UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

bwHPC Symposium Freiburg
25.09.2024

New ML-based analysis techniques in fundamental physics

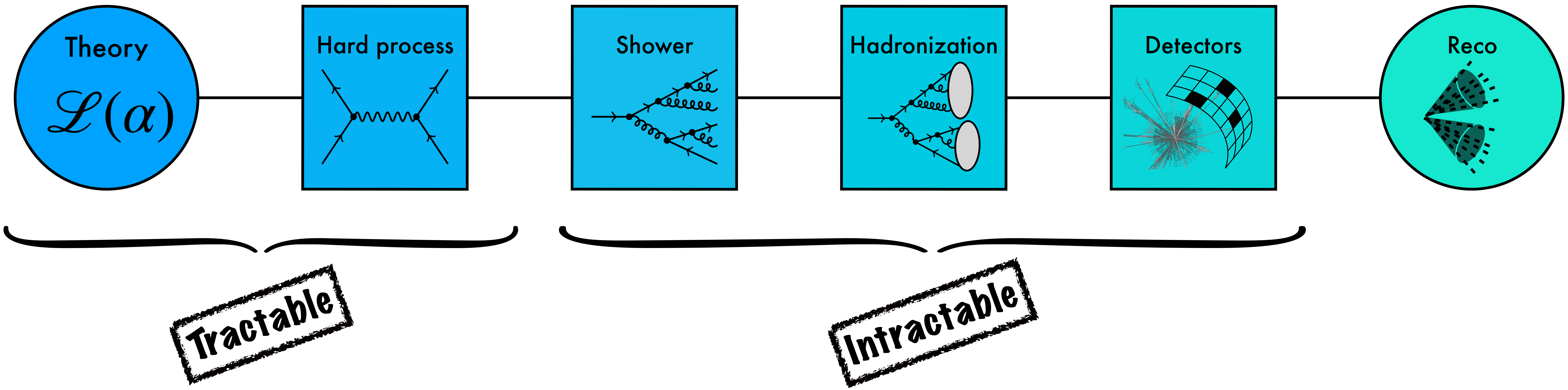
SPONSORED BY THE

Federal Ministry
of Education
and Research

Nathan Huetsch

On behalf of the Heidelberg group led by Tilman Plehn

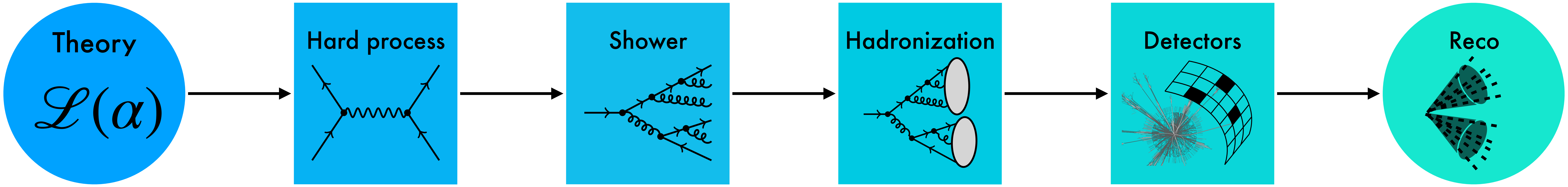
From Theory to Experiment in LHC Physics



In HEP we can never analytically calculate what we measure

We rely on simulations to connect theory and experiment

From Theory to Experiment in LHC Physics



$p(x_{gen} | \alpha)$

$p(x_{reco} | \alpha)$

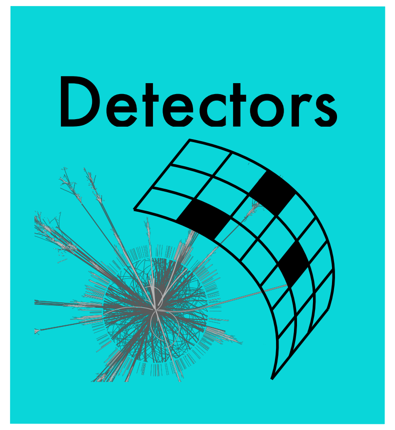
Predicted by some new theory

Measured in an experiment

Unfolding

Detector kernel

$p(x_{rec} \,|\, x_{gen})$

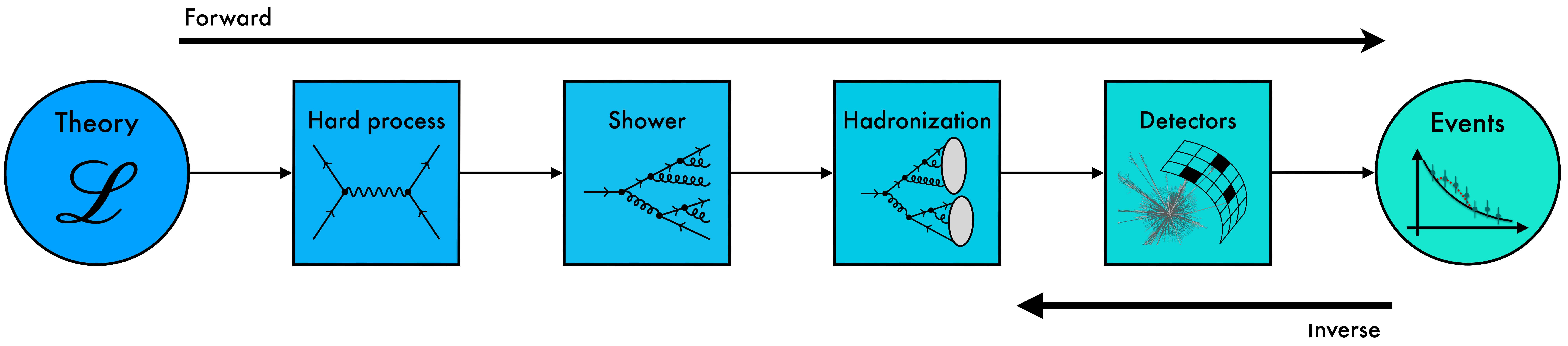
$p(x_{gen}) \longrightarrow$ 

$\longrightarrow \quad p(x_{rec}) = \int p(x_{gen}) p(x_{rec} \,|\, x_{gen}) \, dx_{gen}$

The measured distribution $p(x_{rec})$ is a convolution of the generation level distribution $p(x_{rec})$ with detector the response kernel $p(x_{rec} \,|\, x_{gen})$

The task of statistically correcting for these effects is called Unfolding

Why Unfolding?



Theory analyses don’t care about detectors

Comparing data from different experiments (Global Analysis)

For some analysis direct access to theory parameters

Resolution

Data preservation

How Unfolding?

Classical methods:

Have been around and used for a while now

Computationally very efficient

Restricted to binned, 1-dimensional distributions

ML-based methods:

Used for the first time in an ATLAS analysis this year!

Computationally more expensive

Allow unbinned, full-dimensional unfolding of measurements

How Unfolding?

ML-based methods:

Used for the first time in an ATLAS analysis this year!

Computationally more expensive

Allow unbinned, full-dimensional unfolding of measurements

ATLAS analysis [2405.20041]

Omnifold [1911.09107]

Makes use of NN classifiers to iteratively reweight a simulation prior until it fits the measurements

Generative Unfolding [1912.00477]

Makes use of generative NNs to learn the conditional distribution $p(x_{gen} | x_{rec})$ from simulations

How Unfolding?

ML-based methods:

Used for the first time in an ATLAS analysis this year!

Computationally more expensive

Allow unbinned, full-dimensional unfolding of measurements

Omnifold [1911.09107]

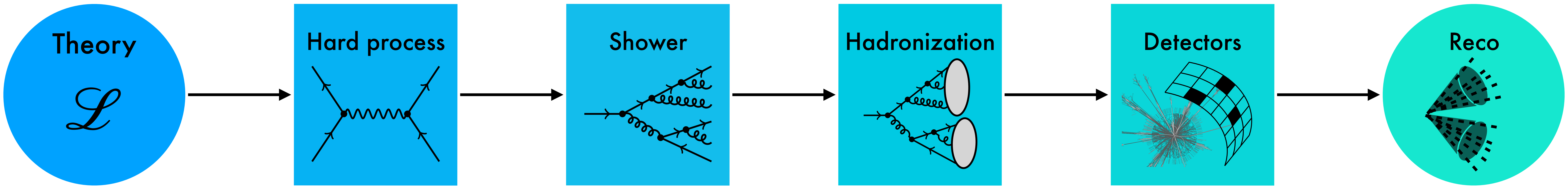
Makes use of NN classifiers to iteratively reweight a simulation prior

Both suffer from prior dependence

Generative Unfolding [1912.00477]

Makes use of generative NNs to learn the conditional distribution $p(x_{gen} | x_{rec})$ from simulations

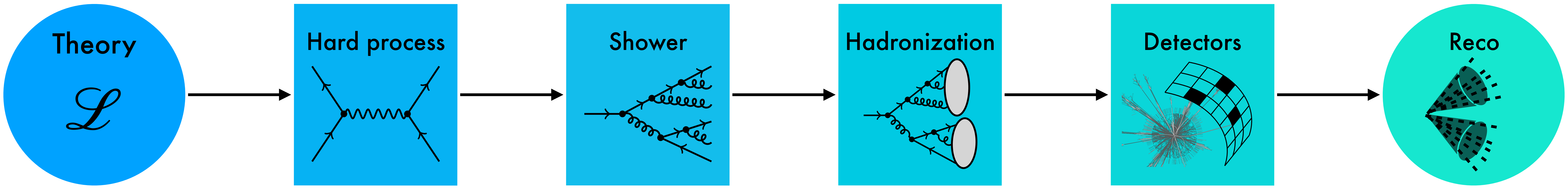
Revisiting the problem



$$\int dx_{gen} \qquad p(x_{gen}) \qquad p(x_{reco} \,|\, x_{gen}) \quad = \quad p(x_{reco})$$

The task is to find the generation level distribution $p(x_{rec})$ **that gave rise to the observed distribution** $p(x_{rec})$

Revisiting the problem



$$\int dx_{gen} \quad p(x_{gen}) \quad p(x_{reco}\,|\,x_{gen}) \quad = \quad p(x_{reco})$$

The task is to find the generation level distribution $p(x_{rec})$ that gave rise to the observed distribution $p(x_{rec})$

Just directly optimize for this objective!

Transfer-based Unfolding

1) Use a generative NN $p_\theta(x_{gen})$ to encode the unfolded generation level distribution

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$ using the forward detector kernel $p(x_{rec} | x_{gen})$

3**)** Compare the result to the measured detector level distribution $p(x_{rec})$

4**)** Update $p_\theta(x_{gen})$ until the convoluted $p_\theta(x_{rec})$ matches the measured $p(x_{rec})$

Transfer-based Unfolding

1) Use a generative NN $p_\theta(x_{gen})$ to encode the unfolded generation level distribution

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$ using the forward detector kernel $p(x_{rec} | x_{gen})$

→ **Doing this with the actual detector simulation is not feasible.**
Train a surrogate NN to encode the detector kernel $p_\phi(x_{rec} | x_{gen})$

→ **The integral has to be approximated with a Monte-Carlo estimate**

3**)** Compare the result to the measured detector level distribution $p(x_{rec})$

→ **Maximize the likelihood of the true data under our model distribution** $p_\theta(x_{rec})$

4**)** Update $p_\theta(x_{gen})$ until the convoluted $p_\theta(x_{rec})$ matches the measured $p(x_{rec})$

Transfer-based Unfolding

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen})p(x_{rec}\,|\,x_{gen})\;dx_{gen}$ using the forward detector kernel $p(x_{rec}\,|\,x_{gen})$

$$p_\theta(x_{rec}) = \int p_\theta(x_{gen})p(x_{rec}\,|\,x_{gen})\;dx_{gen}$$

$$\approx \int p_\theta(x_{gen})p_\phi(x_{rec}\,|\,x_{gen})\;dx_{gen}$$

Replace simulation with surrogate NN

$$\approx \sum_{i=1}^{N_{MC}} p_\theta(x_{i,gen})p_\phi(x_{rec}\,|\,x_{i,gen})$$

Monte-Carlo approximation of the integral

Transfer-based Unfolding

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$ using the forward detector kernel $p(x_{rec} | x_{gen})$

$$p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$$

$$\approx \int p_\theta(x_{gen}) p_\phi(x_{rec} | x_{gen}) \, dx_{gen}$$

Replace simulation with surrogate NN

$$\approx \sum_{i=1}^{N_{MC}} p_\theta(x_{i,gen}) p_\phi(x_{rec} | x_{i,gen})$$

Monte-Carlo approximation of the integral

This equation is for one individual data point.
When training the network we have to calculate this for each data point in each iteration.
To get a reasonable MC approximation we have to draw $\mathcal{O}(100)$ samples per data point

Transfer-based Unfolding

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen})p(x_{rec}\,|\,x_{gen})\;dx_{gen}$ using the forward detector kernel $p(x_{rec}\,|\,x_{gen})$

$$p_\theta(x_{rec}) = \int p_\theta(x_{gen})p(x_{rec}\,|\,x_{gen})\;dx_{gen}$$

$$\approx \int p_\theta(x_{gen})p_\phi(x_{rec}\,|\,x_{gen})\;dx_{gen}$$

Replace simulation with surrogate NN

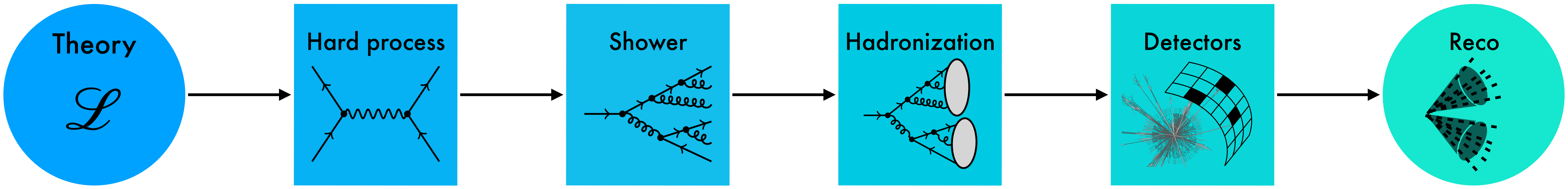
$$\approx \sum_{i=1}^{N_{MC}} p_\theta(x_{i,gen})p_\phi(x_{rec}\,|\,x_{i,gen})$$

Monte-Carlo approximation of the integral

This equation is for one individual data point.
When training the network we have to calculate this for each data point in each iteration.
To get a reasonable MC approximation we have to draw $\mathcal{O}(100)$ samples per data point

Requires large GPUs to train

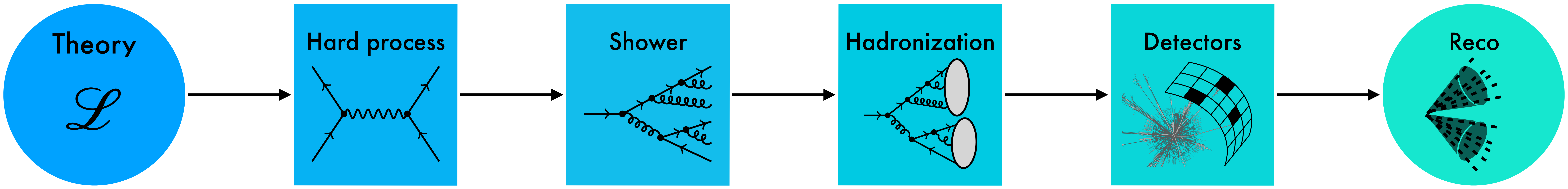
Revisiting the problem one last time



$$\int dx_{gen} \quad p(x_{gen}) \quad \underbrace{p(x_{reco} \,|\, x_{gen})} \quad = \quad p(x_{reco})$$

The task is to find the generation level distribution $p(x_{rec})$ that gave rise to the observed distribution $p(x_{rec})$

Revisiting the problem one last time



$$\int dx_{gen} \quad p(x_{gen}) \quad p(x_{reco}\,|\,x_{gen}) \quad = \quad p(x_{reco})$$

The task is to find [illegible] level distribution $p(x_{rec})$ that gave rise to the observed distribution $p(x_{rec})$

Solution is not unique

Requires ensemble of neural networks to map out the space of possible generation level distributions that could have given rise to the observed reconstruction level data

Transfer-based Unfolding

2) Calculate $p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$ using the forward detector kernel $p(x_{rec} | x_{gen})$

$$p_\theta(x_{rec}) = \int p_\theta(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$$

$$\approx \int p_\theta(x_{gen}) p_\phi(x_{rec} | x_{gen}) \, dx_{gen}$$

Replace simulation with surrogate NN

$$\approx \sum_{i=1}^{N_{MC}} p_\theta(x_{i,gen}) p_\phi(x_{rec} | x_{i,gen})$$

Monte-Carlo approximation of the integral

This equation is for one individual data point.
When training the network we have to calculate this for each data point in each iteration.
To get a reasonable MC approximation we have to draw $\mathcal{O}(100)$ samples per data point

Train $\mathcal{O}(30)$ networks in parallel to map out the possible solution space

Requires very large GPUs to train

Moving on to Cosmology!

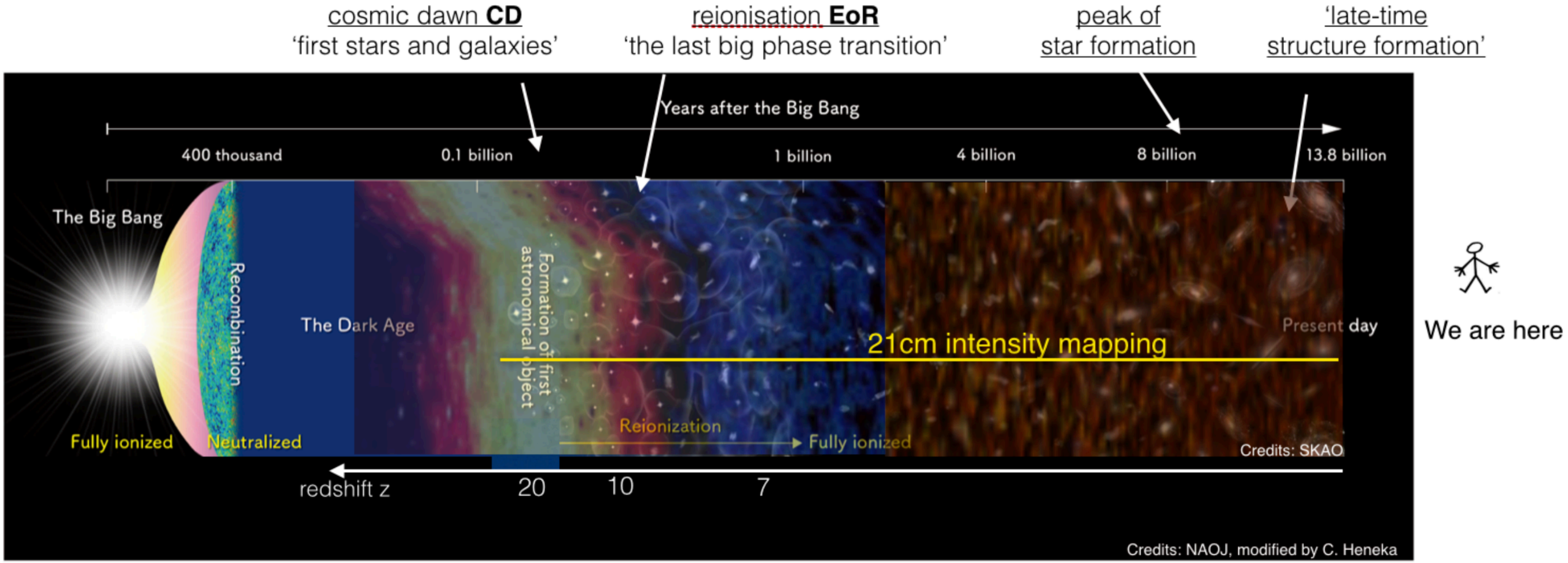
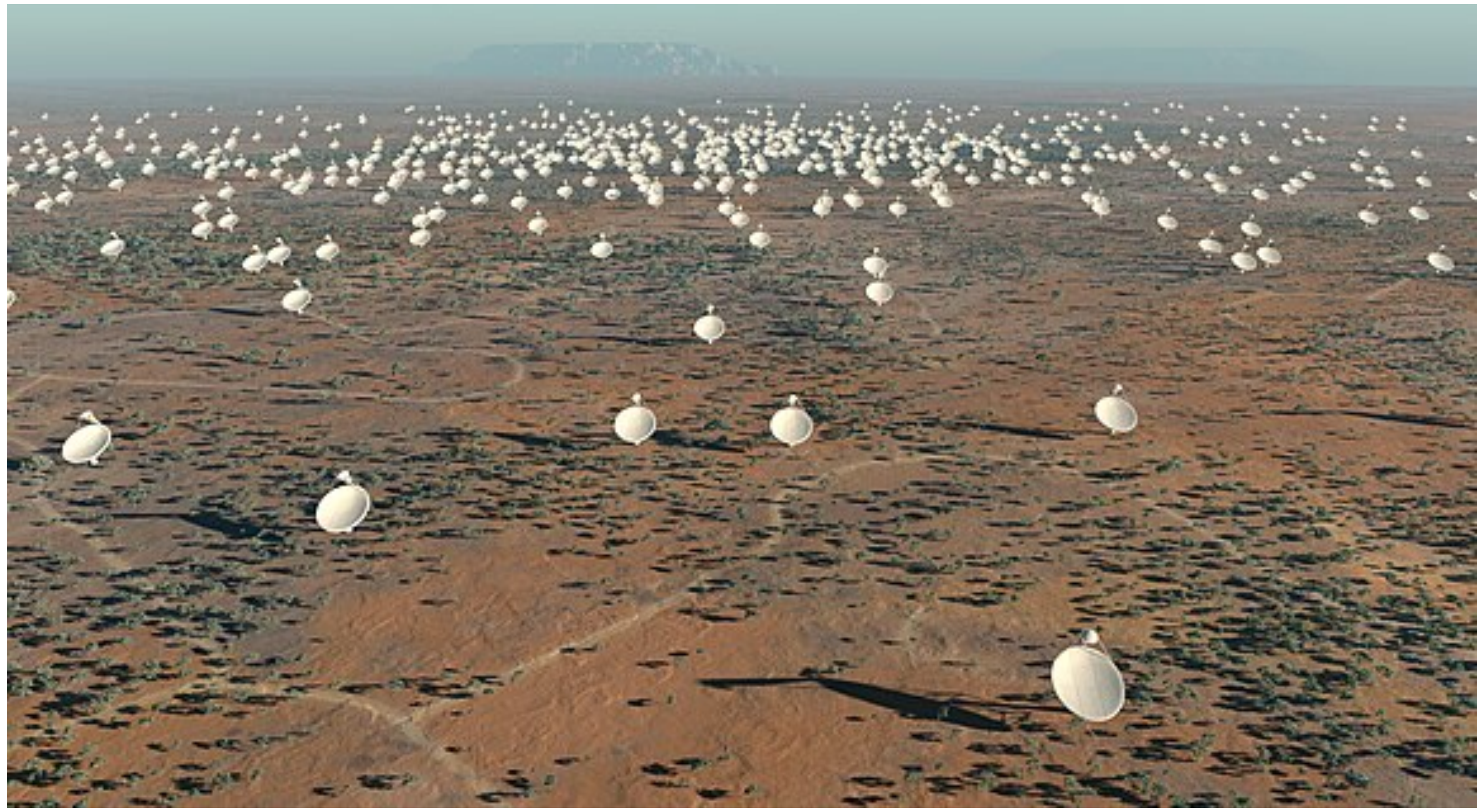


Figure credit: Caroline Heneka **https://indico.nikhef.nl/event/4875/contributions/20257/attachments/8256/11861/EuCAIFcon-Heneka.pdf**

Square Kilometre Array (SKA)



https://en.wikipedia.org/wiki/Square_Kilometre_Array

International telescope project currently being built

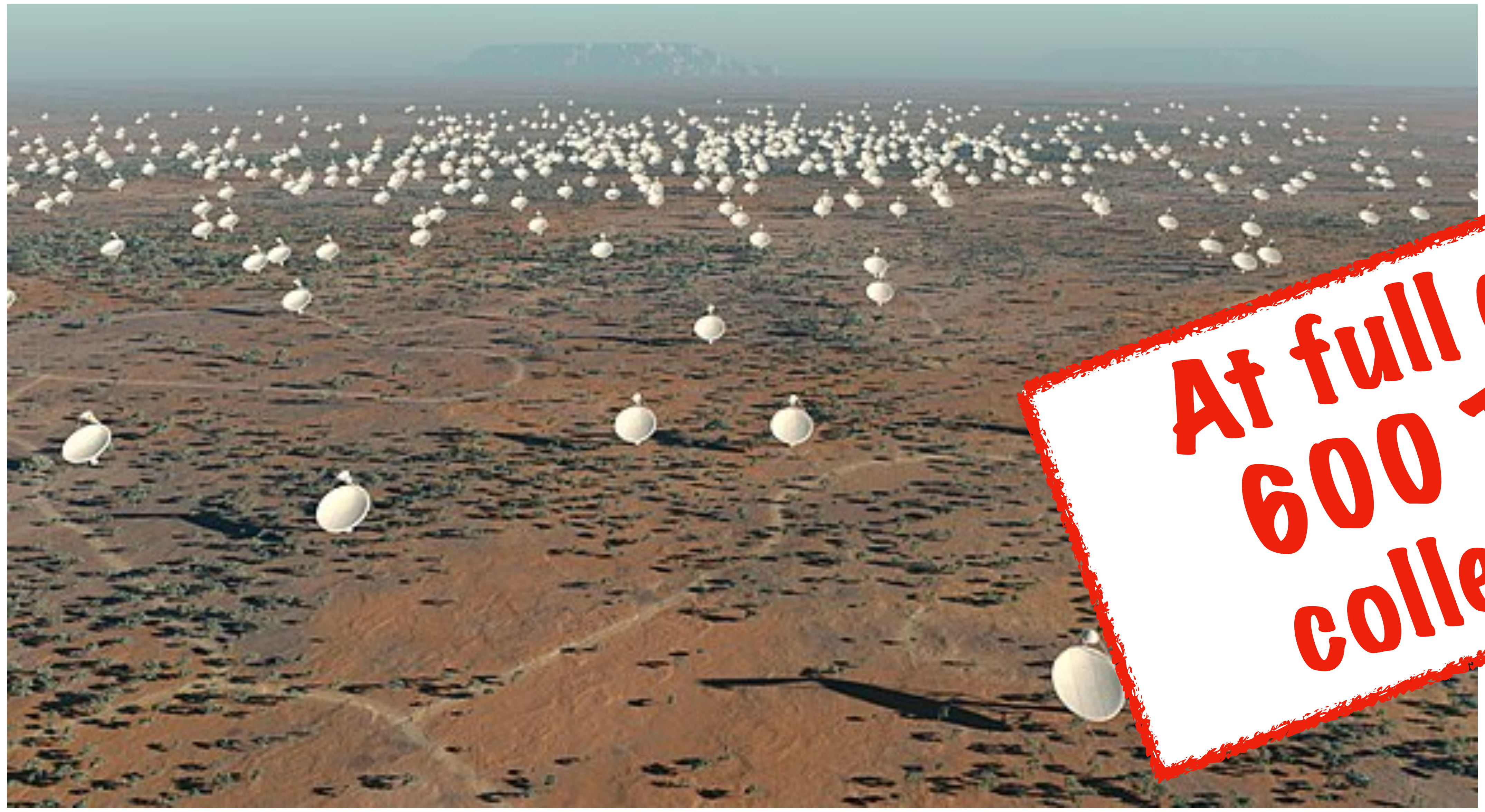
Expected to start collecting data in the mid 2020s

Will provide the highest-resolution astronomy images ever collected at a much higher frequency than all previous telescopes

Will allow us to observe the Dark Ages for the first time

Will (hopefully) improve our understanding of galaxy evolution, cosmological structure formation, the thermal history of the Universe …

Square Kilometre Array (SKA)



https://en.wikipedia.org/wiki/Square_Kilometre_Array

International telescope project currently being built

Exp[illegible]llecting data in the mid 2020s

[illegible]est-resolution astronomy images ever
[illegible]gher frequency than all previous telescopes

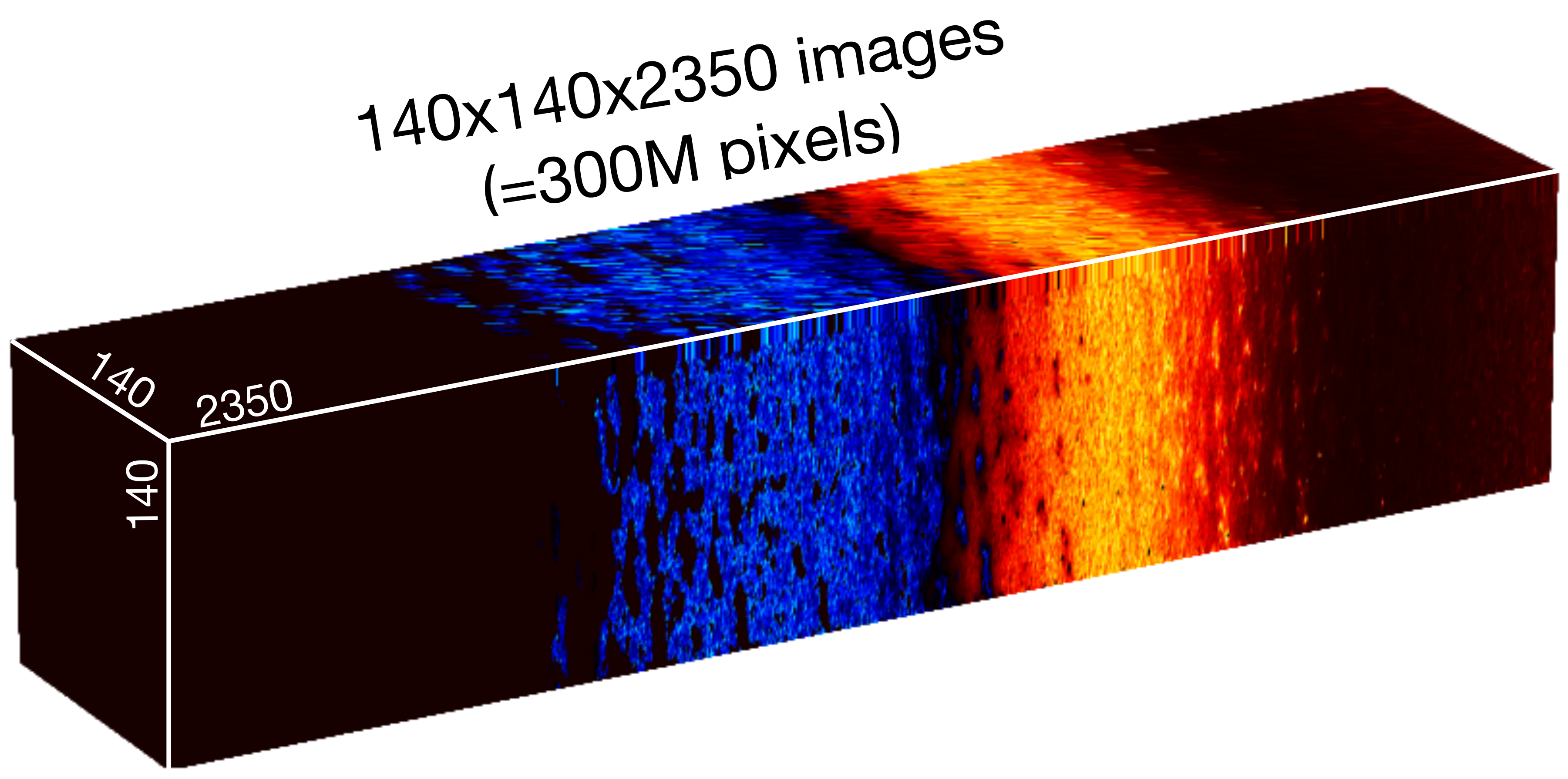
[illegible]us to observe the Dark Ages for the first time

Will (hopefully) improve our understanding of galaxy evolution, cosmological structure formation, the thermal history of the Universe …

At full capacity: 600 TB/s data collection rate

ML for SKA images [Credit to Ayo Ore]

Images recorded by the SKA will contain > 100 M pixels



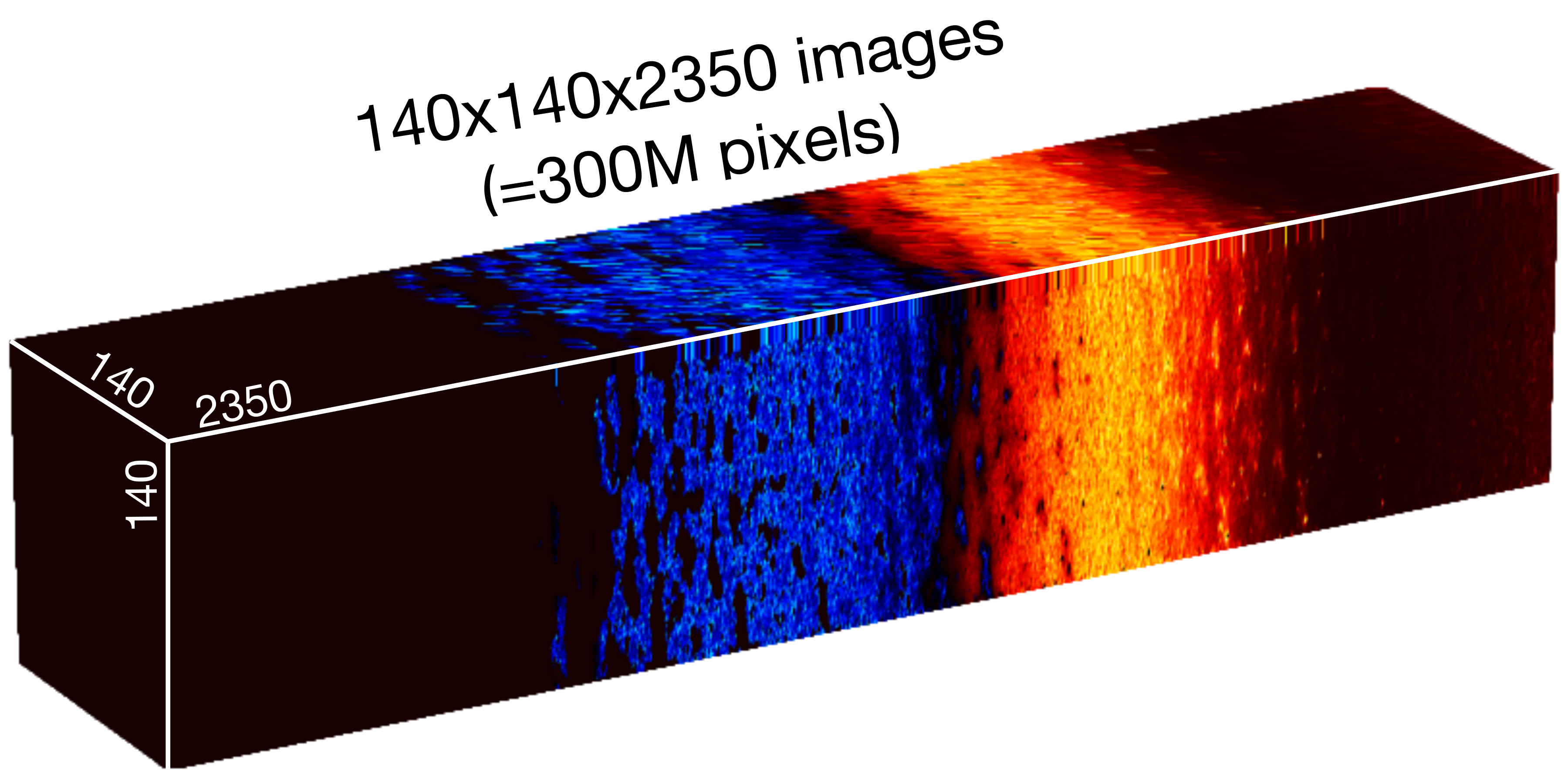
ML for SKA images [Credit to Ayo Ore]

Images recorded by the SKA will contain > 100 M pixels

Statistical analysis in such high dimension is intractable…
No choice but to compress the information

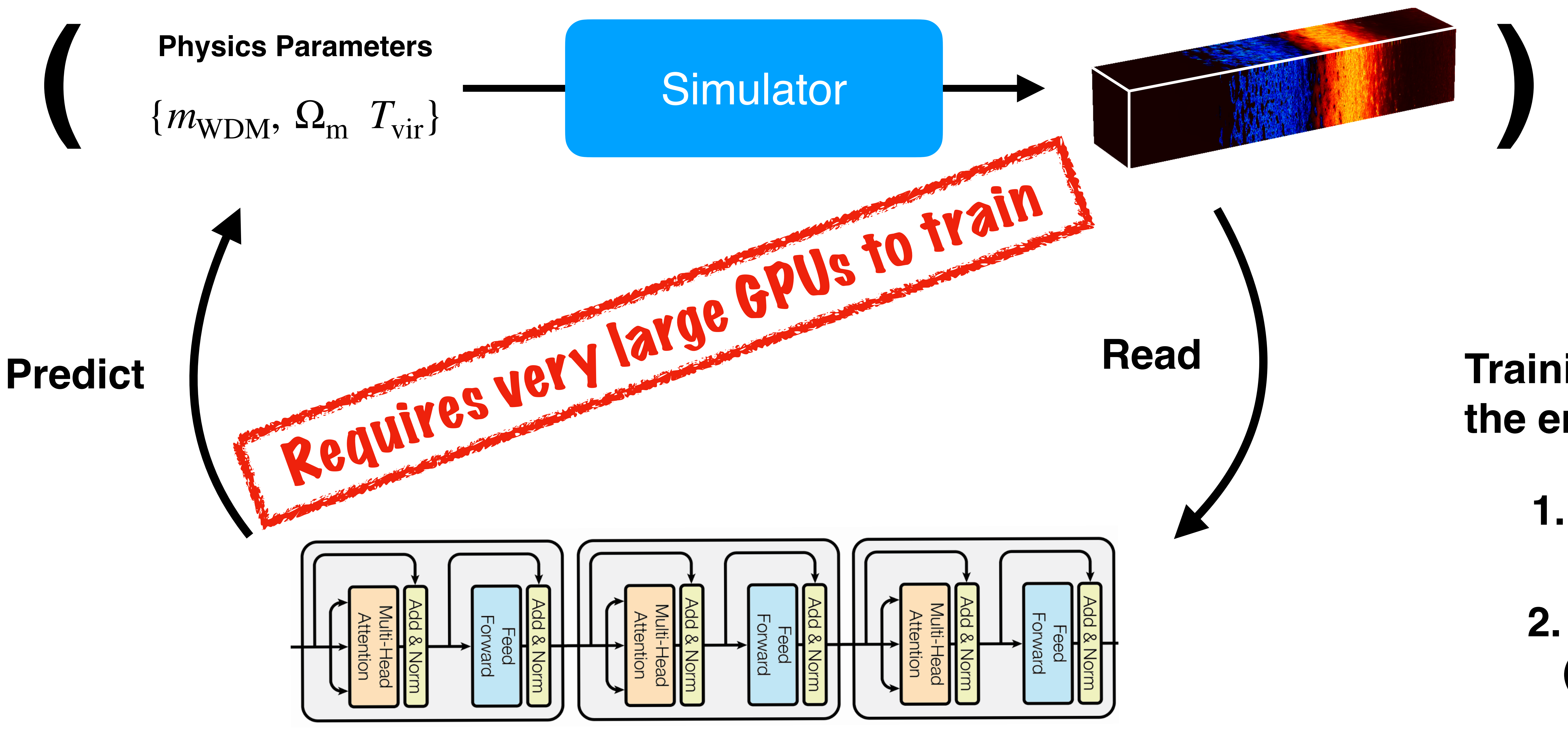
A classic summary method is the power spectrum, also used to study the CMB, but it is known to not be optimal for SKA images.

With ML, one can replace hand crafted summaries by learned representations.



ML with SKA images [Credit to Ayo Ore]

Many images are required to train neural networks, but high-resolution simulations are slow and large



x5000 = 1TB

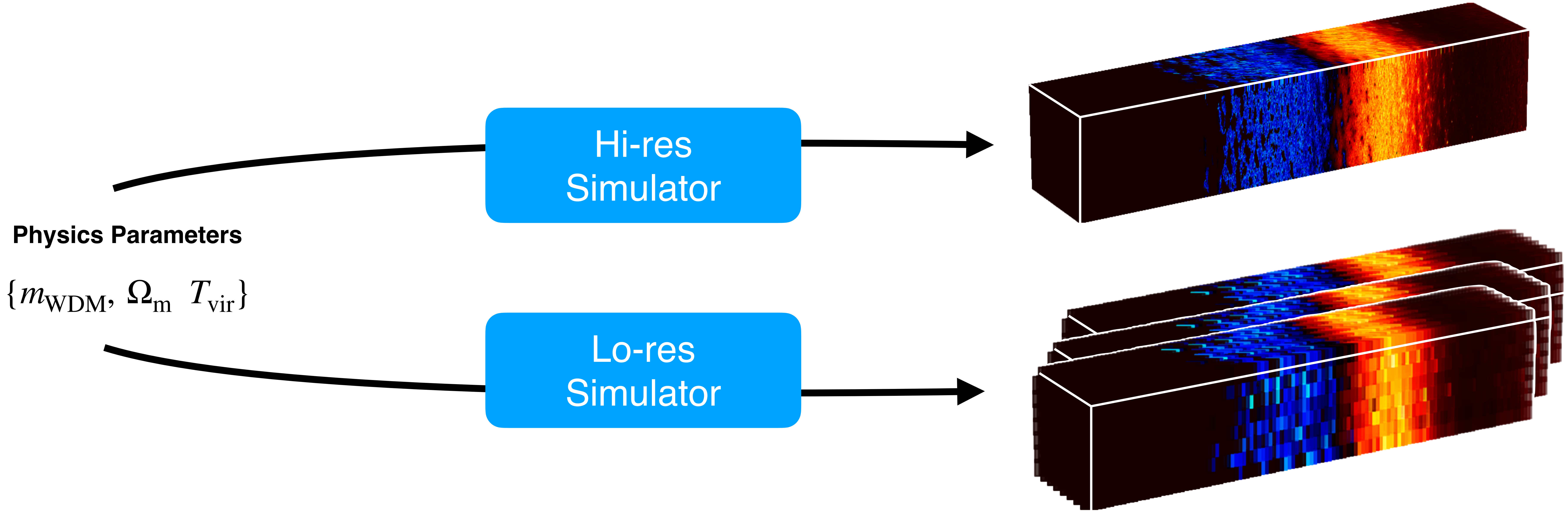
Transformer Neural Network
(~1M parameters)

Training consists of many iterations over the entire dataset, requiring either

1. **Large memory**
   (dataset stored in RAM)
2. **Fast disk speed**
   (reading in chunks every epoch)

Foundation models for SKA [Credit to Ayo Ore]

Simulation quality can be exchanged with speed



Goal: Leverage large volumes of lo-res images in order to improve performance at hi-res

Approach:
1. **Pre-train a large neural network to summarize lo-res images**
2. **Fine-tune the network using small hi-res dataset**

Requires very large GPUs to train

Summary and Outlook

Fundamental physics is entering into the big-data era

ML allows to scale established methods up to this

ML allows the development of completely new analysis tools that get rid of previous approximations and simplifications

Some methods have been used in practice for a while now (event taggers), some are moving from proof-of-concept to deployment now (unfolding)

Use bigger GPUs

Convince more people in physics that ML is cool

Convince more people in computing that physics is cool