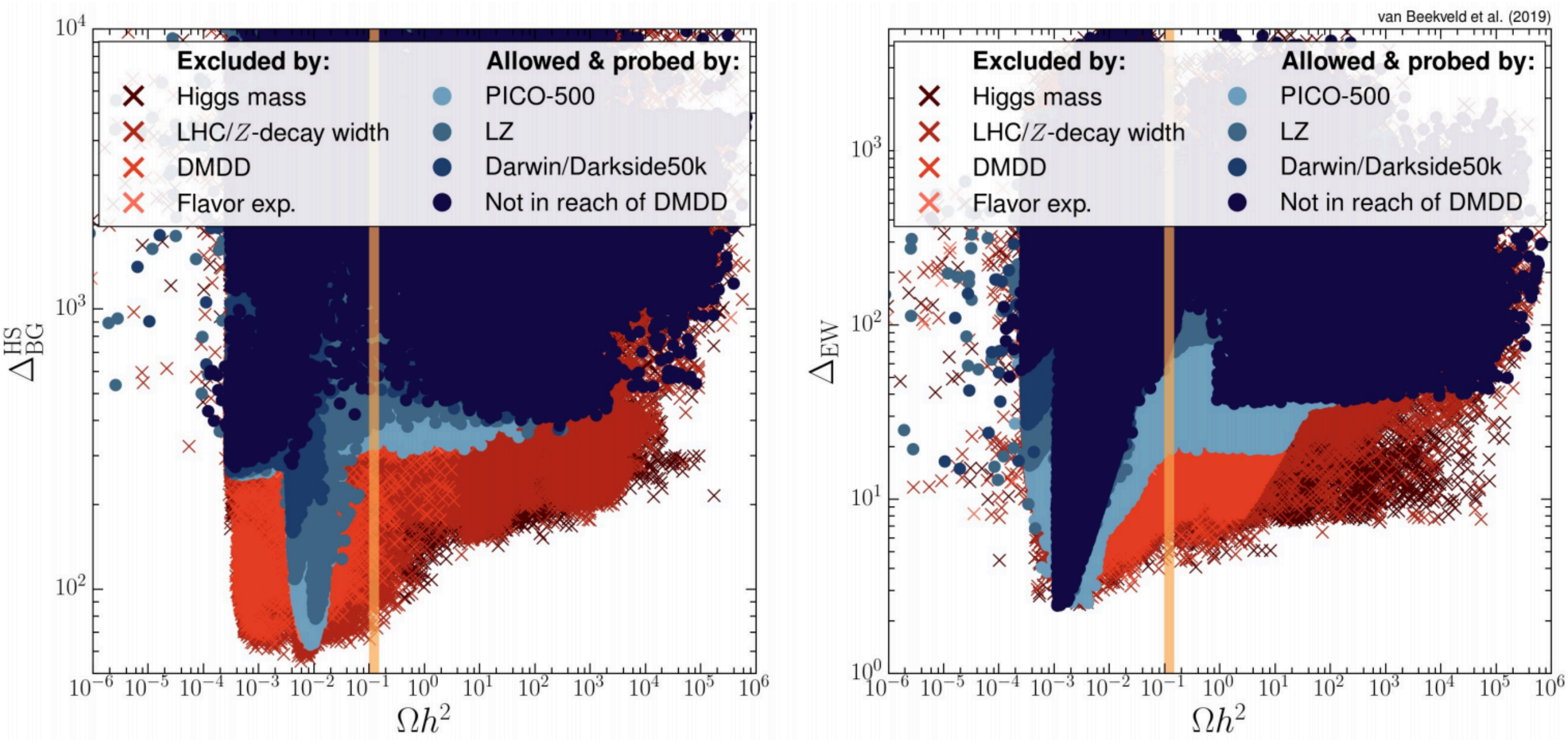
FIG. 5. Accuracy development on model exclusion of the 19-dimensional model for new physics (pMSSM) for random sampling and active learning using a dropout Neural Network with infinite pool. True labeling was provided by a machine learning algorithm trained on model points and labels provided by ATLAS [1]. The gain of active learning with respect to random sampling (as described by Equation 2) is 3 to 4. The bands show the range in which all curves of that colour lay when the experiment was repeated 7 times.

Why are you still interested in SUSY ?

Because …

Use cases → How fine-tuned is Supersymmetry in the 19d MSSM ?

https://arxiv.org/pdf/1906.10706.pdf



Les Houches project:
SUSY-AI --> PhenoAI

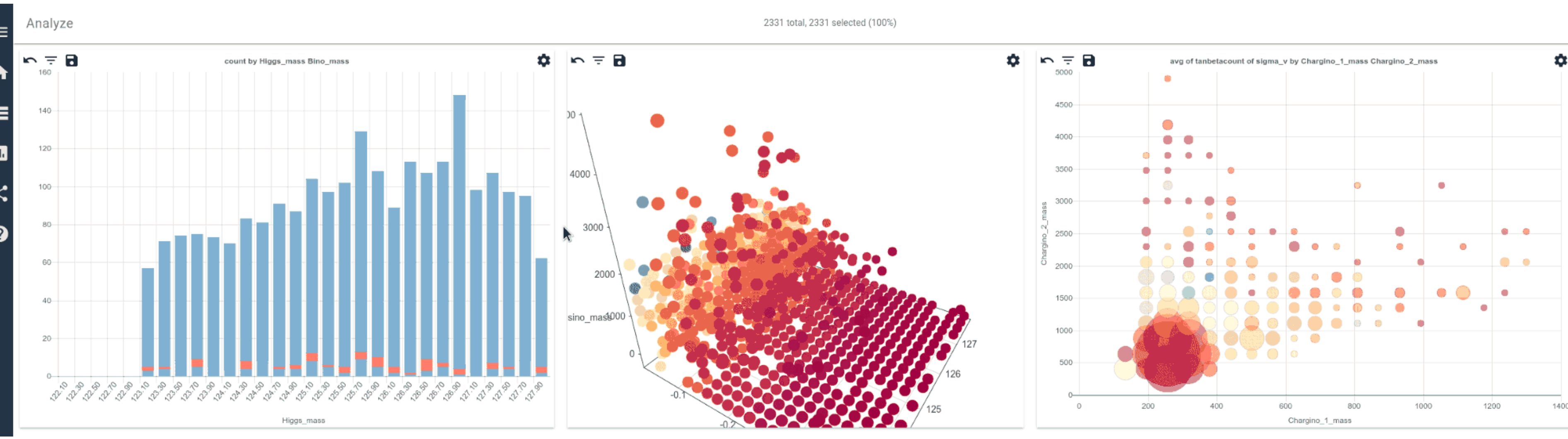
- Encoding model constraints for everybody in the full model parameter space (e.g. LHC constraints on various high-dimensional models)

- No simplifications of models needed !
- ”model-space” can provide training data

Need to store pheno data

- Remember the animated newspapers from Harry Potter ?

Pheno model database ?



Within
idark project
Dutch escience
Center

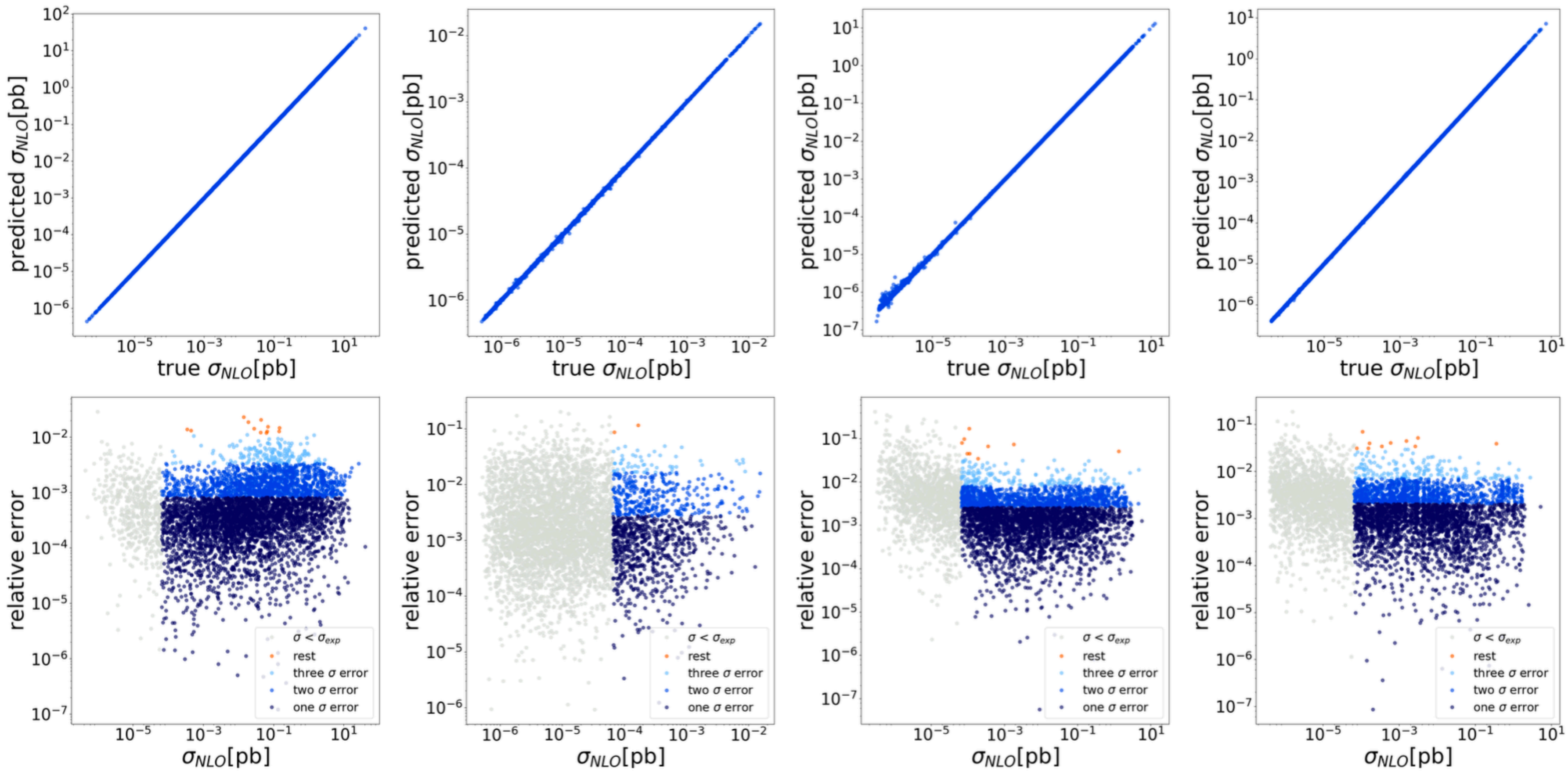
Faruk Diblen
Jisk Attema

Collect model solutions in a database
Use them as target !
Use Machine Learning to interpolate between them → Generalization of DM searches

www.idarksurvey.com

DeepXs: DM Cross sections

- Running NLO code to derive SUSY cross sections can take up to 10 minutes
- Can we “learn the cross sections” and derive in a microsecond for *any* model parameter set? ➔ https://arxiv.org/abs/1810.08312



a) $\tilde{\chi}_1^+ \tilde{\chi}_1^-$ b) $\tilde{\chi}_2^0 \tilde{\chi}_2^0$ c) $\tilde{\chi}_2^0 \tilde{\chi}_1^+$ d) $\tilde{\chi}_2^0 \tilde{\chi}_1^-$

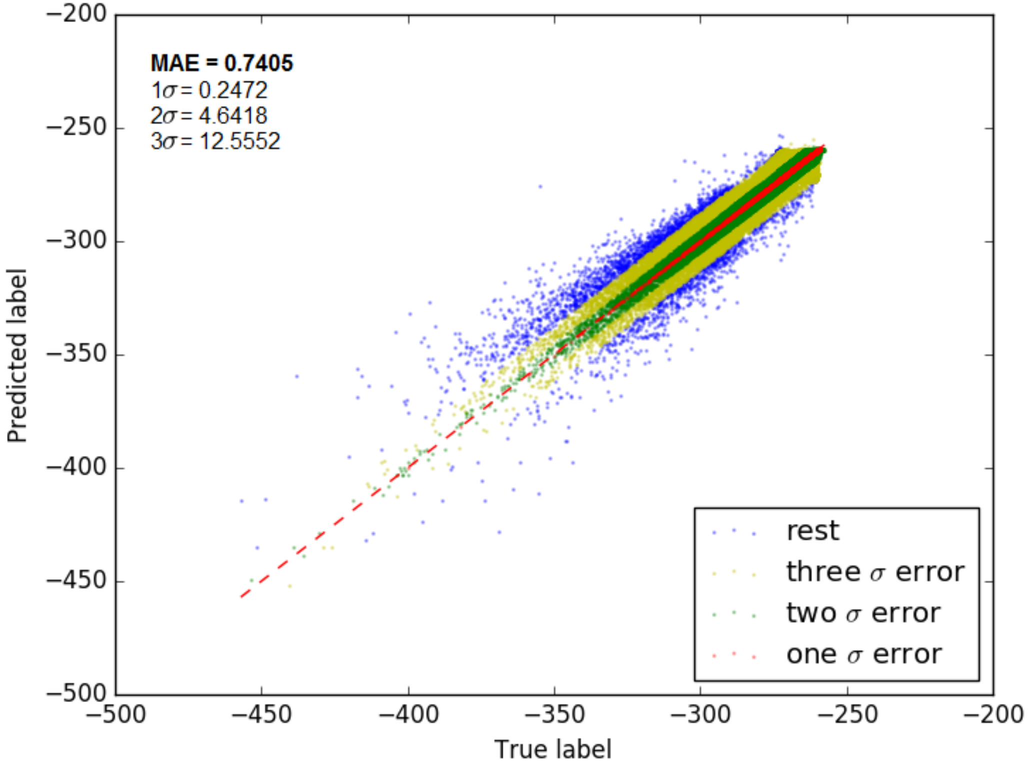
FIG. 1: The true vs. predicted NLO cross-sections (top) and the relative error vs. true NLO cross-section with confidence intervals (bottom) for the same 10^4 samples in both plots

inference at NLO with inference times that improve the Monte Carlo integration procedures that have been available so far by a factor of ≈ 6.9 million from ≈ 3 minutes to $\approx 26\mu s$ per evaluation.

Regression: Likelihoods

- Fitting groups derive likelihood plots for given models
- Can we “learn the likelihoods” in return in a microsecond for *any* model parameter set ?

BSM-AI regression example… Learning GAMBIT likelihoods



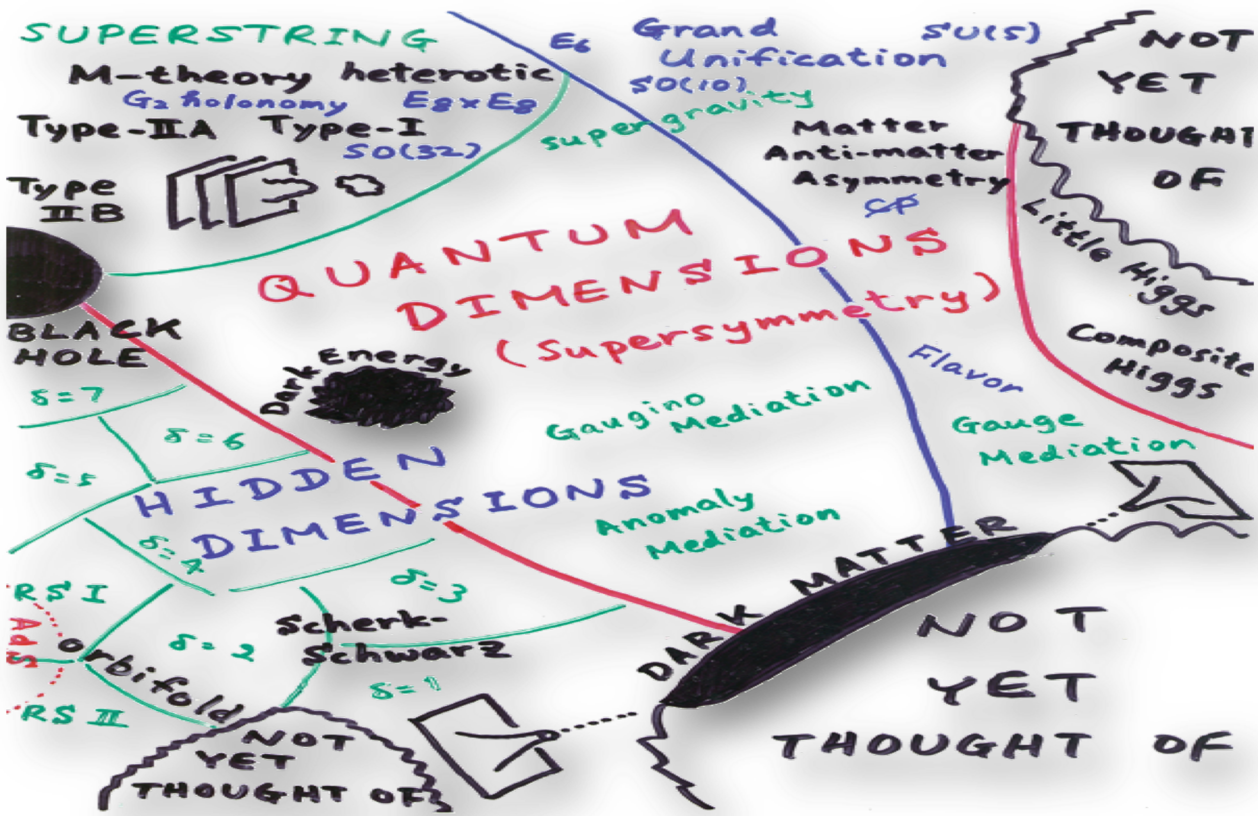
MSSM - 7

https://arxiv.org/abs/1705.07917

Plot by Sydney Otten

5. (Unknown) data anomalies

Model parameter space (pre- LHC)



Unknown signals/ unknown labels

Typical task at the LHC is **supervised discrimination of signal and background** (particle ID, Higgs search)

→Discriminator

(typically BDT/TMVA, now Deep Network)

Interesting:

What can we do if the signal is unknown ?

Related to a simpler question:

What can we do if the signal is vaguely known (i.e. a simulation is possible) ?

Today: I have a new toy for you, I put it somewhere in your room. The size is 0.1-100 cm. Can you find it ?

We propose to construct an automatized „scan“ of the full room/data for something interesting…

→This may help LHC to find new physics

→We can embed into this “scan“ our prejudice how new physics could look like, e.g. in the example on the previous slide this could be done via a „toy“ detection software trained on all known toys with the ability to do some extrapolation and interpolation …

Many hypotheses …

Searching for new physics with ‚minimal/less‘ assumptions on the signal

Consequences:

Less signal assumptions → more hypothesis tests (multiple testing)
→ more/all channels and data selections

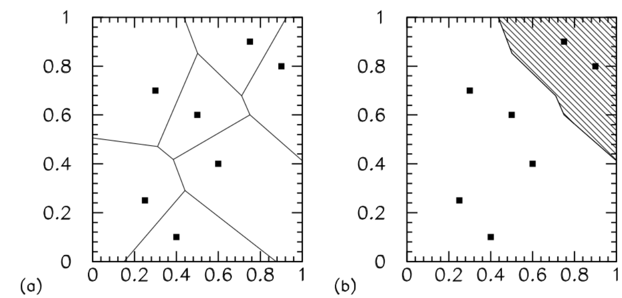
Implementations:

- Search with an "algorithm|: **automatizing data selections and testing**
- **Automatize/Generalize** the construction of **the background model**

Automatized generic searches in HEP: history

- Before **2000**: Some (unpublished) ideas and work (e.g. by Thomas Hebbeker and M.W. Krasny) to construct generic (non-model dependent searches) by comparing data with background expectation in a broad class of high pt events
- **2000**: First „automatized search without fixed model assumptions“ in HEP
  → **Sherlock/Sleuth at D0** experiment/ Tevatron (B. Knuteson and others)

Partition of events into (many !) Voronoi regions defined by data (N=1 region, N=2 regions, etc.)



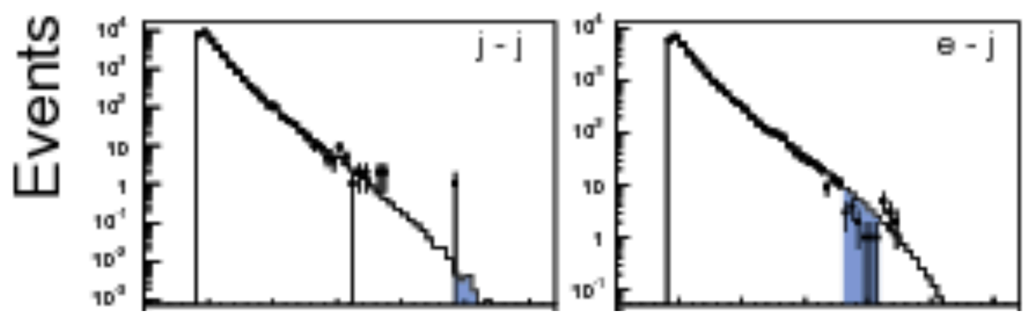
Criteria which regions are interested (e.g. corners, high pt…)
and should be considered for hypothesis tests (p-value to test if data is consistent with SM expectation)

→ Select region with smallest p-value
→ ***Problem: (Too) many trials + Needs multivariate understanding of backgrounds***

https://arxiv.org/pdf/hep-ex/0006011.pdf

Automatized generic searches in HEP: history

- 2004: **General Search at H1** experiment at HERA. (*2nd paper 2009)*
  All "HERA channels" 1-d search algorithm with smaller trial factor
  -> similar/better sensitivity
  - simpler to understand the background



- 2007: **Global Searches at CDF** with 1-d algorithms (one algorithm became "bumphunter" in 2011)
- 2010-2016: Start of work for LHC (several internal notes in ATLAS, one in CMS)
- 2011: First public **ATLAS** CONF note (4.7 fb^{-1} with 7 TeV)
- 2011: **CMS** PAS note ("Music") (MC note in 2008)
- 2011: Second ATLAS CONF note (20 fb^{-1} with 8 TeV)
- 2016-2018: **ATLAS released paper to arxiv (submitted to EPJ-C)** , 13 TeV data, 2015 data)

A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment

Goal:

Strategy paper. Generalize previous attempts.
Define a “meta-algorithm“ for
automated / generic / unsupervisedLHC searches
Show with 2015 data that this is - in principle – possible

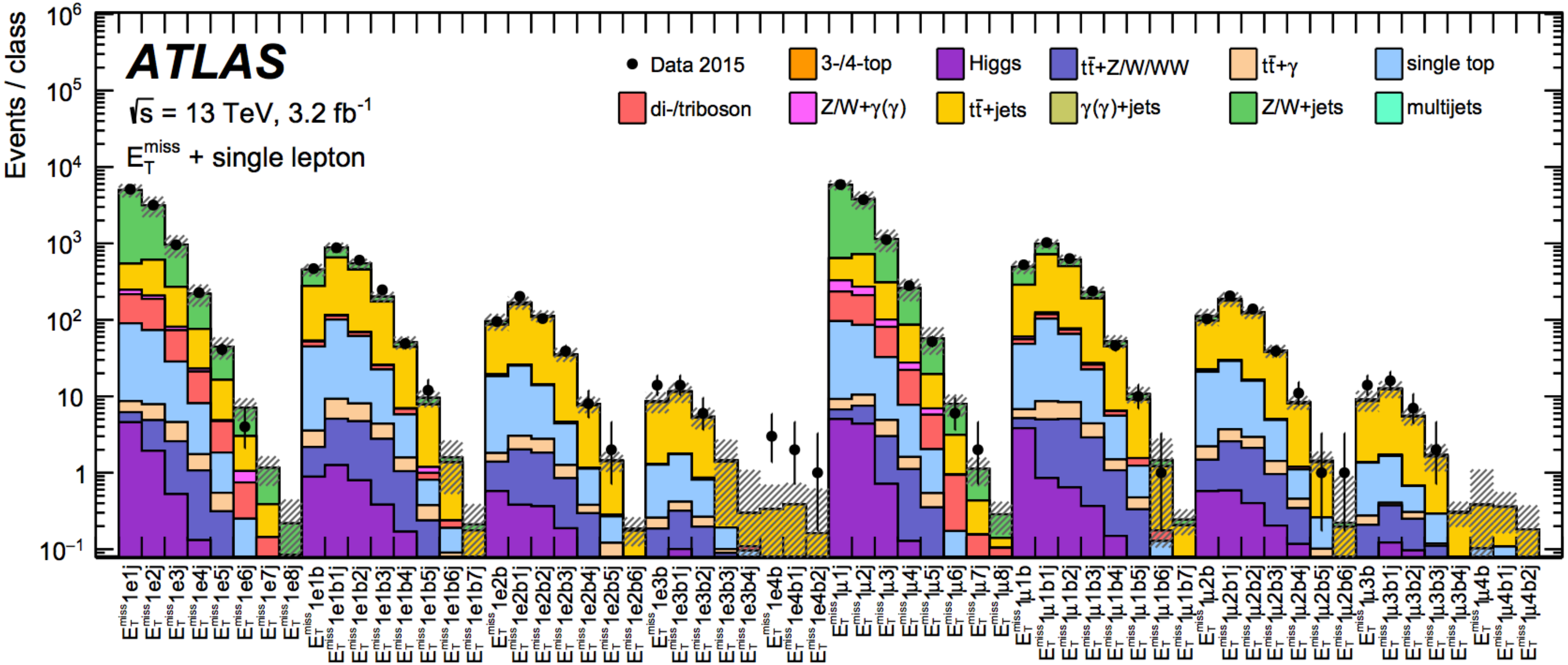
https://arxiv.org/pdf/1807.07447.pdf

A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment

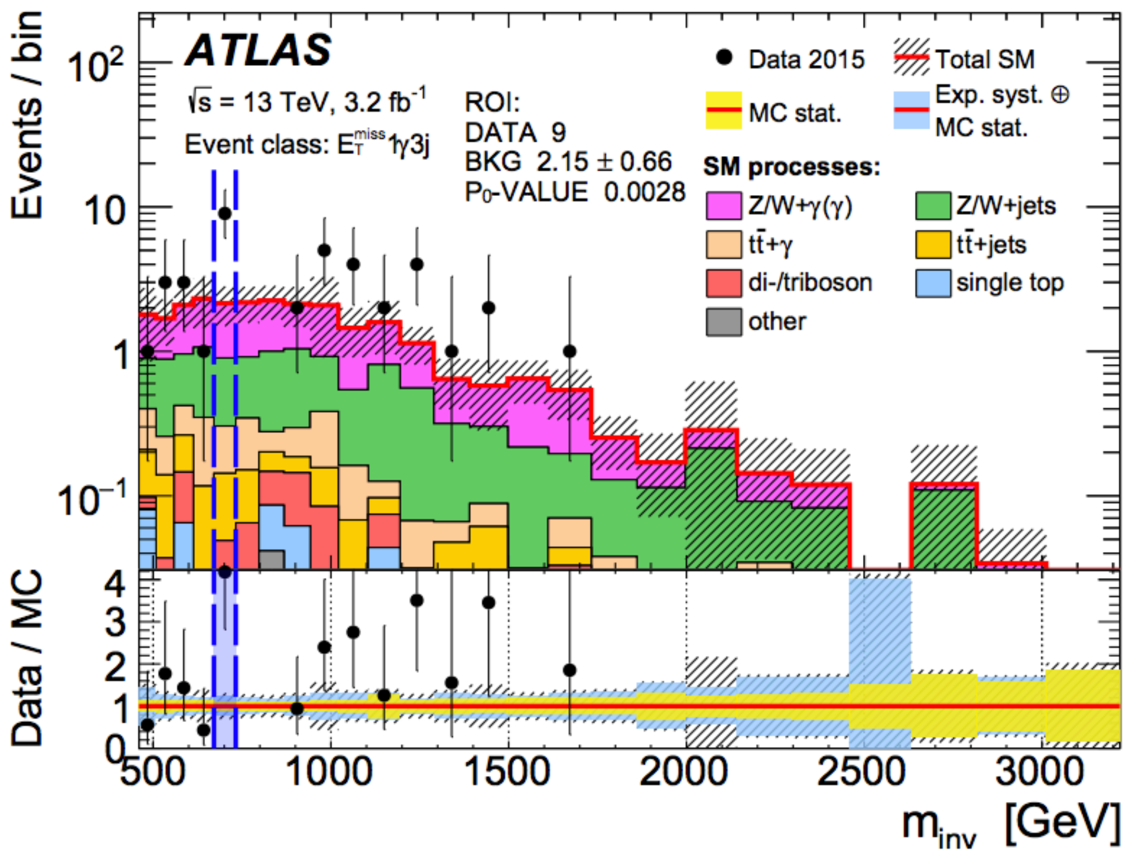
Define a 2-step approach:
First put available resources on generality
Then use available resources to test most interesting deviations…

1. General Search: Automatically testing a large set of signal regions
Observation of one or more significant deviations in some phase-space region(s)
➔ Trigger to perform dedicated and model-dependent analyses
where these **'data-derived' phase-space region(s) can be used as signal regions**

In ATLAS > 800 channels !
> 10^5 regions !



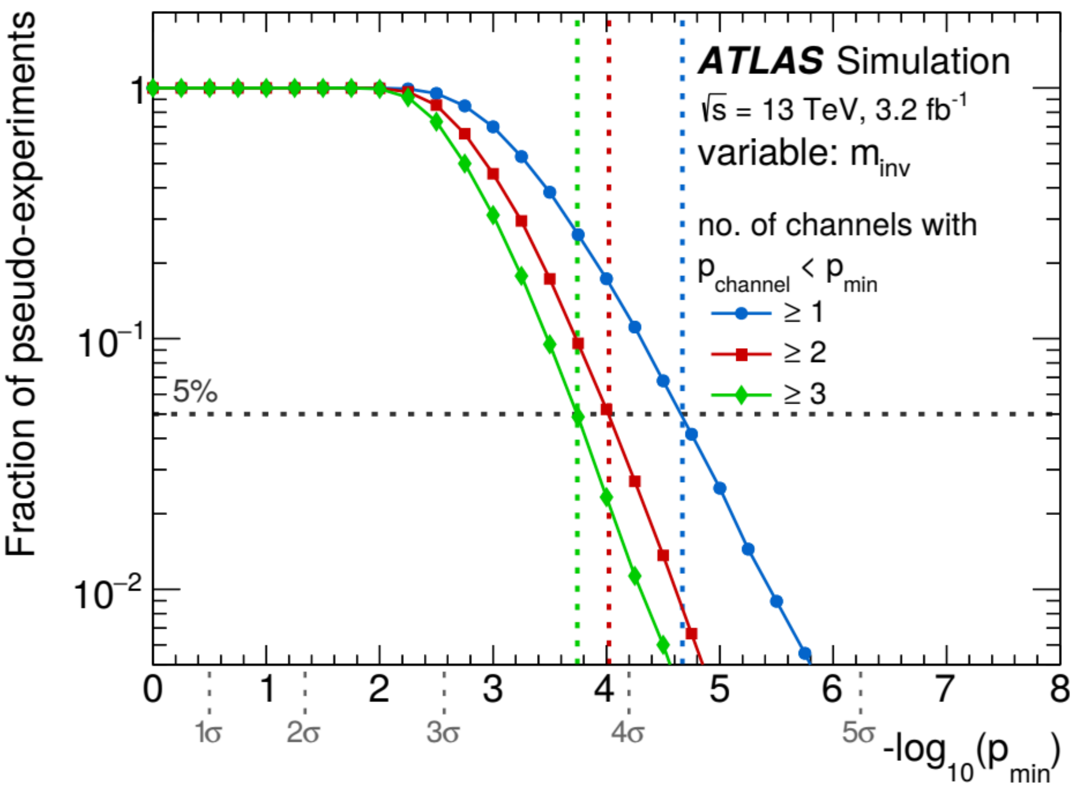
> 800 channels …. (plot shows a small selection)



> 30000 regions (hypothesis tests)

Determine p-value thresholds by asking how many toy datasets would give such a deviation

→ A regions is interesting if you find channels with p-values more significant than in 95% of the toys



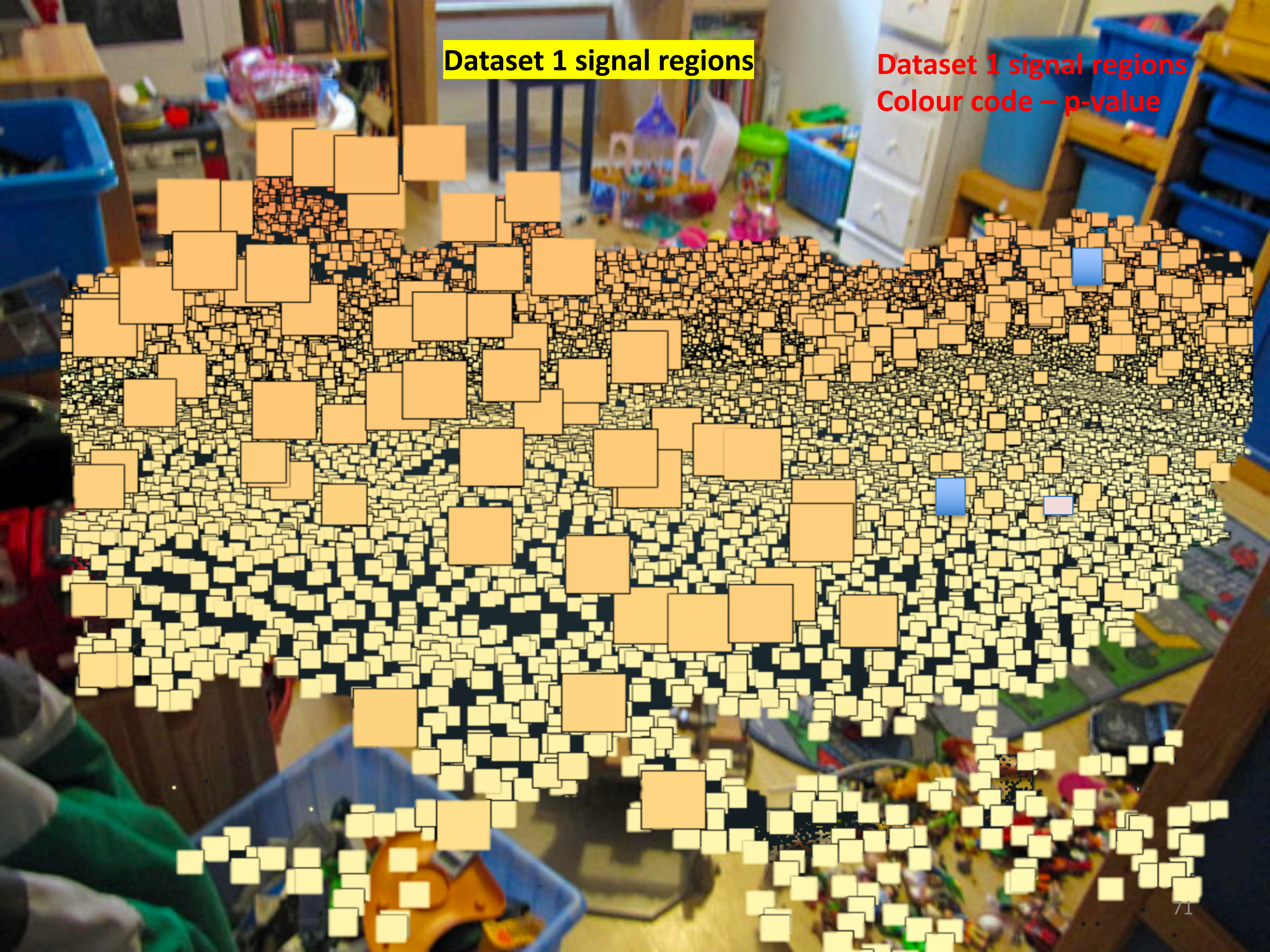
A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment

Define a 2-step approach:
First put available resources on generality
Then use available resources to test most interesting deviations…

1. General Search: Automatically testing a large set of signal regions
Observation of one or more significant deviations in some phase-space region(s)
➔ Trigger to perform dedicated and model-dependent analyses
where these **'data-derived' phase-space region(s) can be used as signal regions**
2. Dedicated Search
- "Wave function collapsed" to test most interesting deviations with available resources
- On 2nd dataset (➔ Statistically independent, unbiased p-value !!)

Dataset 1 signal regions

Dataset 1 signal regions



Dataset 1 step 2 -> dedicated search

Dedicated
Searches -->

(e.g. defining
control regions or
signal hypothesis)

(in the toy example the kids are counting the Lego figures and are trying to estimate how many they had …)

Dataset 1 step 2 -> dedicated search

Assume two regions have
Background issues
→ Dedicated search does
not confirm deviation

step 2 -> test in independent dataset 2

P-value = 10-7
In independent dataset
→ Publish
→ Follow up with CMS

(in the toy example the kids are asking their father for confirmation)

ATLAS scan 2015 data results

Table 4: List of the three channels with the smallest p_{channel}-values in the scan of the m_{inv} distributions.

Largest deviations in m_{inv} scan				
Channel	p_{channel} $(\cdot 10^{-3})$	N_{obs}	$N_{\text{SM}} \pm \delta N_{\text{SM}}$	Region [GeV]
$E_{\text{T}}^{\text{miss}}\, 1\gamma\, 3j$	2.81	9	2.15 ± 0.66	670–732
$1\mu\, 1e\, 4b\, 2j$	2.91	2	0.042 ± 0.037	1227–1569
$1e\, 1b\, 4j$	3.44	160	105 ± 14	726–809

Table 5: List of the three channels with the smallest p_{channel}-values in the scan of the m_{eff} distributions.

Largest deviations in m_{eff} scan				
Channel	p_{channel} $(\cdot 10^{-3})$	N_{obs}	$N_{\text{SM}} \pm \delta N_{\text{SM}}$	Region [GeV]
$1\mu\, 1e\, 4b\, 2j$	2.66	2	0.040 ± 0.036	992–1227
$1\mu\, 1\gamma\, 5j$	3.98	4	0.45 ± 0.18	750–895
$3b\, 1j$	4.87	4	0.42 ± 0.24	3401–3923

No deviation above treshhold …

→ No data-derived signal region yet

Need better variables, “smarter” regions.. Better background model

Supervised ? Unsupervised ? Reinforcement learning task ?

→ Need community effort to help

→ Need <u>> 1 algorithm !</u>(and comparison)

New ideas for searches with unknown signal -> Selection of recent developments in 2017/2018 !

- Fit a ML based background model to be less sensitive on MC prediction (gaussian processes in **arXiv:1709.05681** **)**
- Autoencoders as "filters" for SM events 1808.08992
- Unsupervised techniques (clustering as hypothesis test…)

K- Nearest Neighbour to estimate the point density of two samples, KL-test statistics to compare the samples

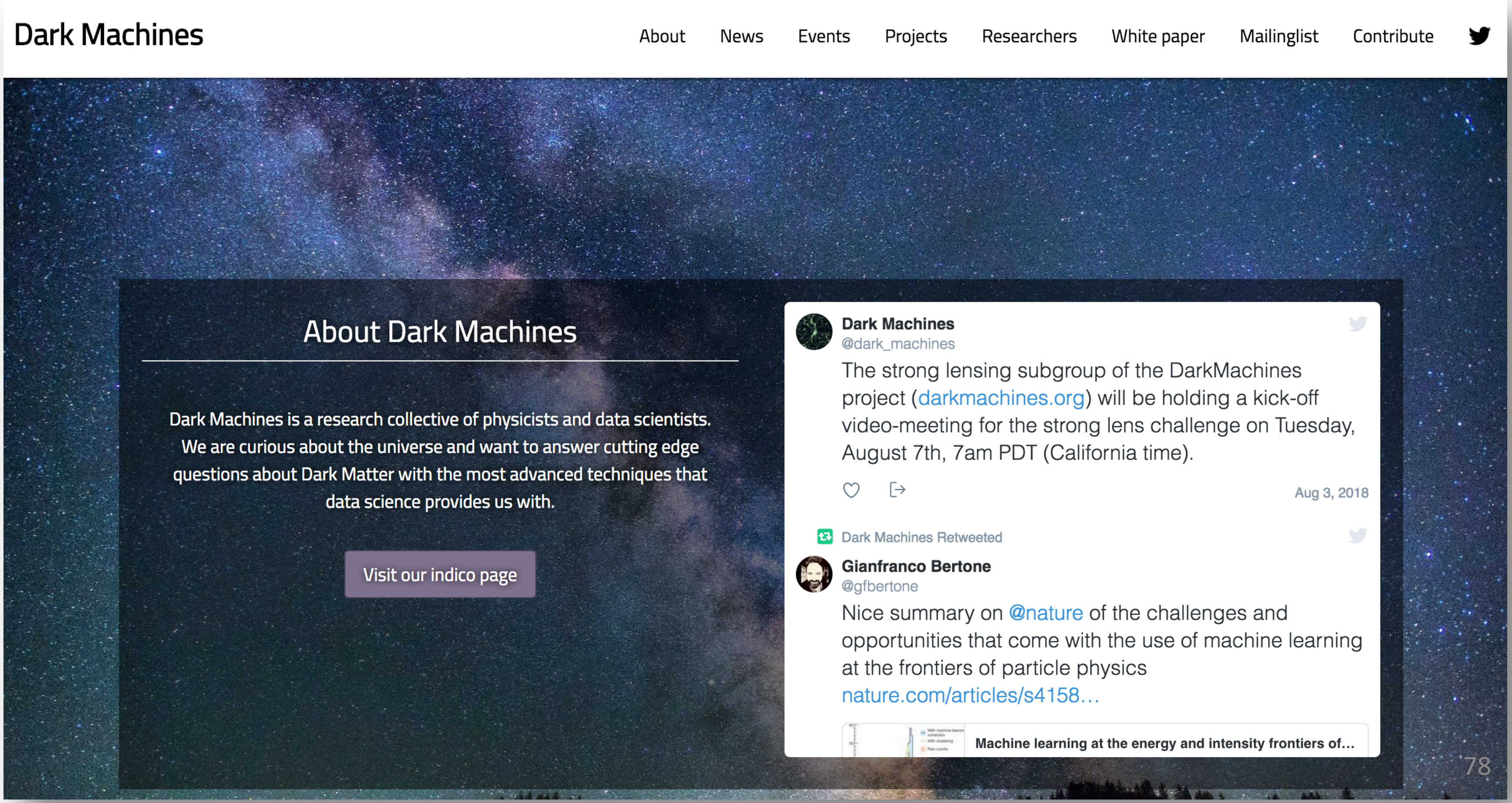
- Classification without Labels (CWOLA) **arXiv:1805.02664:**

Here the idea is to train a NN to seperate signal region + sideband region (as two samples) --> this can be possible due to a signal in the signal region …

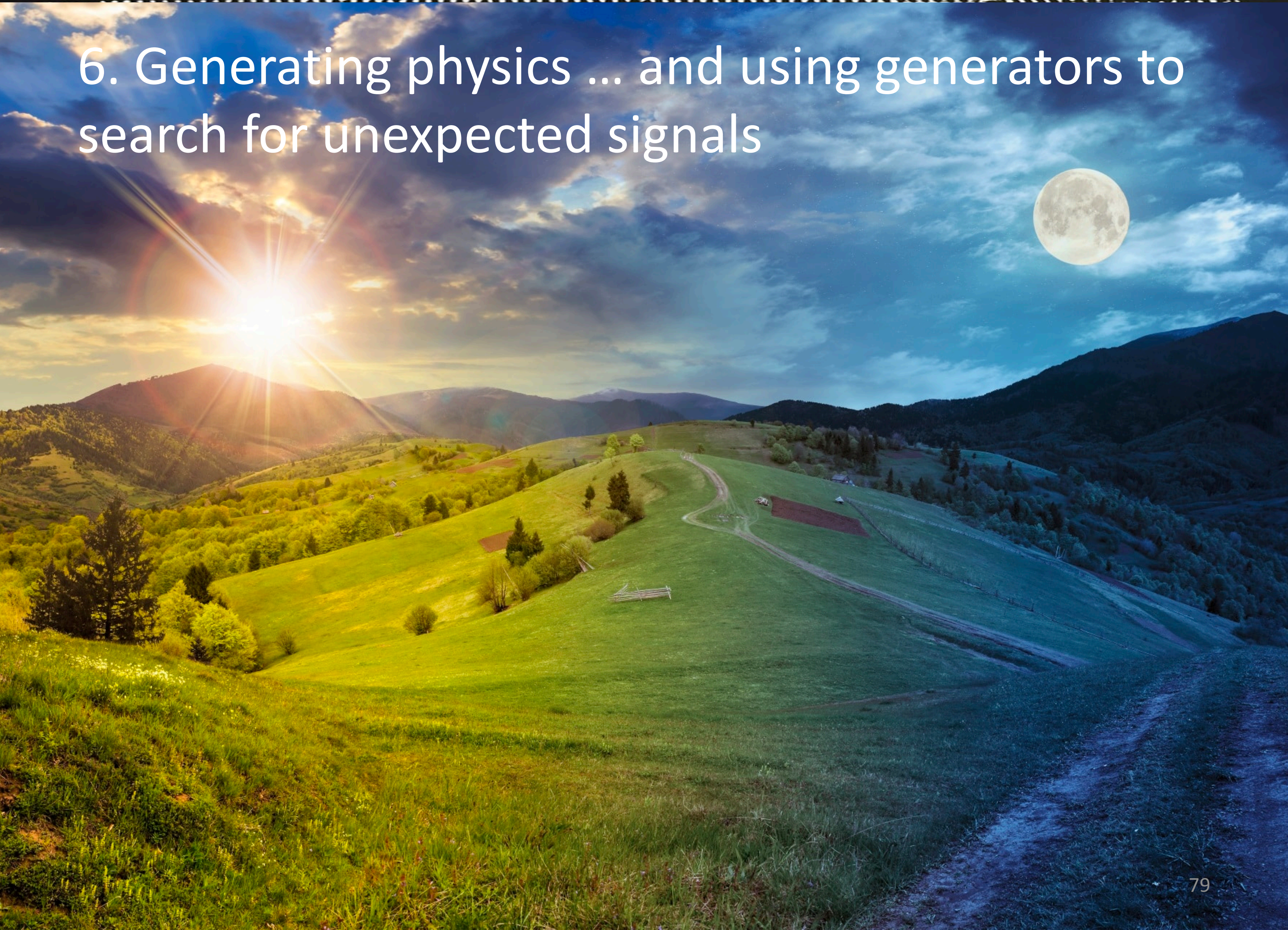
- "Novelty detection algorithm" **arXiv:1807.10261 ,**
- unsupervised KL divergence **arXiv:1807.06038**
- Self-organzing maps…outlier detection with autoencoders …
- … various more !!! (can't catch up anymore, can you ?)
- **Which one is good ? Which one to use ? Need comparison !!!**

Next steps: Compare / Optimize different approaches

e.g. in „unsupervised searches“ group of darkmachines
(Amir Farbin, Erzebet Merenyi, Andrea di Simone, Maurizio Pierini)
e.g. in ATLAS with General Search as prototype data ?



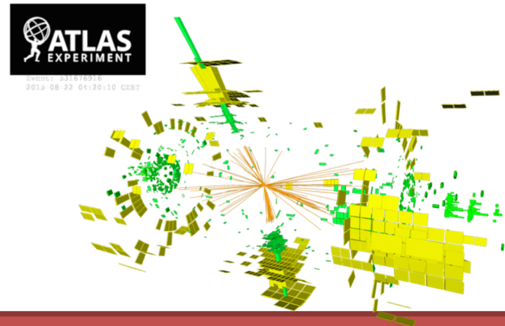
6. Generating physics … and using generators to search for unexpected signals



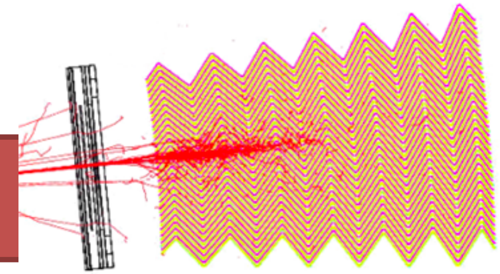
Question: Can we make physical (collider, astroparticle, etc) events with a generative model ?

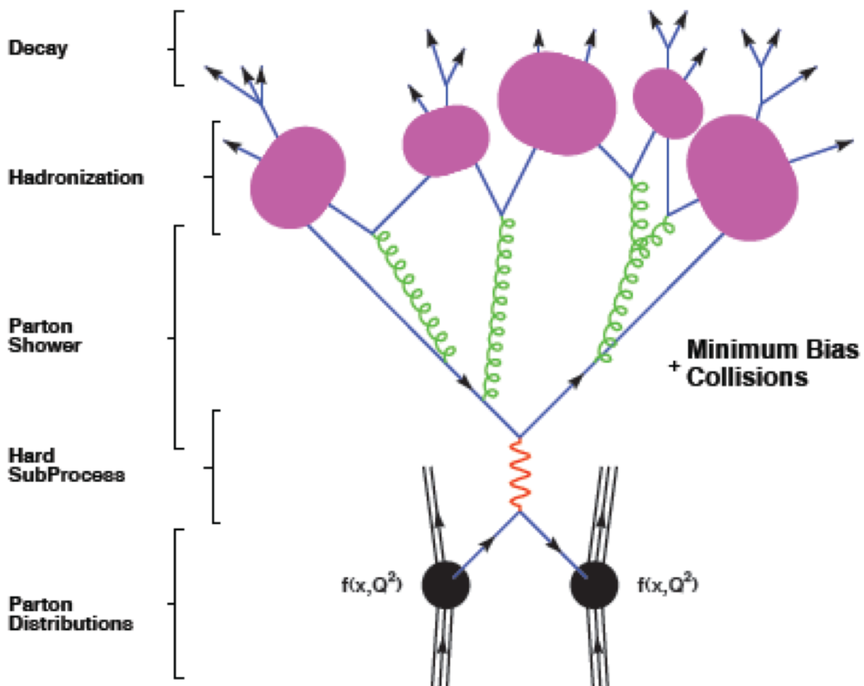
Simulation: Traditional

Energy and angles of reconstructed particles



Detector Simulator

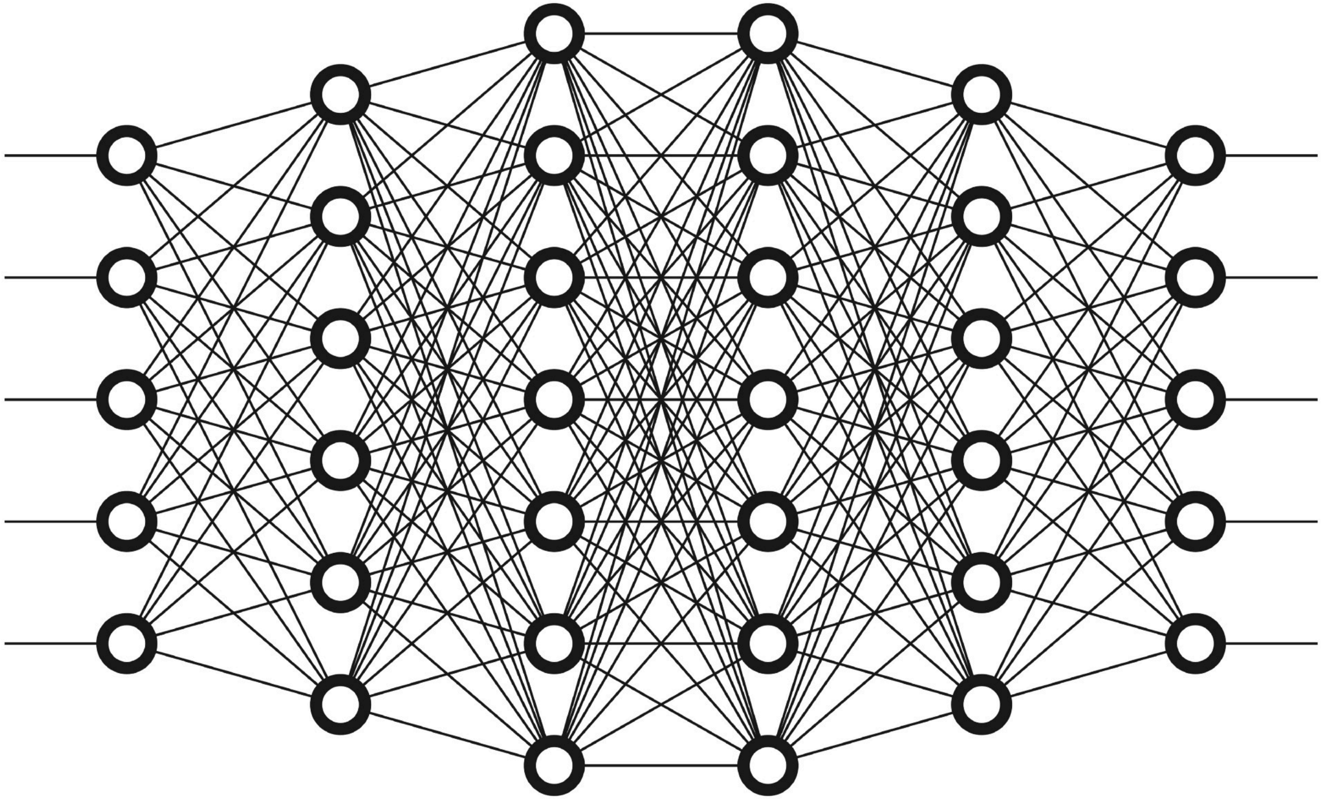




Input:
Random numbers

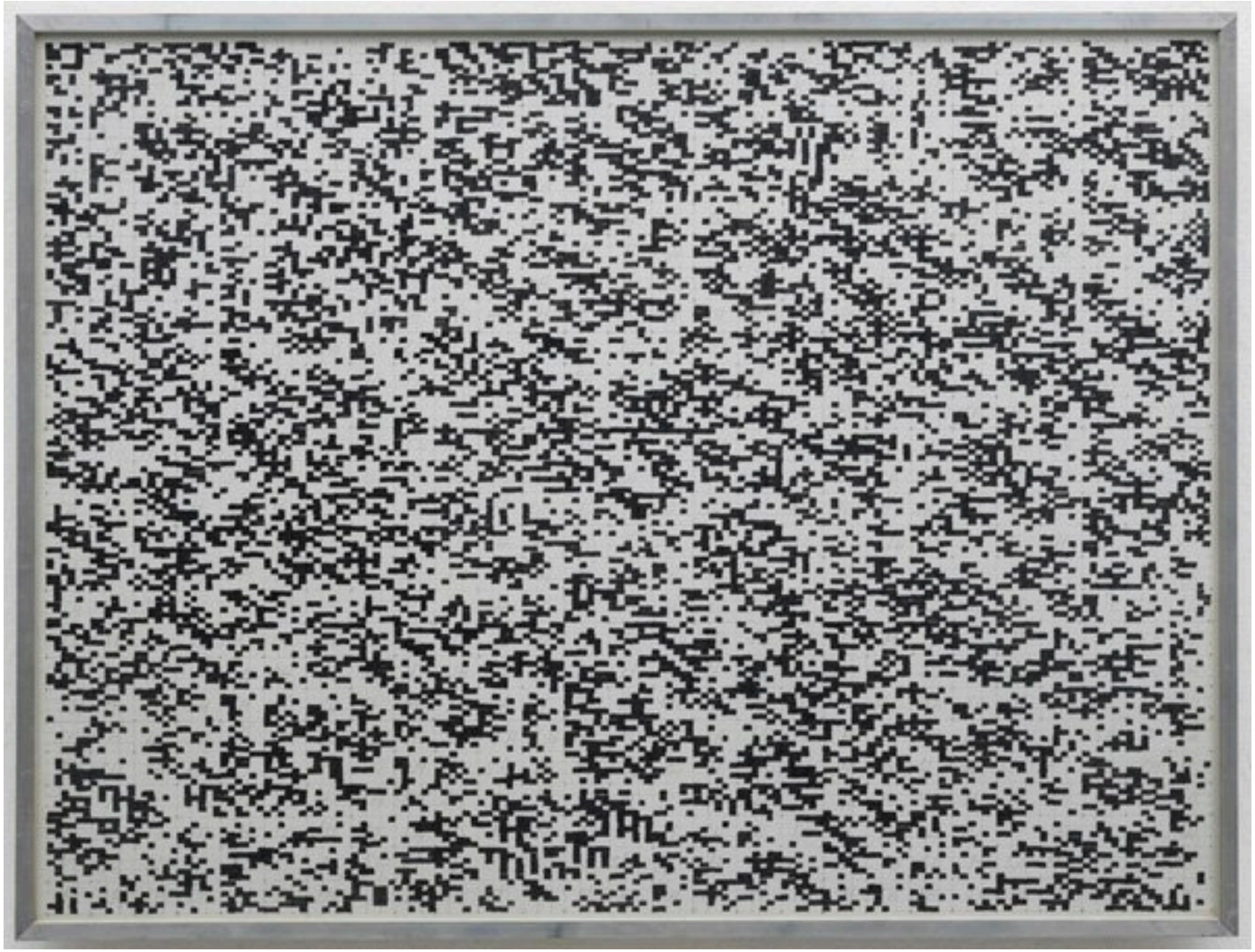
Simulation: Us

Energy and angles of reconstructed particles



Input:
Random numbers

Random numbers…



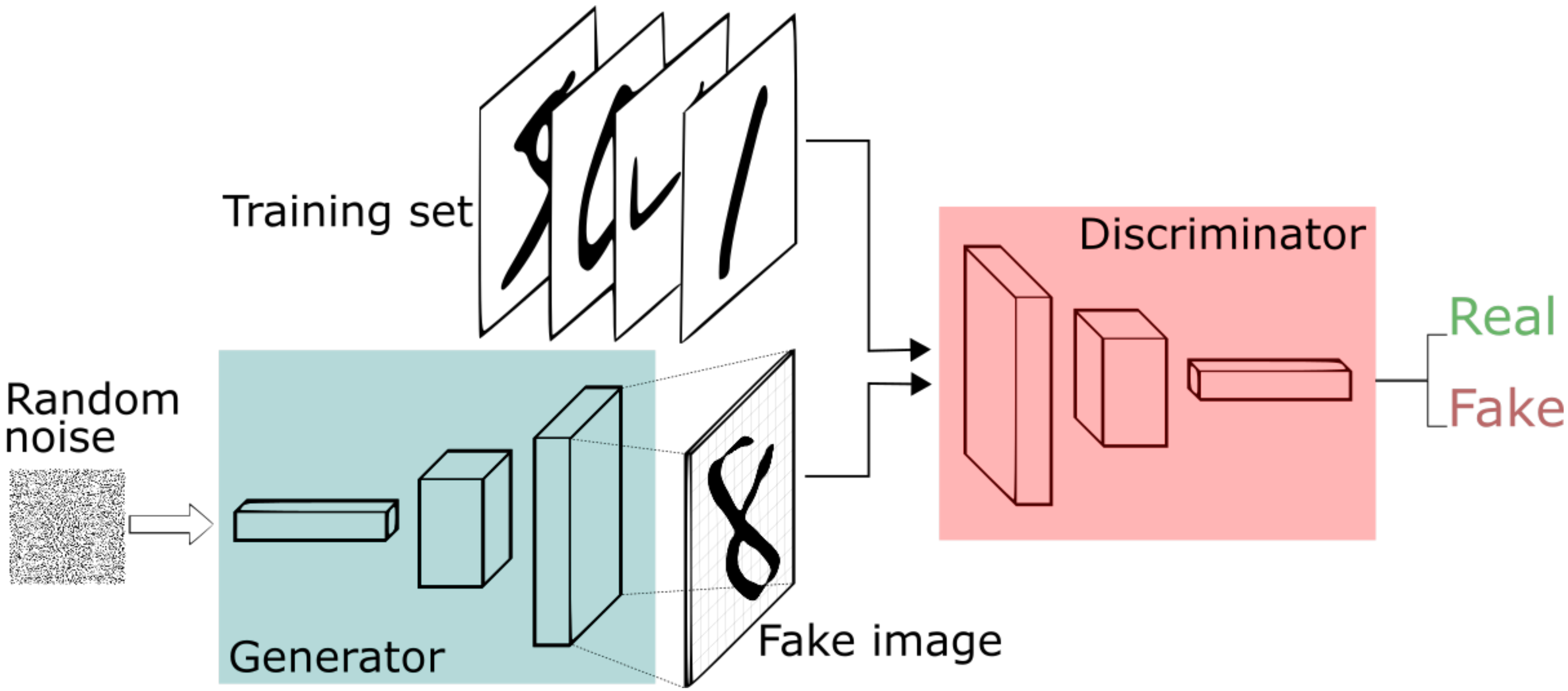
Götz, Karl Otto: Statistisch-metrischer
Versuch 4:2:2:1, Entwurf Sommer 1959

Random input → Art

Tinguely, Meta Matics

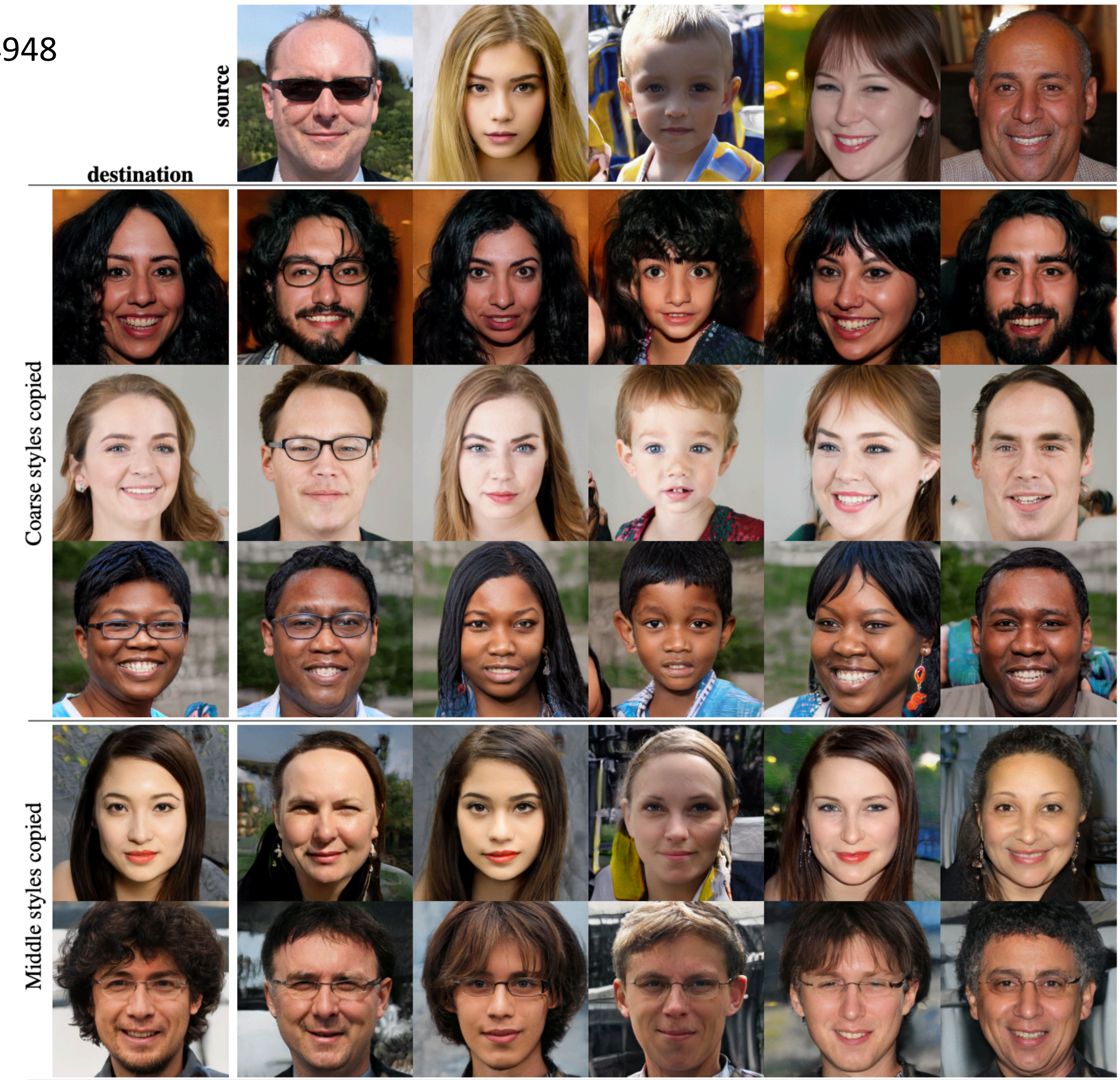
Network simulations ?

Generative Adversarial Networks state of the art:

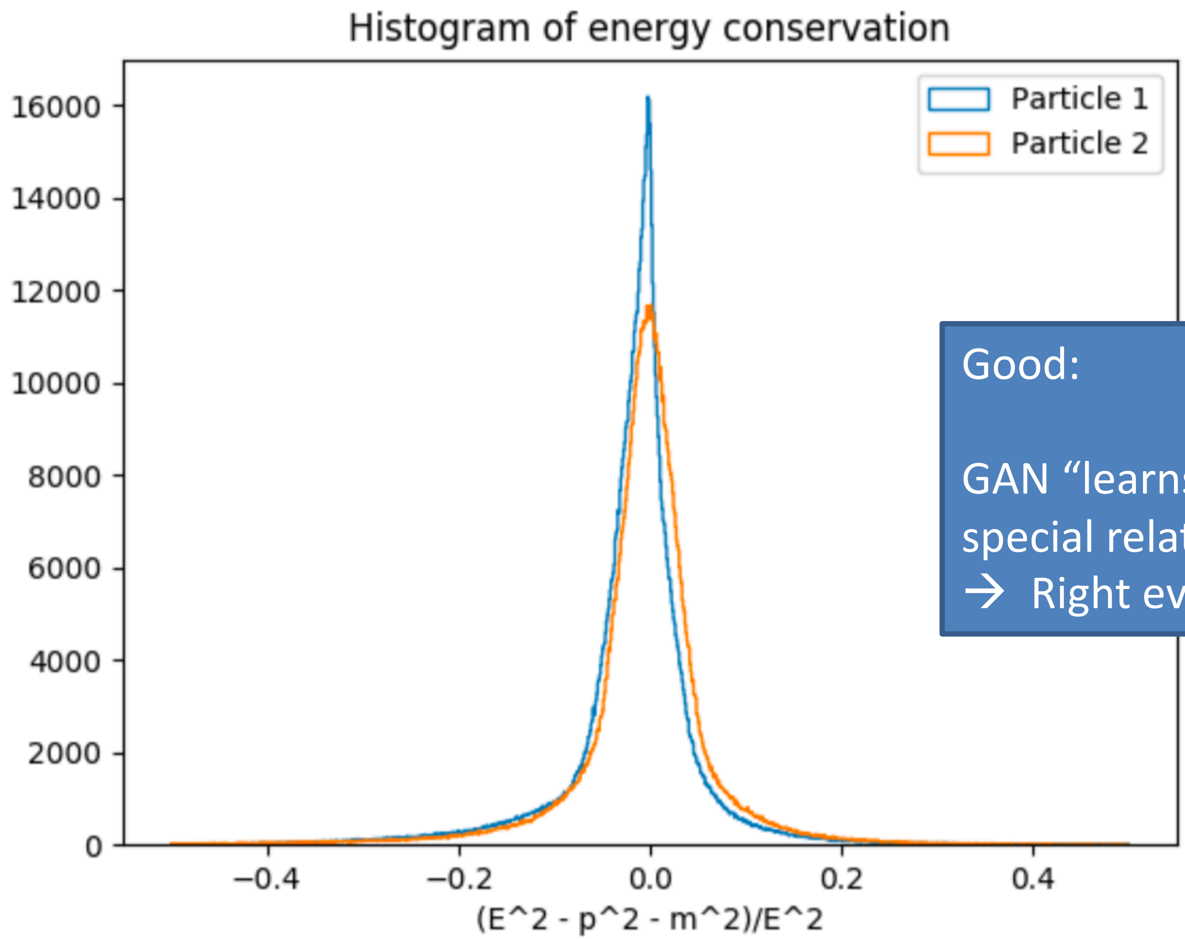


1812.04948

(


source
destination
Coarse styles copied
Middle styles copied

Distributions of Particle Collision “Events“ with GANs

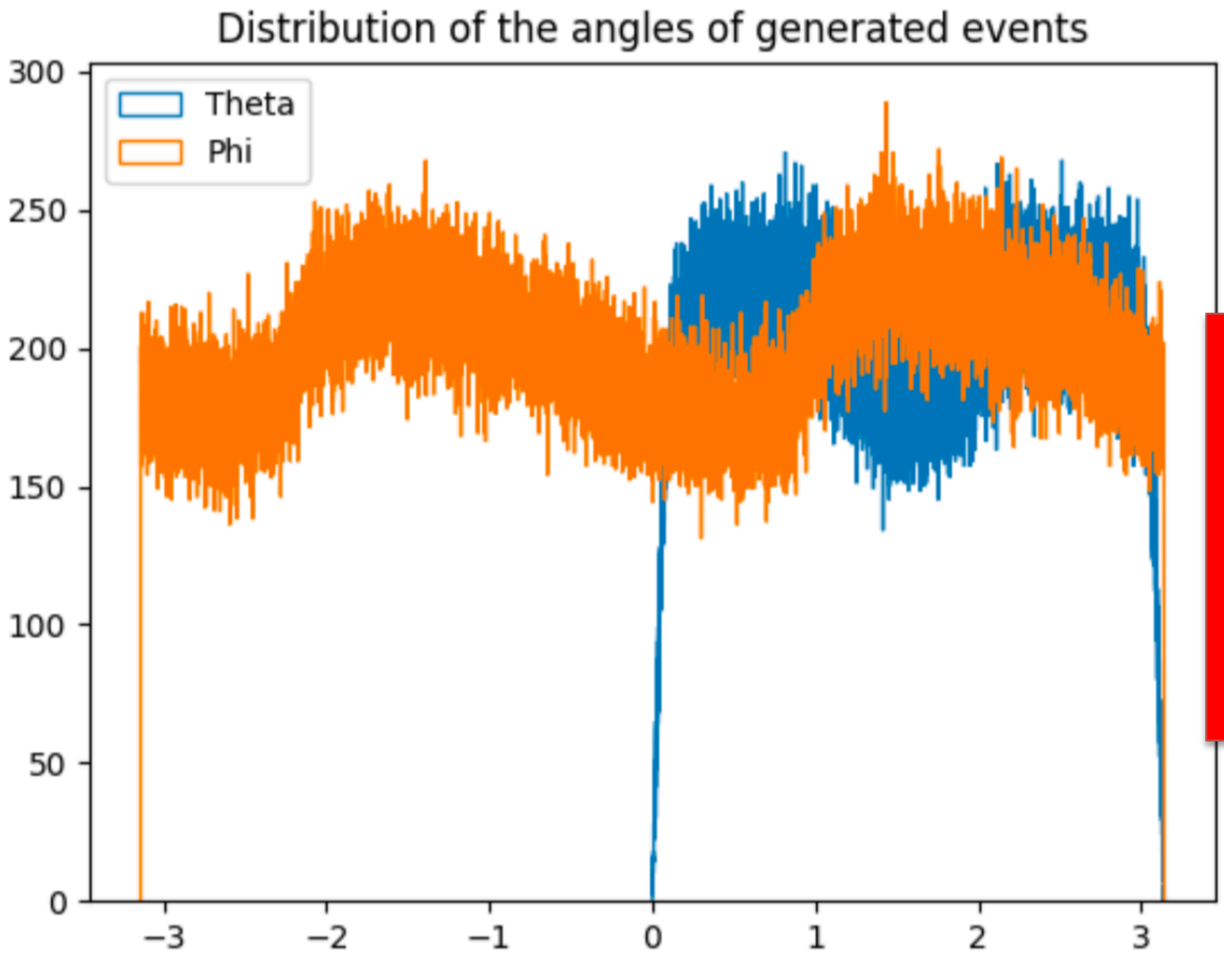


Good:

GAN “learns“ special relativity !

→ Right events

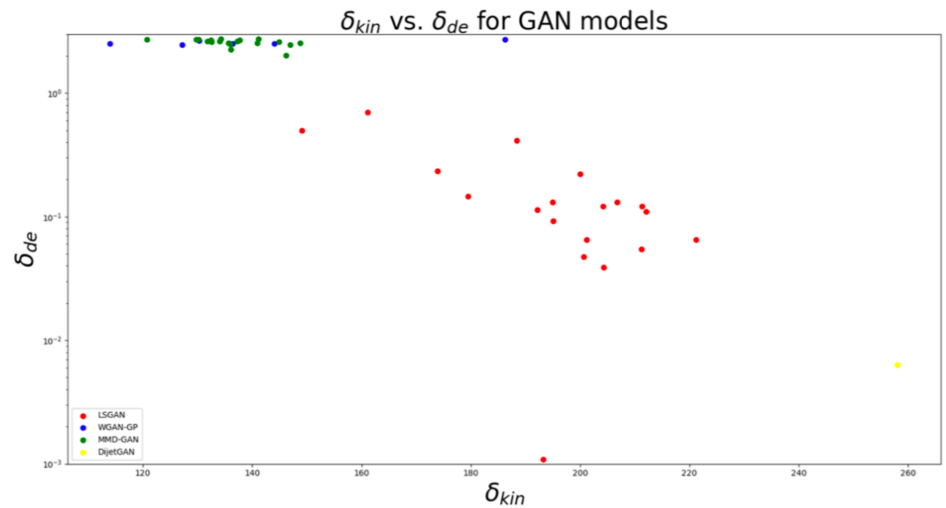
Distributions of Particle Collision "Events" with GANs

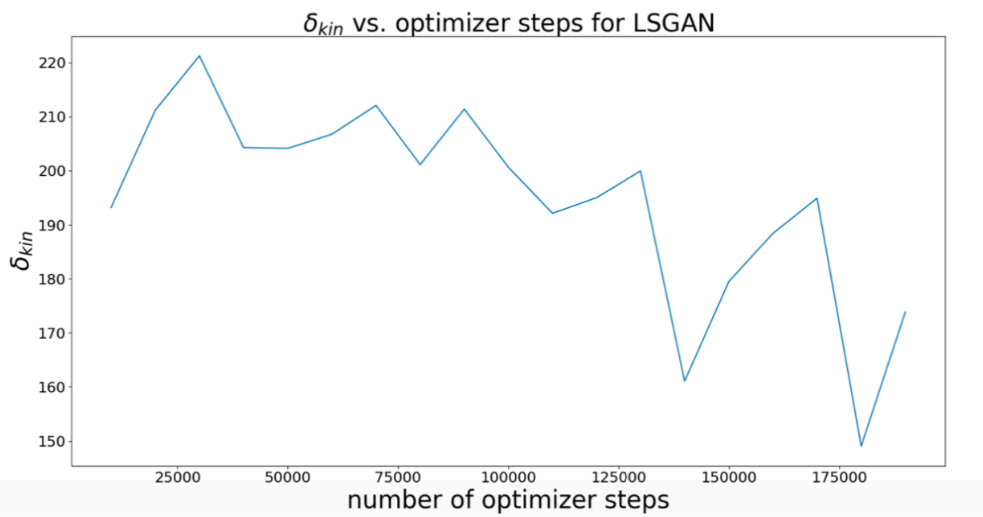


BAD:

GAN does not make events of different types with right frequencies !

Our GAN attempt





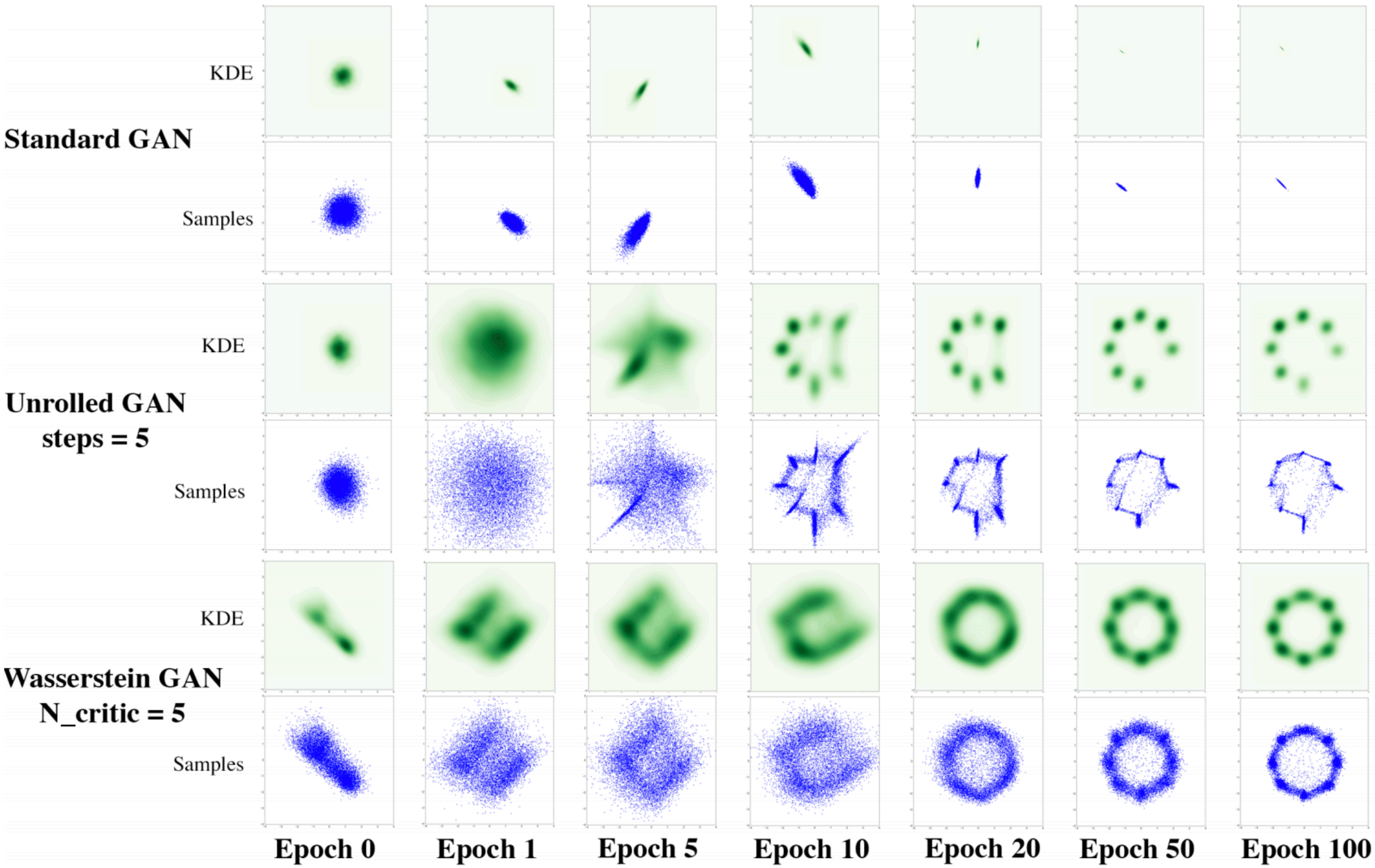
Tried 5 different GAN architectures

Training highly unstable

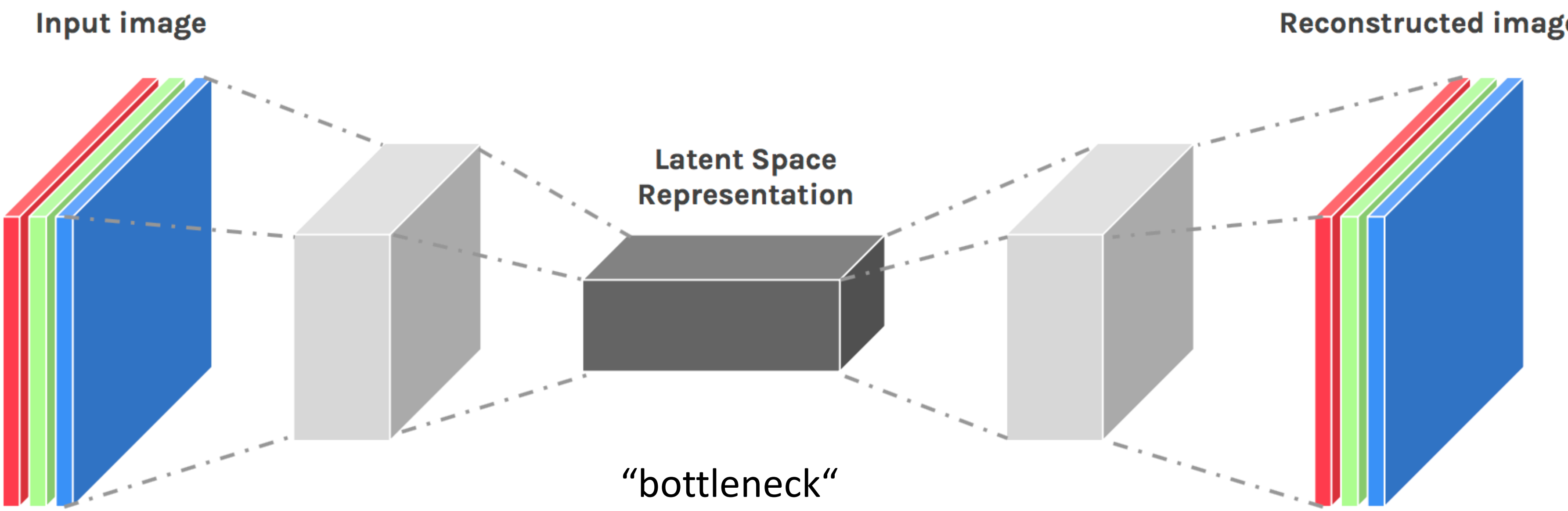
Not able to sample 25d density

Densities with GANs

http://proceedings.mlr.press/v70/arjovsky17a/arjovsky17a.pdf



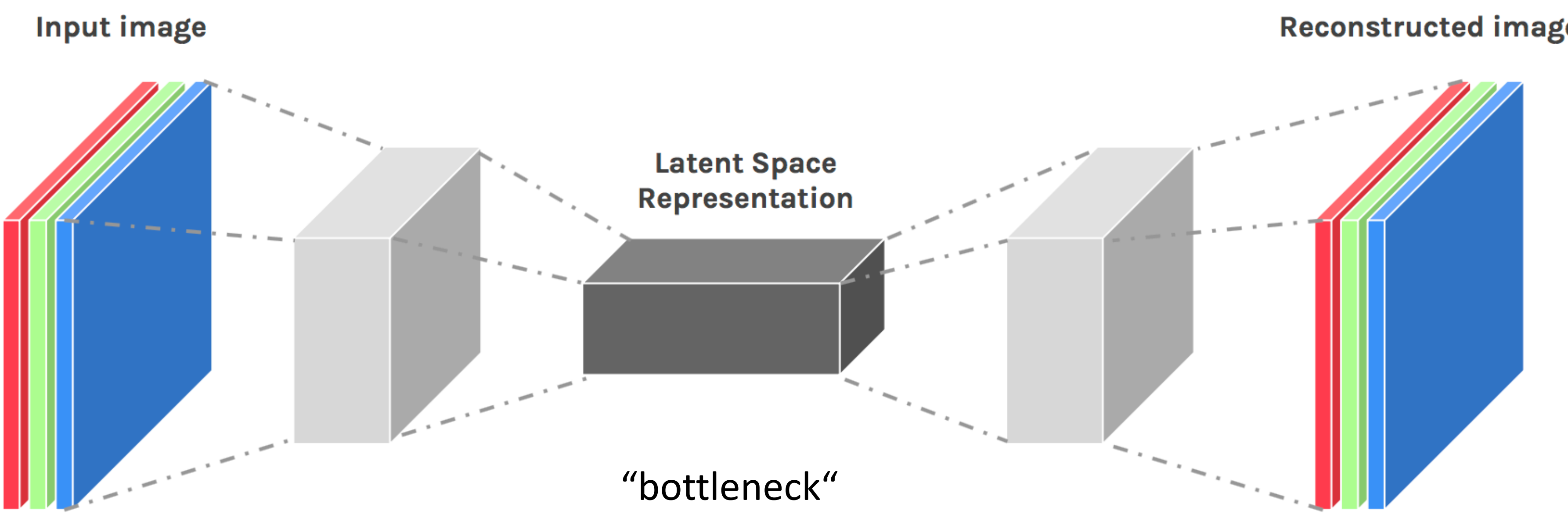
Autoencoders



Can find new physics
with reconstruction Loss

We actually use a better version:
„Dutch“ Autoencoder
(Variational Autoencoder by Dederik Kingma and Max Welling)

Variational Autoencoders



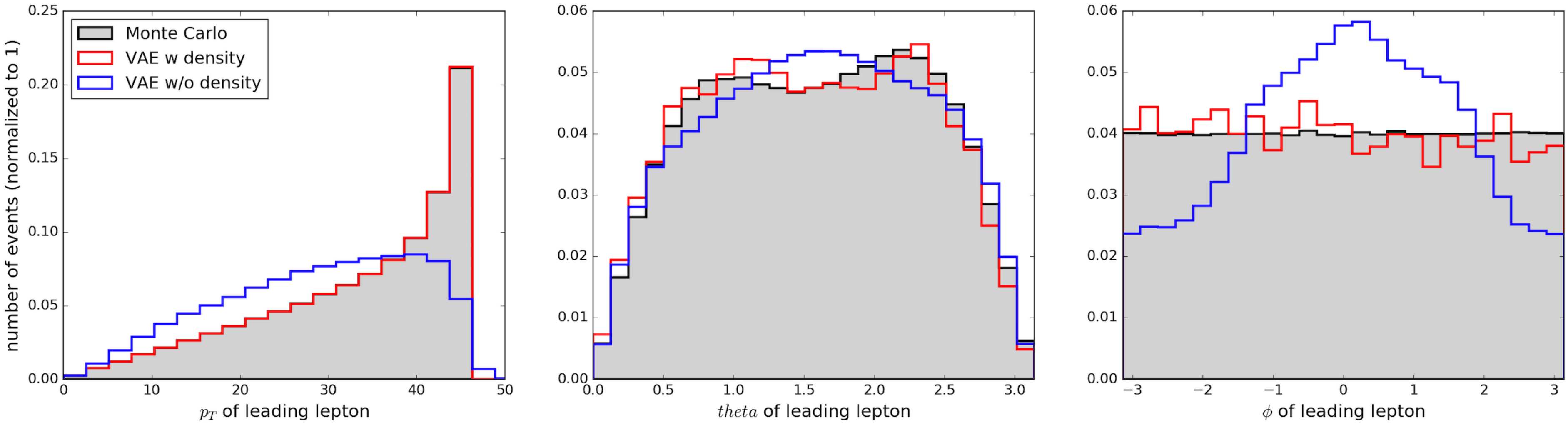
“bottleneck“

Encoder output is the mean and the variance of d Gaussians
Decoder input is z : a sample drawn from these d Gaussians

We proposed to use B-VAE (B<<1)

$$L = \frac{1}{M}\sum_{i=1}^{M}(1-B)\cdot MSE + B\cdot D_{KL}.$$

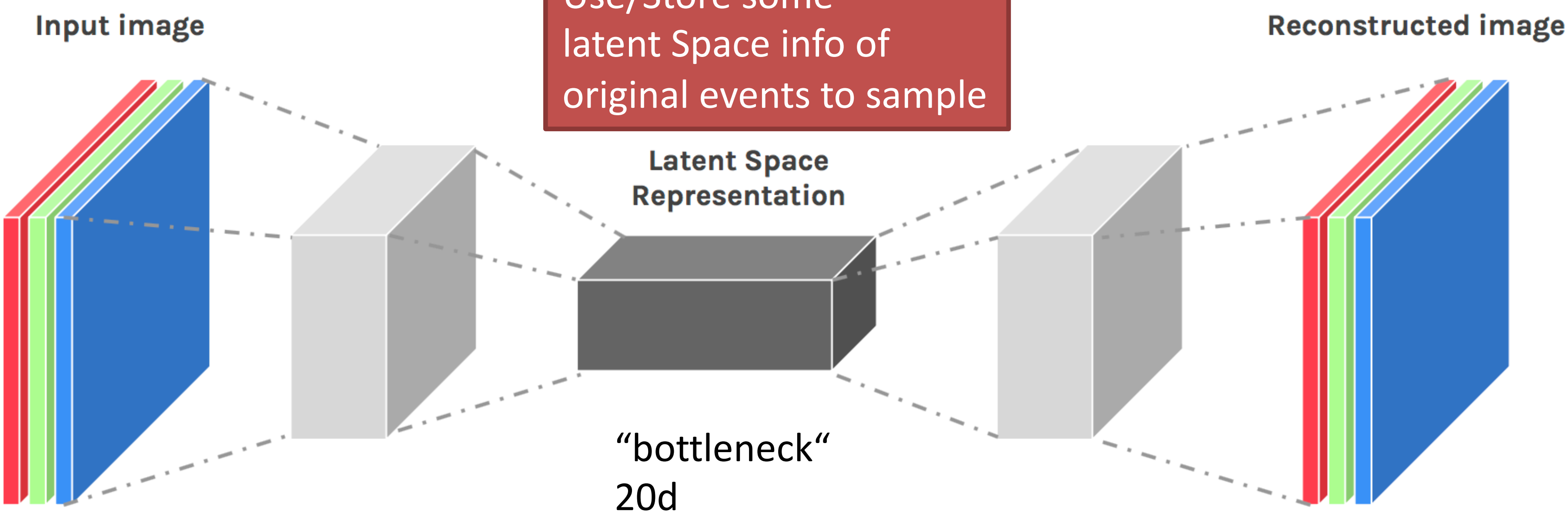
Distributions of Particle Collision “Events“ with variational autoencoders



BAD:
Autoencoder typically does not
make events of different types with right
frequencies !

Autoencoder
(+ event info in latent space)

Use/Store some latent Space info of original events to sample



“bottleneck“
20d

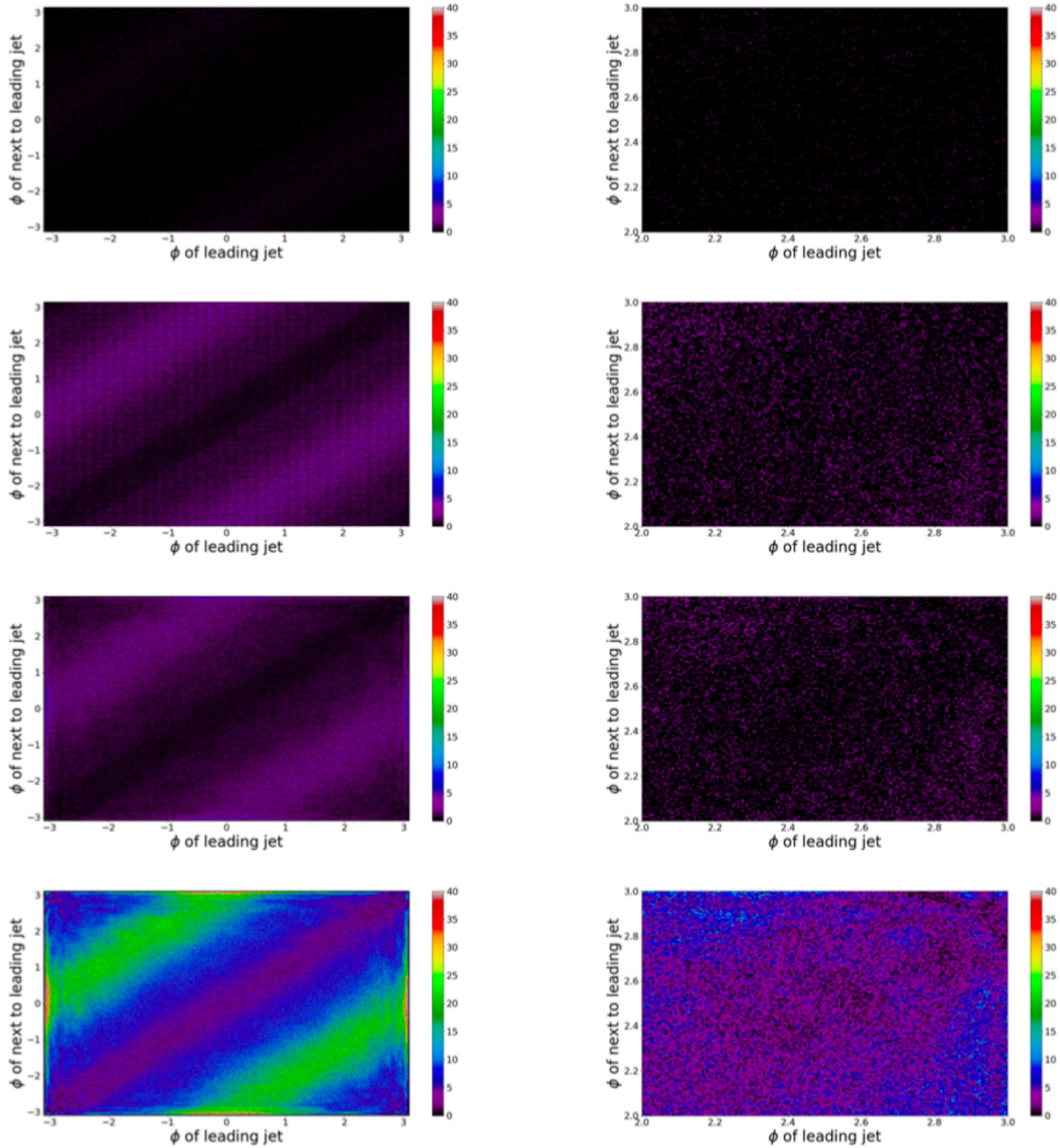
24d

24d

VAE: p(z) is typically from d dimensional gauss,
VAE with buffer:

$$p_{\phi,X_L}(z) = \sum_{i=1}^{m} q_\phi(z|x^i)p(x^i) \text{ with } \mathrm{p}(\mathrm{x}^{\mathrm{i}}) = \frac{1}{\mathrm{m}}.$$

https://arxiv.org/abs/1901.00875 , mainly Sydney Otten (ex RWTH)



a) full range

b) zoom in on $[2,3] \times [2,3]$

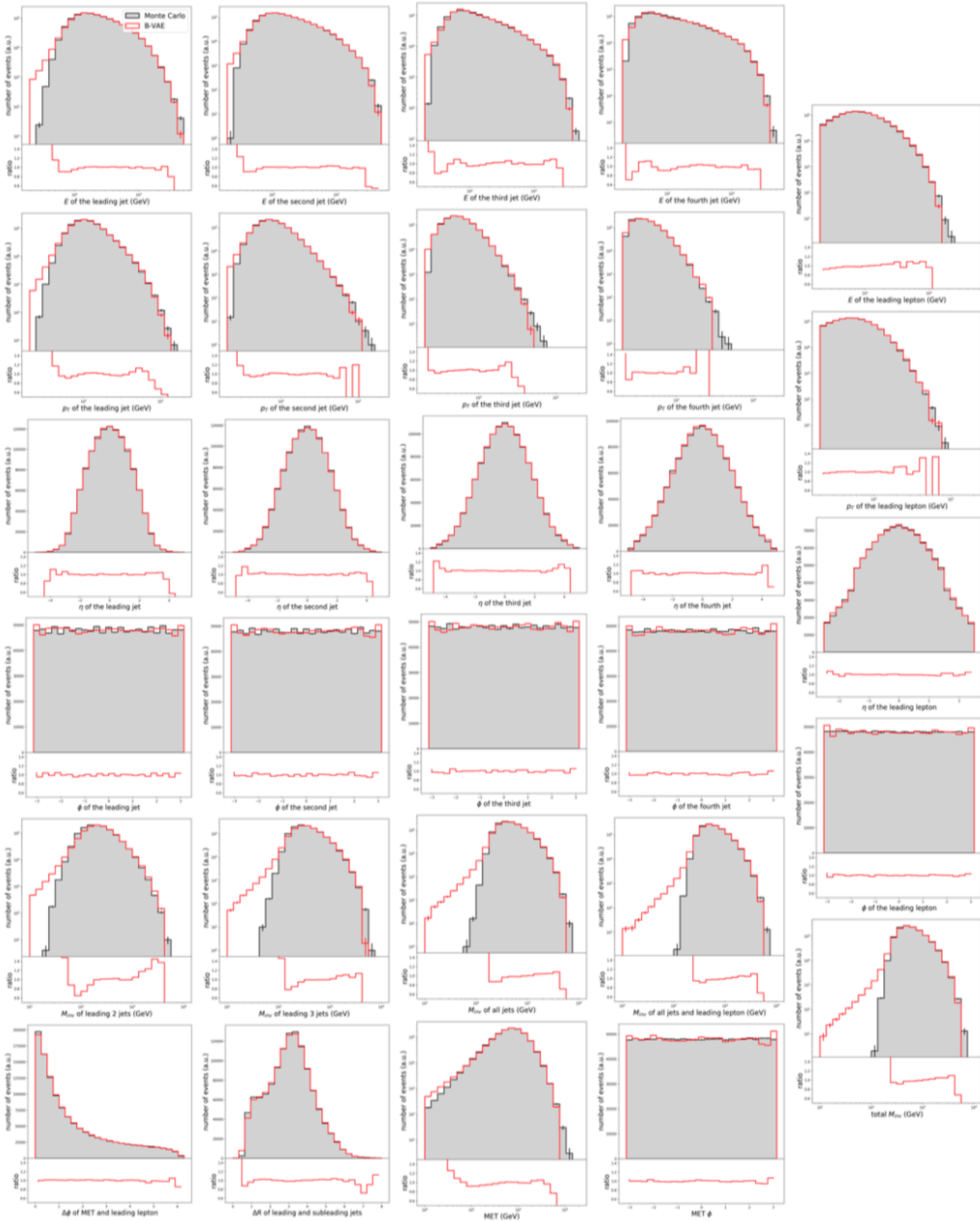
Training data
100k

2d projection
of density

Monte Carlo
1M events

B-VAE
1M events

B-VAE
12M events



Sampling top top -> 6 particles

https://arxiv.org/abs/1901.00875

1d distributions :
Red: B-VAE
Grey: MC (Madgraph+Delphes)

Rank	$(\dim(z), B, \alpha = 1, \gamma)$	δ_{kin}	δ_{de}
1	$(20, 10^{-6}, 1, 0.01)$	483.5	0.0067
2	$(20, 10^{-7}, 1, 0.01)$	481.2	0.0068
3	$(16, 10^{-7}, 1, 0.01)$	471.8	0.0081

Model	δ_{kin}	δ_{de}
KDE	249.0	0.4934
GMM, 50	279.9	1.4457
GMM, 100	291.2	1.5141
GMM, 1000	307.1	1.5232
5 % Smearing	505.3	0.1316
10 % Smearing	442.6	0.3186

TABLE I: KDE and GMM model performanc evaluated on figures of merit δ_{kin} and δ_{de}.

Noise input vs True events input → B-VAE as anomaly detector

15

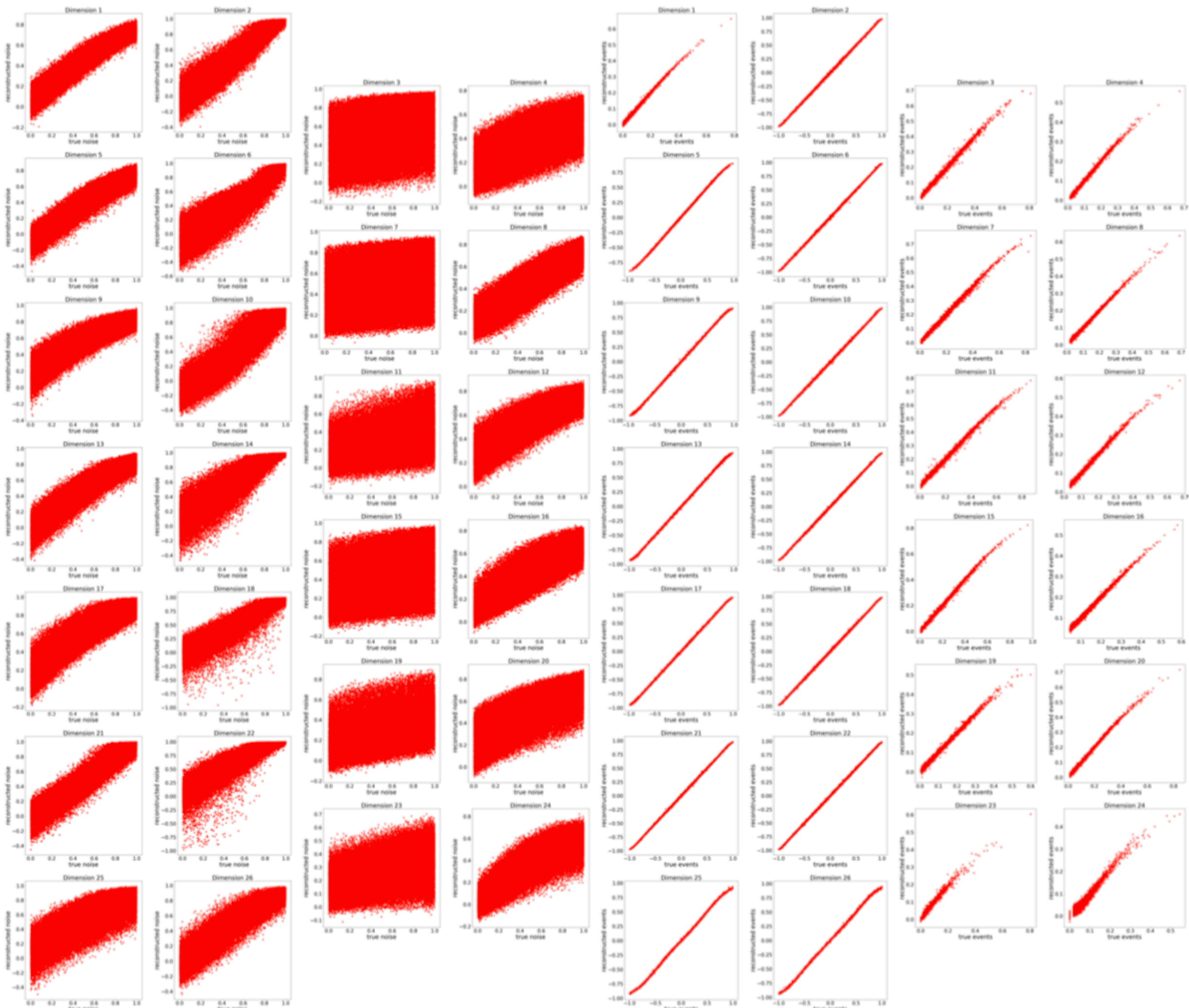


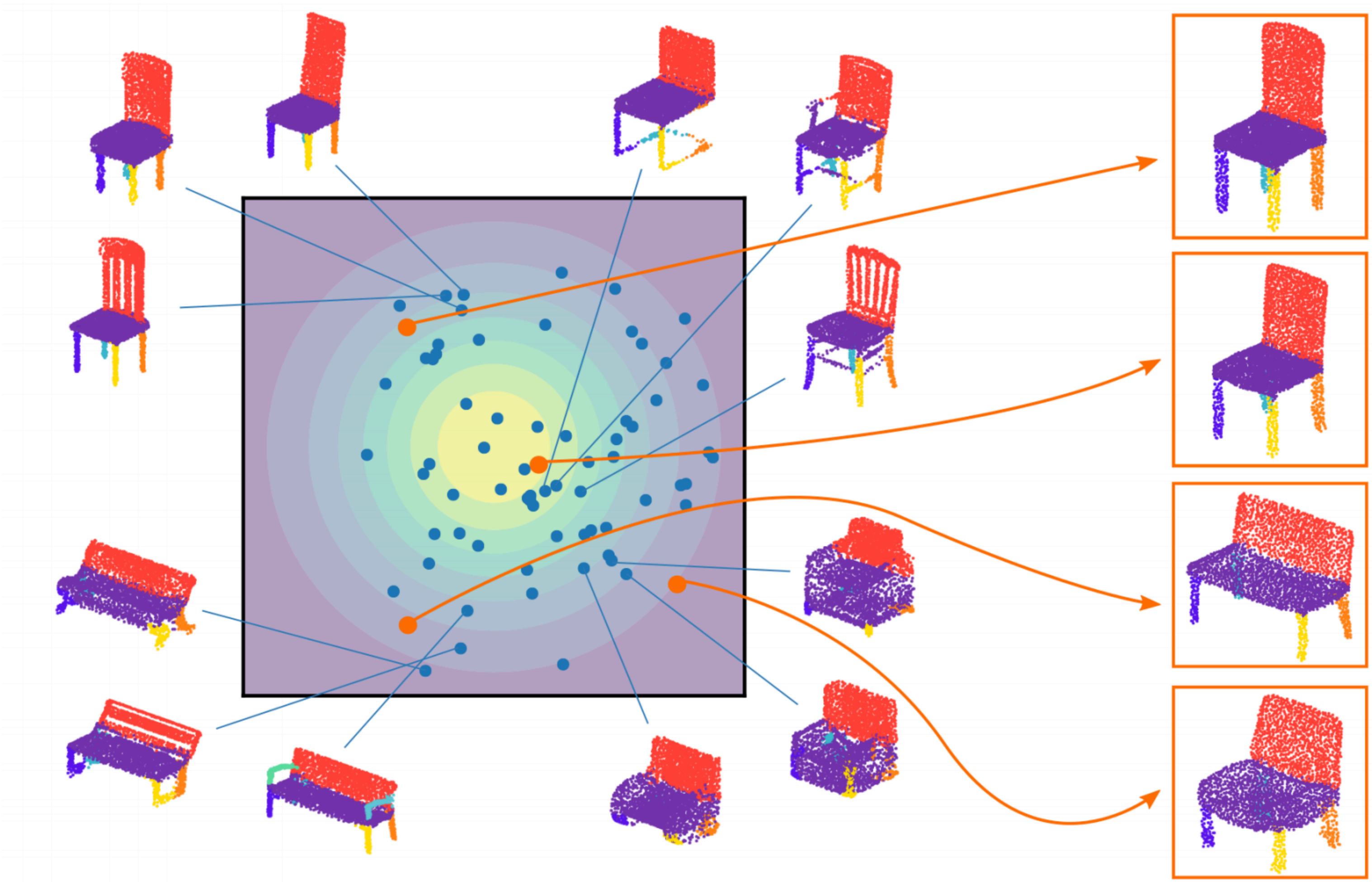
FIG. 1: Input vs. Reconstruction of uniform noise $x \sim U(0,1)$ (first four columns) and real events (last four columns) for a VAE with $\dim(z) = 20$ and $B = 10^{-6}$.

Why is this useful ?

Can „store“ events in *lower dimensional* latent space and interpolate between them

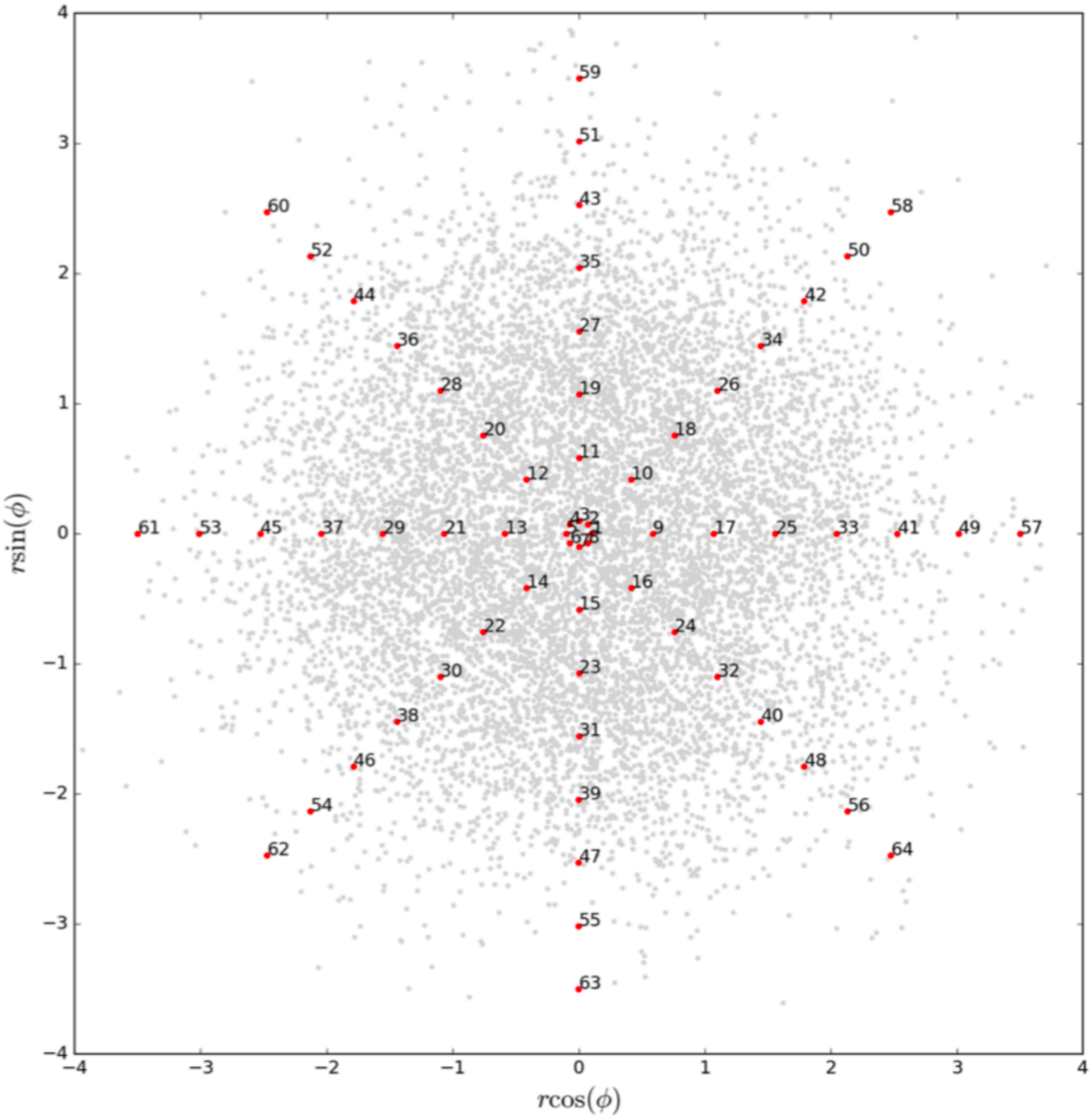
→New events (by interpolation and compression/decompression)
→New concepts (by interpolation)
→New models (by interpolation)
→Better random number sampling
→Ultrafast (Million events per second compared to 1 event per minute …)

Concept of a latent space of sofas and chairs



arxiv 1610.07584

Top top Latent space PCA1 vs PCA2



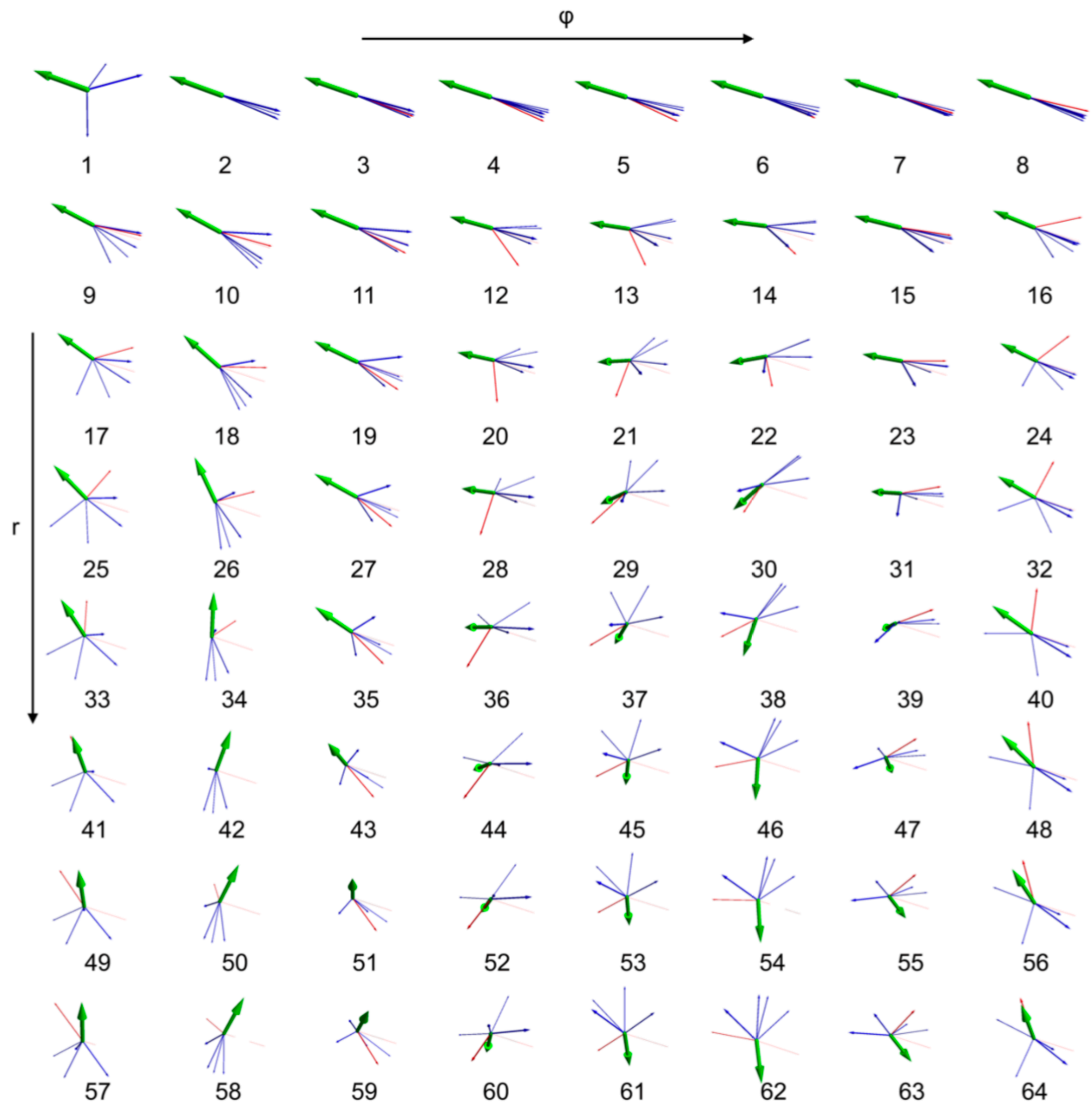


FIG. 7: Visualization of the first two components of a principal component analysis of encoded Monte Carlo events in latent space. This shows an 8 × 8 grid of event displays following the red dots in Figure 6. These 64 points chosen

Summary

1. Automatization
2. Data derived signal regions
3. Learn the model space
4. Learn the simulator

Use this for your search

Extra Slides

Main message: Parameter determination of the physical model with a Neural Network

- Finally our goal is to **determine the model parameters** from 1 image ("real data")
- We do this by **training the network on "simulation"** ("simulated data")
- We need to **ensure that simulation agrees with data:** *Is the true image in the simulation parameter space of images ?* ***If not DM parameters maybe wrong !***

In simple words we do a "fit" to the image including all kinds of "unknown correlations" using a deep convolutional network trained on simulations