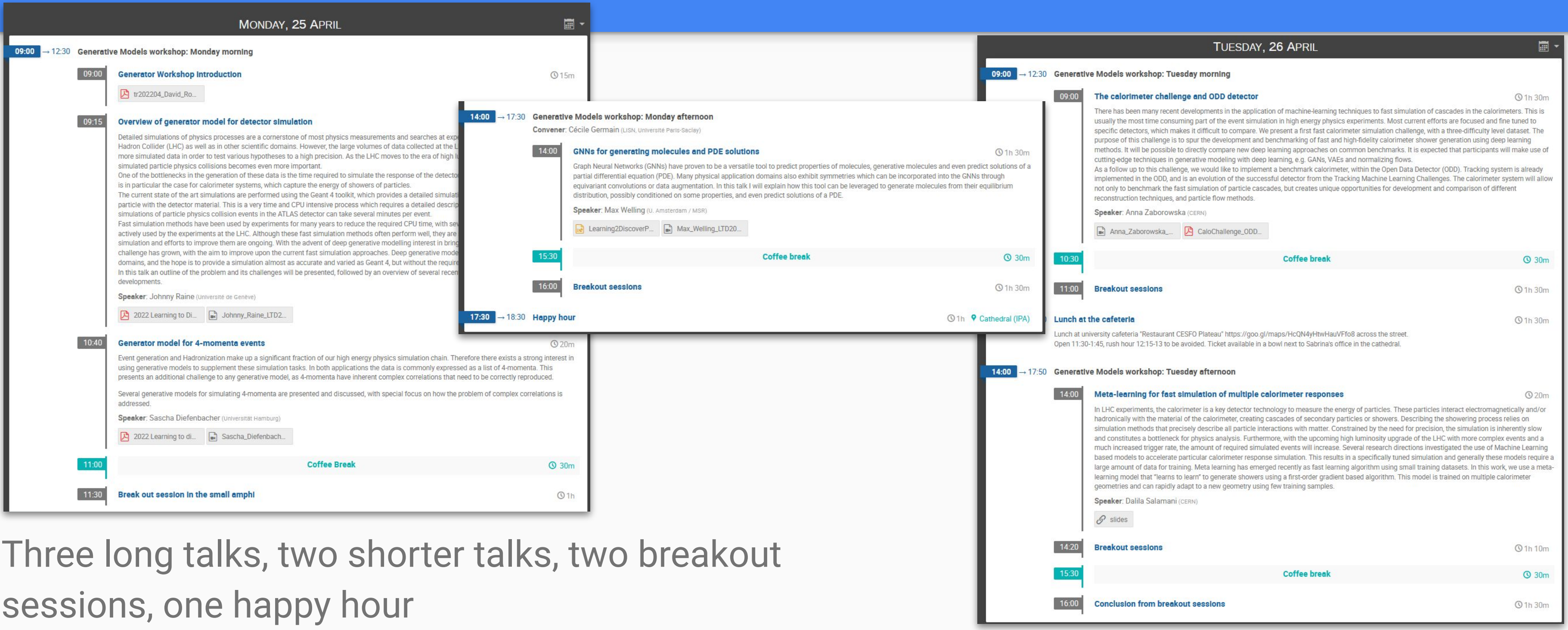
Summary of Generative Modelling Workshop

Learning to Discover, Paris 2022

Johnny Raine, University of Geneva

Overview



MONDAY, 25 APRIL

09:00 → 12:30 Generative Models workshop: Monday morning

09:00 Generator Workshop Introduction ⏱15m

tr202204_David_Ro...

09:15 Overview of generator model for detector simulation

Detailed simulations of physics processes are a cornerstone of most physics measurements and searches at expe
Hadron Collider (LHC) as well as in other scientific domains. However, the large volumes of data collected at the L
more simulated data in order to test various hypotheses to a high precision. As the LHC moves to the era of high lu
simulated particle physics collisions becomes even more important.
One of the bottlenecks in the generation of these data is the time required to simulate the response of the detecto
is in particular the case for calorimeter systems, which capture the energy of showers of particles.
The current state of the art simulations are performed using the Geant 4 toolkit, which provides a detailed simulati
particle with the detector material. This is a very time and CPU intensive process which requires a detailed descrip
simulations of particle physics collision events in the ATLAS detector can take several minutes per event.
Fast simulation methods have been used by experiments for many years to reduce the required CPU time, with sev
actively used by the experiments at the LHC. Although these fast simulation methods often perform well, they are
simulation and efforts to improve them are ongoing. With the advent of deep generative modelling interest in bring
challenge has grown, with the aim to improve upon the current fast simulation approaches. Deep generative mode
domains, and the hope is to provide a simulation almost as accurate and varied as Geant 4, but without the require
In this talk an outline of the problem and its challenges will be presented, followed by an overview of several recen
developments.

Speaker: Johnny Raine (Université de Genève)

2022 Learning to Di... | Johnny_Raine_LTD2...

10:40 Generator model for 4-momenta events ⏱20m

Event generation and Hadronization make up a significant fraction of our high energy physics simulation chain. Therefore there exists a strong interest in using generative models to supplement these simulation tasks. In both applications the data is commonly expressed as a list of 4-momenta. This presents an additional challenge to any generative model, as 4-momenta have inherent complex correlations that need to be correctly reproduced.

Several generative models for simulating 4-momenta are presented and discussed, with special focus on how the problem of complex correlations is addressed.

Speaker: Sascha Diefenbacher (Universität Hamburg)

2022 Learning to di... | Sascha_Diefenbach...

11:00 Coffee Break ⏱ 30m

11:30 Break out session in the small amphi ⏱1h

14:00 → 17:30 Generative Models workshop: Monday afternoon

Convener: Cécile Germain (LISN, Université Paris-Saclay)

14:00 GNNs for generating molecules and PDE solutions ⏱1h 30m

Graph Neural Networks (GNNs) have proven to be a versatile tool to predict properties of molecules, generative molecules and even predict solutions of a partial differential equation (PDE). Many physical application domains also exhibit symmetries which can be incorporated into the GNNs through equivariant convolutions or data augmentation. In this talk I will explain how this tool can be leveraged to generate molecules from their equilibrium distribution, possibly conditioned on some properties, and even predict solutions of a PDE.

Speaker: Max Welling (U. Amsterdam / MSR)

Learning2DiscoverP... | Max_Welling_LTD20...

15:30 Coffee break ⏱ 30m

16:00 Breakout sessions ⏱1h 30m

17:30 → 18:30 Happy hour ⏱1h Cathedral (IPA)

TUESDAY, 26 APRIL

09:00 → 12:30 Generative Models workshop: Tuesday morning

09:00 The calorimeter challenge and ODD detector ⏱1h 30m

There has been many recent developments in the application of machine-learning techniques to fast simulation of cascades in the calorimeters. This is usually the most time consuming part of the event simulation in high energy physics experiments. Most current efforts are focused and fine tuned to specific detectors, which makes it difficult to compare. We present a first fast calorimeter simulation challenge, with a three-difficulty level dataset. The purpose of this challenge is to spur the development and benchmarking of fast and high-fidelity calorimeter shower generation using deep learning methods. It will be possible to directly compare new deep learning approaches on common benchmarks. It is expected that participants will make use of cutting-edge techniques in generative modeling with deep learning, e.g. GANs, VAEs and normalizing flows.
As a follow up to this challenge, we would like to implement a benchmark calorimeter, within the Open Data Detector (ODD). Tracking system is already implemented in the ODD, and is an evolution of the successful detector from the Tracking Machine Learning Challenges. The calorimeter system will allow not only to benchmark the fast simulation of particle cascades, but creates unique opportunities for development and comparison of different reconstruction techniques, and particle flow methods.

Speaker: Anna Zaborowska (CERN)

Anna_Zaborowska_... | CaloChallenge_ODD...

10:30 Coffee break ⏱ 30m

11:00 Breakout sessions ⏱1h 30m

Lunch at the cafeteria ⏱1h 30m

Lunch at university cafeteria "Restaurant CESFO Plateau" https://goo.gl/maps/HcQN4yHtwHauVFfo8 across the street.
Open 11:30-1:45, rush hour 12:15-13 to be avoided. Ticket available in a bowl next to Sabrina's office in the cathedral.

14:00 → 17:50 Generative Models workshop: Tuesday afternoon

14:00 Meta-learning for fast simulation of multiple calorimeter responses ⏱20m

In LHC experiments, the calorimeter is a key detector technology to measure the energy of particles. These particles interact electromagnetically and/or hadronically with the material of the calorimeter, creating cascades of secondary particles or showers. Describing the showering process relies on simulation methods that precisely describe all particle interactions with matter. Constrained by the need for precision, the simulation is inherently slow and constitutes a bottleneck for physics analysis. Furthermore, with the upcoming high luminosity upgrade of the LHC with more complex events and a much increased trigger rate, the amount of required simulated events will increase. Several research directions investigated the use of Machine Learning based models to accelerate particular calorimeter response simulation. This results in a specifically tuned simulation and generally these models require a large amount of data for training. Meta learning has emerged recently as fast learning algorithm using small training datasets. In this work, we use a meta-learning model that "learns to learn" to generate showers using a first-order gradient based algorithm. This model is trained on multiple calorimeter geometries and can rapidly adapt to a new geometry using few training samples.

Speaker: Dalila Salamani (CERN)

slides

14:20 Breakout sessions ⏱1h 10m

15:30 Coffee break ⏱ 30m

16:00 Conclusion from breakout sessions ⏱1h 30m

Three long talks, two shorter talks, two breakout sessions, one happy hour

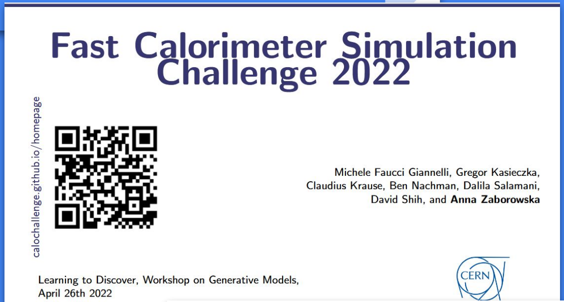
Topics

1. Detector simulation
2. Four-vector simulation
3. Molecules
4. PDE Solving

Detector Simulation

Generative Modelling for Detector Simulation

Learning to Discover, Paris 2022
Johnny Raine, University of Geneva

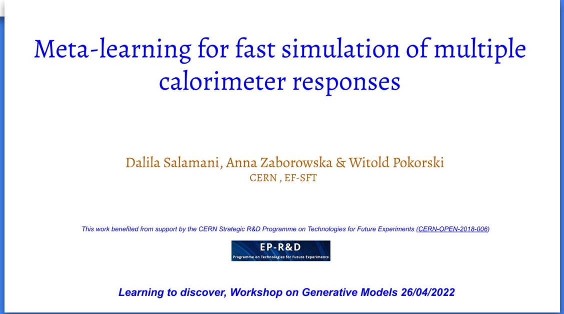


Fast Calorimeter Simulation Challenge 2022

calochallenge.github.io/homepage

Michele Faucci Giannelli, Gregor Kasieczka,
Claudius Krause, Ben Nachman, Dalila Salamani,
David Shih, and **Anna Zaborowska**

Learning to Discover, Workshop on Generative Models,
April 26th 2022



Meta-learning for fast simulation of multiple calorimeter responses

Dalila Salamani, Anna Zaborowska & Witold Pokorski
CERN , EF-SFT

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (CERN-OPEN-2018-006)

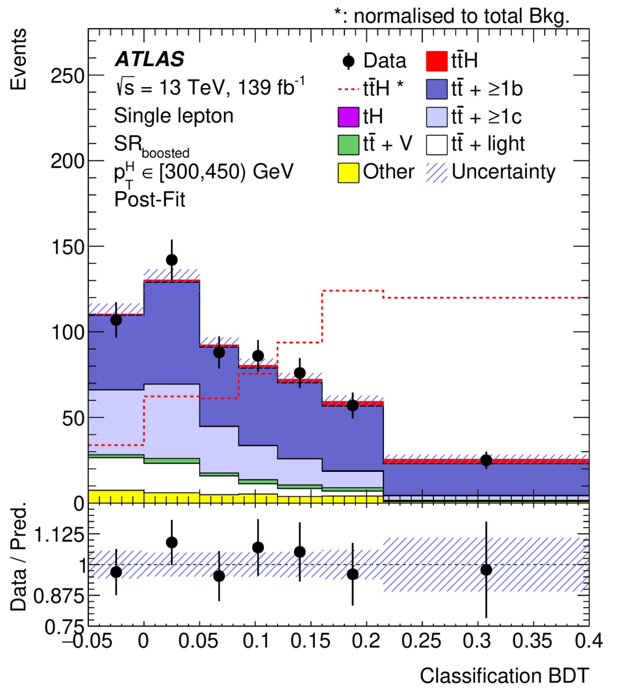
EP-R&D
Programme on Technologies for Future Experiments

Learning to discover, Workshop on Generative Models 26/04/2022

Introduction

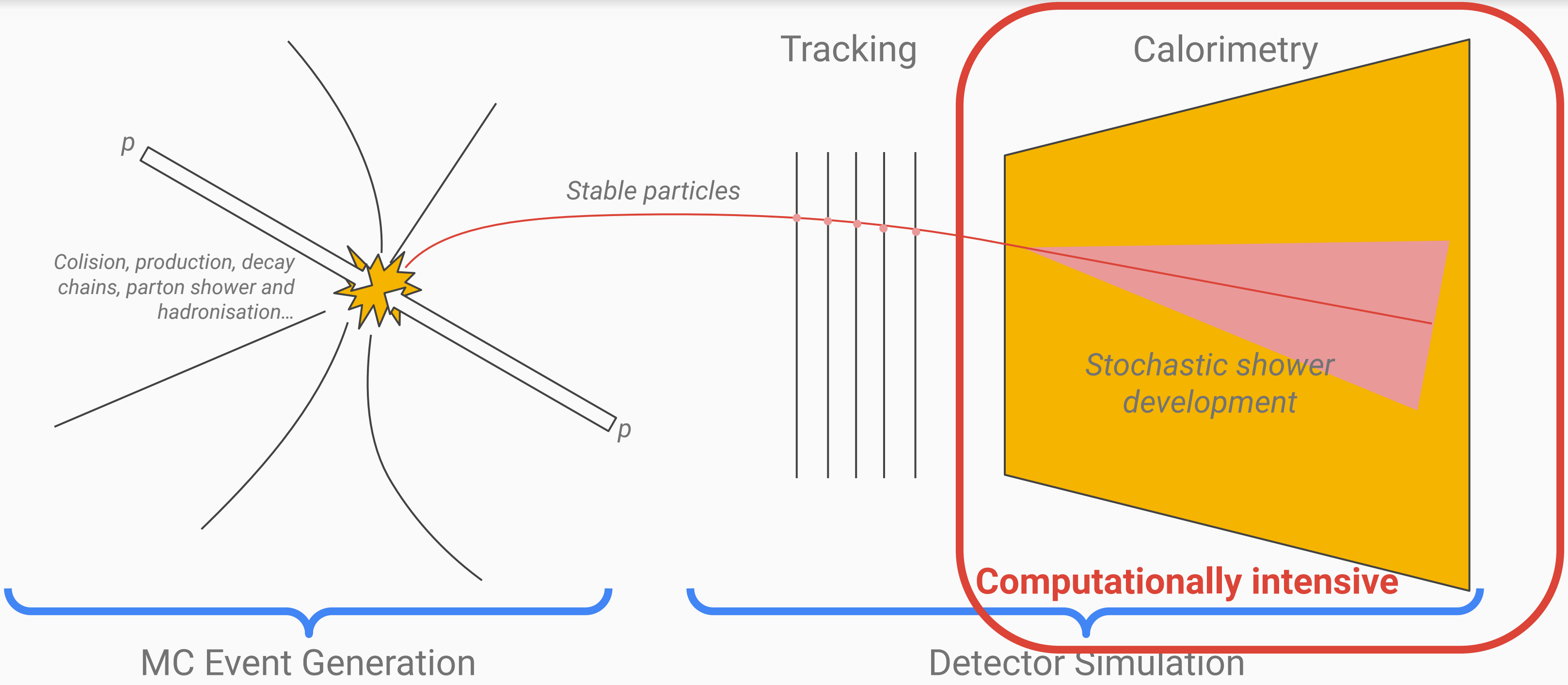
Key part of a physics analyses is to have a reference

- Does data fit Hypothesis 1 or Hypothesis 2?
  - Amount of signal on top of background
  - Deviation away from expectation
- Need accurate simulation of underlying processes->recorded data in most settings



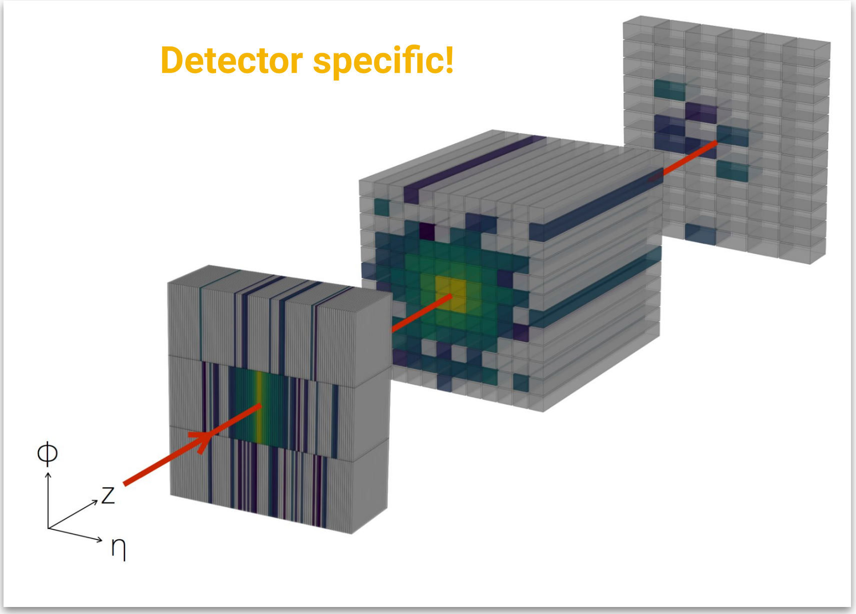
[1]

What happens in a particle collision


p
Colision, production, decay chains, parton shower and hadronisation...
p
Stable particles
Tracking
Calorimetry
Stochastic shower development
Computationally intensive
MC Event Generation
Detector Simulation

Calorimeter Output

- Individual particle traverses detector
- Energy deposited in many cells from secondary particles in shower
- Can build an “image” but
  - **High dynamic range** of “pixels”
  - Often very **sparse**
  - **Stochastic** - same incoming particle results in different shower
- Assess performance with complex distributions over many events

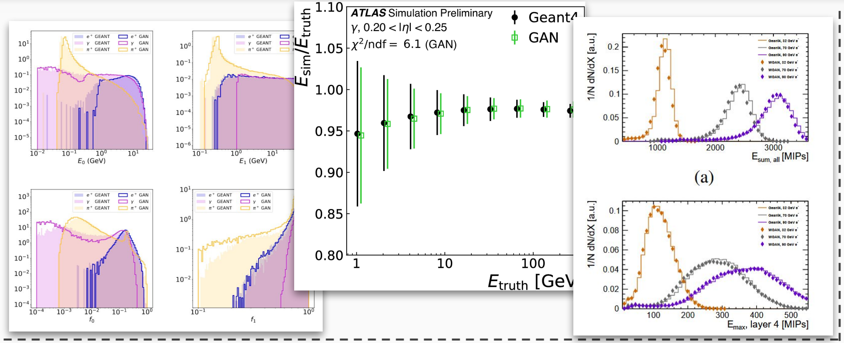


[2]

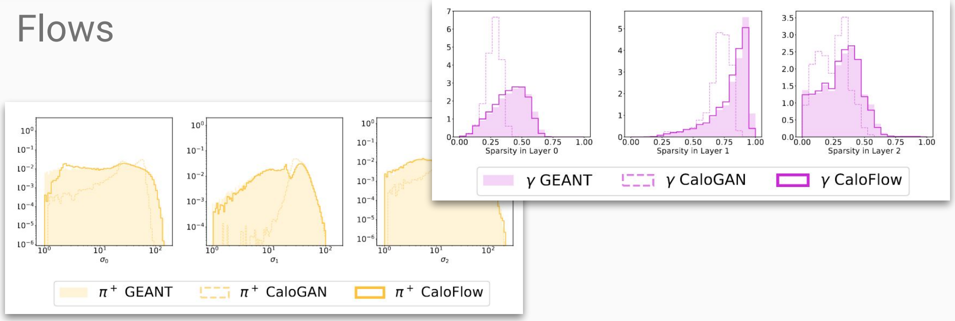
Paganini et al (2017), Krause, Shih (2021),
ATLAS Collaboration (2019, 2019),
Erdmann et al (2018)

Wide range of models

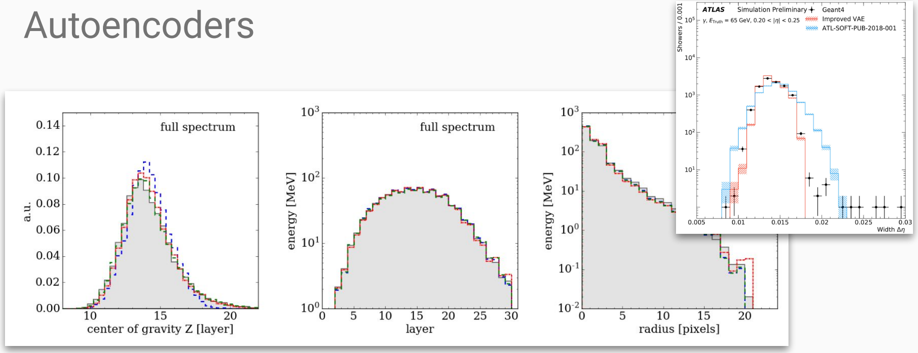
GANs

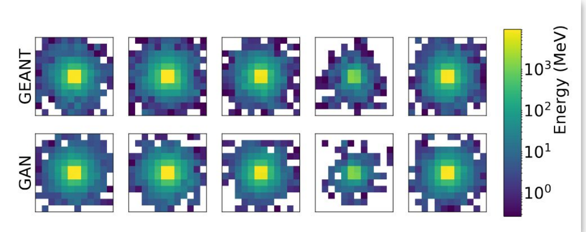


Flows



Autoencoders

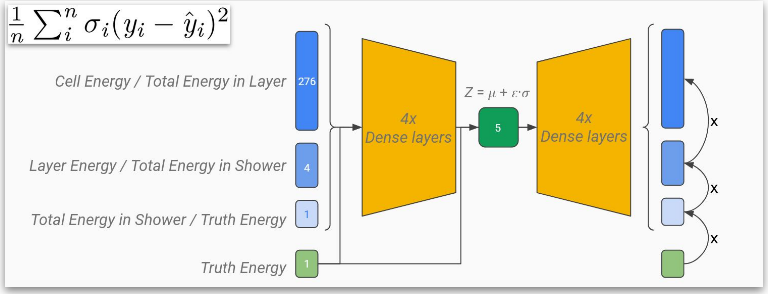




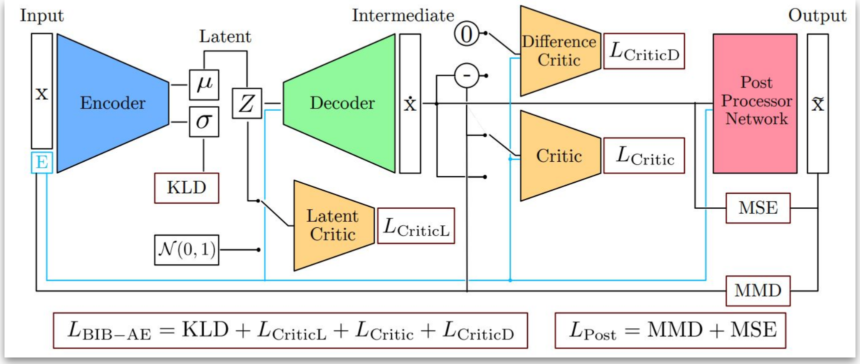
Performance measured over lots of distributions

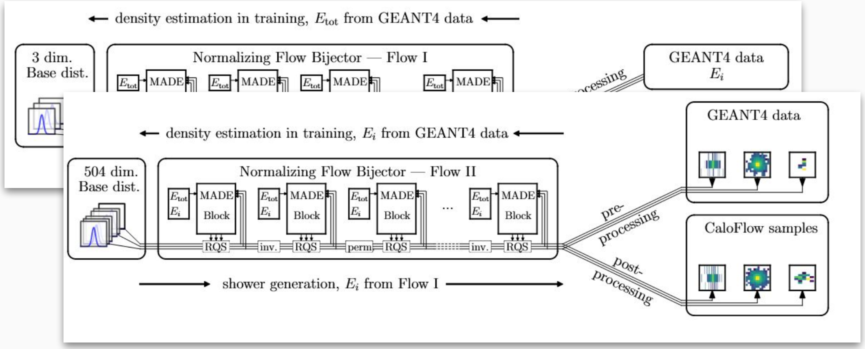
- Impossible to judge individual showers

Wide range of models of varying complexity



$\frac{1}{n}\sum_i^n \sigma_i(y_i - \hat{y}_i)^2$
Cell Energy / Total Energy in Layer
276
Layer Energy / Total Energy in Shower
4
Total Energy in Shower / Truth Energy
1
Truth Energy
1
4x Dense layers
$Z = \mu + \varepsilon \cdot \sigma$
5
4x Dense layers
x
x
x





density estimation in training, E_{tot} from GEANT4 data
3 dim. Base dist.
Normalizing Flow Bijector — Flow I
E_{tot}
MADE
GEANT4 data E_i
processing
density estimation in training, E_i from GEANT4 data
504 dim. Base dist.
Normalizing Flow Bijector — Flow II
E_{tot}
E_i
MADE Block
RQS
inv.
perm.
...
GEANT4 data
pre-processing
post-processing
CaloFlow samples
shower generation, E_i from Flow I

But lots of common themes

- Focus on better modelling of total energy
  - Additional critic/constrainer networks
  - Auxilliary term in loss
- Widths and sparsity of data is a challenge!

Adapted from A Zaborowska

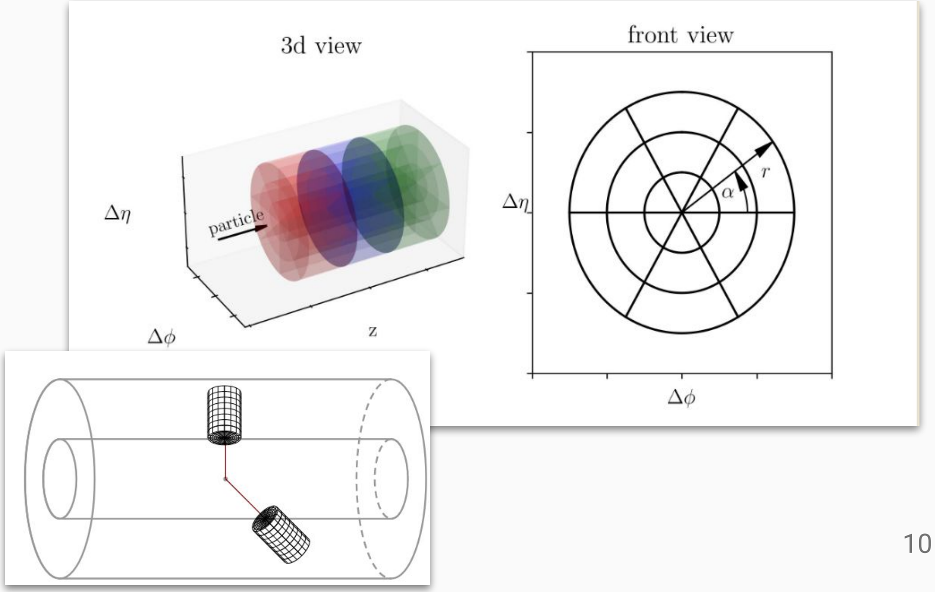
Calorimeter Challenge

Lots of models but not easy to compare - different detectors, different datasets

- Challenge: 3 datasets for benchmarking
- Same format, different granularity
- Increasing complexity through granularity
  - Dim. O(500), O(6000), O(40k)

Dataset 1: ATLAS open data, detector, layers

Dataset 2+3: Simplistic geometry, but 45 layers



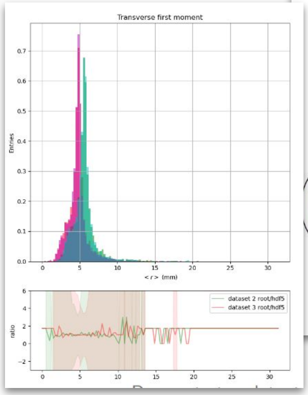
https://calochallenge.github.io/homepage/

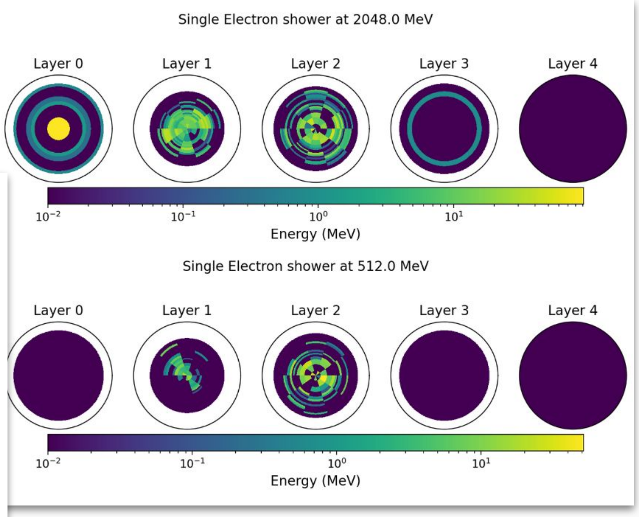
Adapted from A Zaborowska

Calorimeter Challenge

Lots of models but not easy to compare - different detectors, different datasets

- Provide **common** code for evaluating **metrics**, plots
- Inference framework for fair **timing comparison**
- Three granularities, additional slices of detector available





Input data challenges

Sparsity of deposits and reproducing high dynamic range still a tough ask!

Getting tails of distributions hard

- But often key descriptors for underlying physics

Need to accurately learn the PDF of showers including the stochasticity

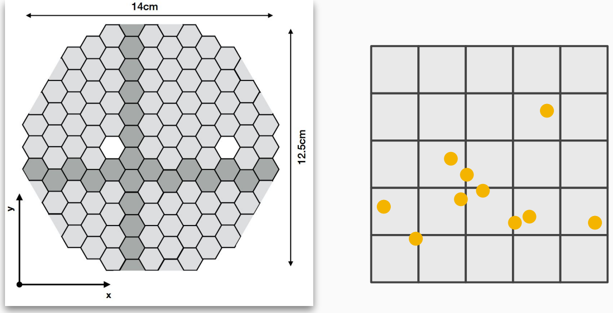
- How to enhance focus on reconstruction of tails but keep this property
- Only a handful of conditional parameters - particle four vector

Detectors not designed to be easy to simulate!

Data format

Not restricted to using readout or detector

- Use more natural representation?



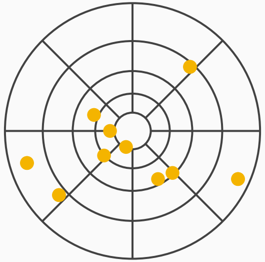
Work with detector cells

Learn in needed domain, granularity

Doesn’t generalise with consistent geometry
Geometry often nontrivial

Showers produce

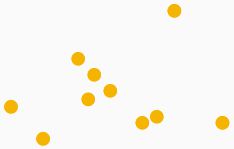
- Variable number of energy deposits
- Correlated spatial ordering but no natural ordering



Map hits to polar grid

Shower evolution cone shaped

To remove edge effects need fine, sparse, binning

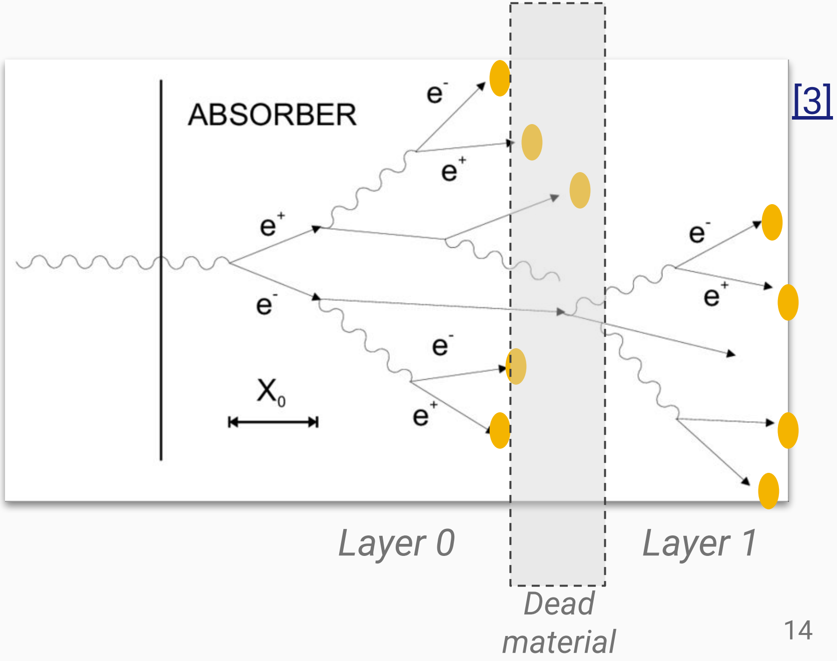


Point cloud

Direct translation of input data
Use graph networks!
Preserve natural symmetries

Brainstorming discussion - Data

- Are we hiding information from the networks?
    - Don't utilise shower development
    - Only look at energy in active material
- How to use low level information
    - Inspiration from molecules?
    - Project into higher/lower dim space?

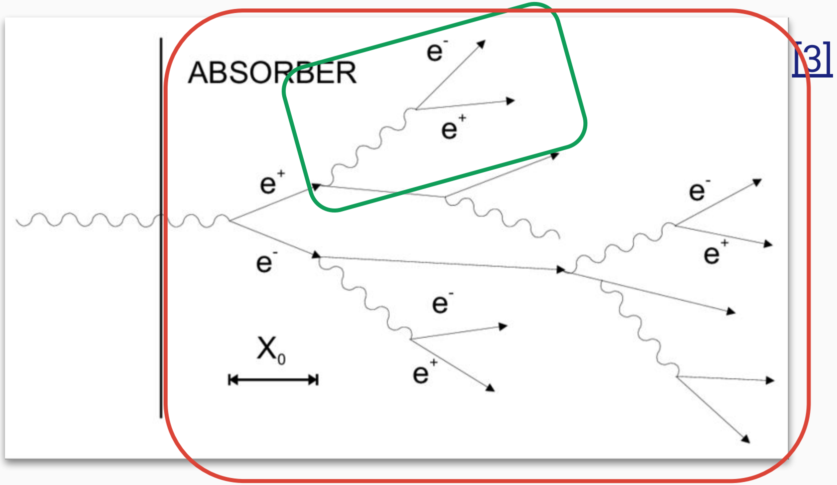


[3]

Brainstorming discussion - Data

- Why not exploit underlying physics development in model design

- Characteristic of shower depends on particle and energy
  - Not what happened before
  - Could somehow exploit in development of shower?



[3]

Brainstorming - Getting into production

- What do we need to generate - can focus too much on one thing!
  - Different particles, regions, physics of interest mandate differences
  - Avoid being fooled you're doing well looking just at one distribution
- Scalability
  - Can we handle higher dimensions, validation
  - Can overshoot needed granularity - more margin for error
- Aim instead to fix faster simulation, don't replace
  - **Multi-fidelity simulation**
- How to **transfer** to more **regions/particles/detectors**

Adapted from D Salamani

Meta-Learning

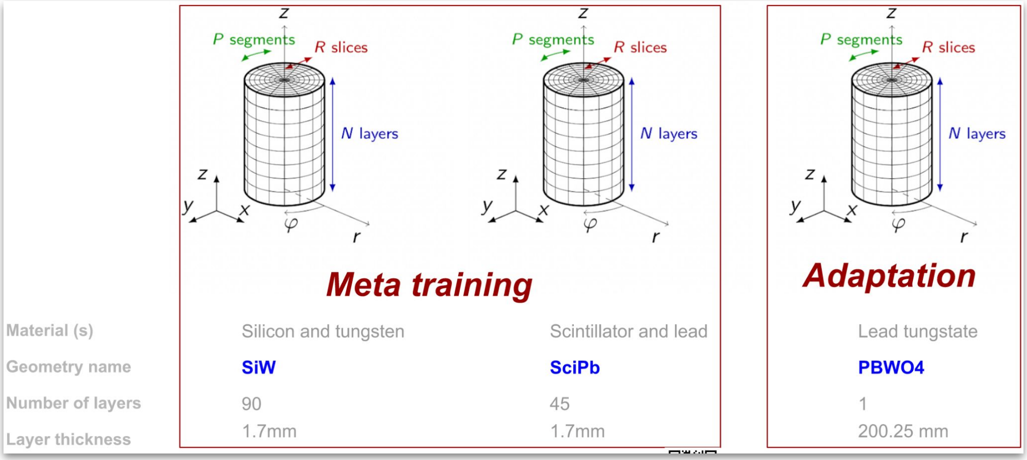
How easily can we adapt for different detector materials?

Autoencoder model, fixed input/output description

- **Generalised description of a shower**

Train on one composition, apply to another

- Provide initial general solution
- Can it save training time with adaptation?



	Meta training		Adaptation
Material (s)	Silicon and tungsten	Scintillator and lead	Lead tungstate
Geometry name	SiW	SciPb	PBWO4
Number of layers	90	45	1
Layer thickness	1.7mm	1.7mm	200.25 mm

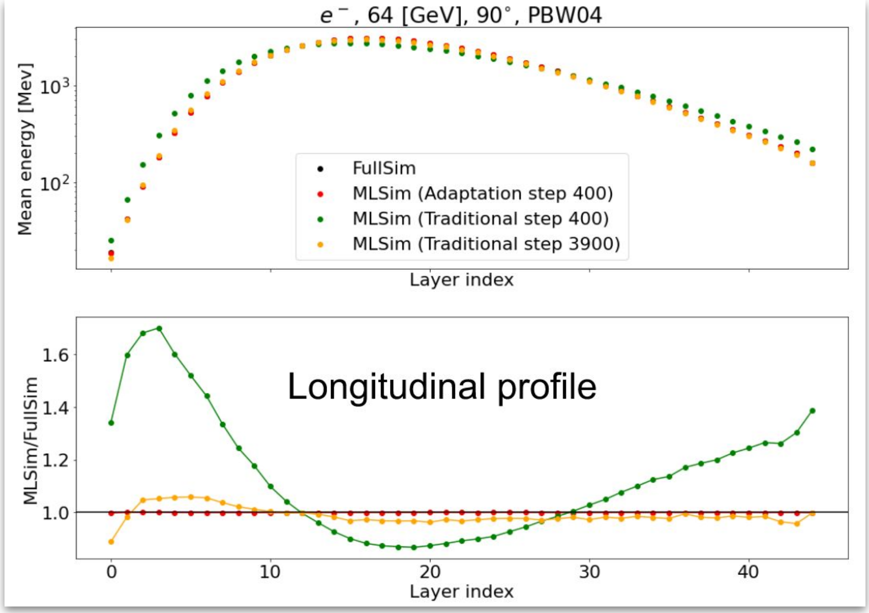
Adapted from D Salamani

Meta-Learning

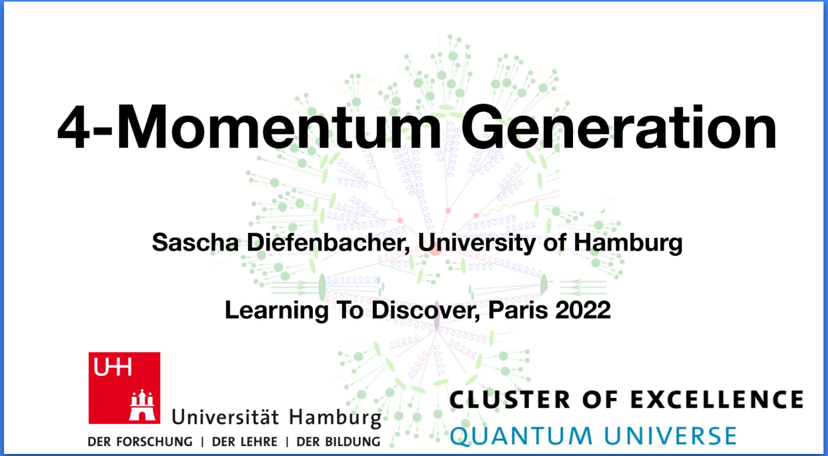
Compare

- Transfer learning w/ 400 epochs
- 400 epochs from scratch
- 3900 epochs from scratch

In generalised shower representation much faster convergence
(order seconds!)



Four-vector Simulation



4-Momentum Generation

Sascha Diefenbacher, University of Hamburg

Learning To Discover, Paris 2022

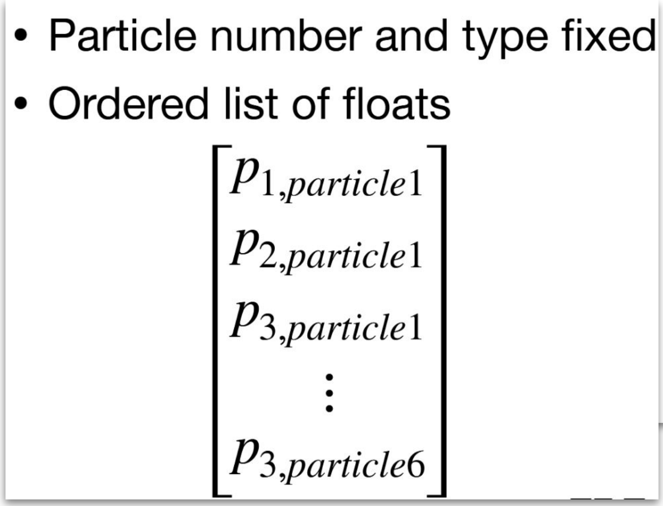
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

Adapted from S Diefenbacher

Four vector simulation

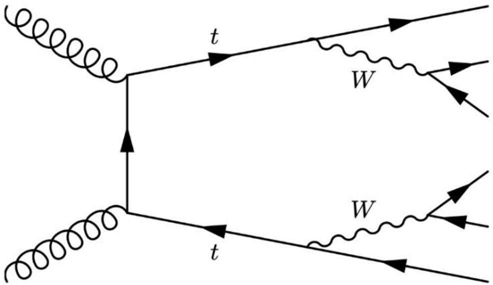
- Look to replace MC generation of underlying physics processes
- Output four momenta of particles
- Studied with
  - GANs
  - Flows



- Particle number and type fixed
- Ordered list of floats

$$\begin{bmatrix} p_{1,particle1} \\ p_{2,particle1} \\ p_{3,particle1} \\ \vdots \\ p_{3,particle6} \end{bmatrix}$$

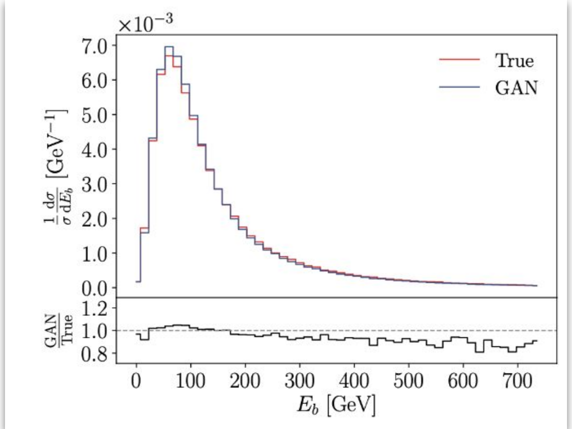
$$pp \to t\bar{t} \to (bW^-)\ (\bar{b}W^+) \to (bq_1\bar{q}_1')\ (\bar{b}q_2\bar{q}_2')$$

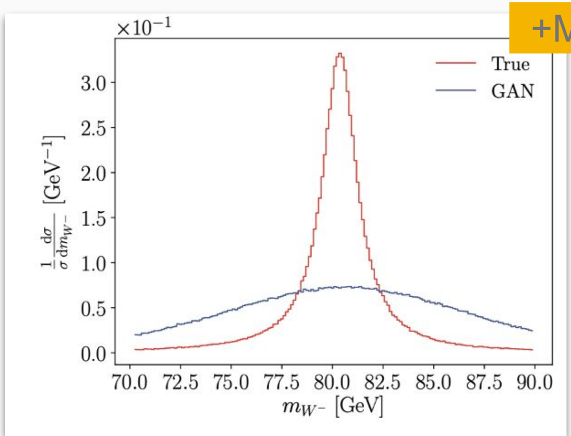


Adapted from S Diefenbacher

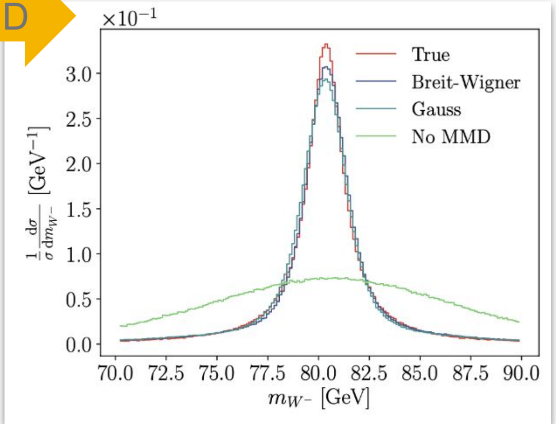
GANs

Using GANs gets direct targets well modelled but building correlated distributions too wide! Add MMD term





+MMD



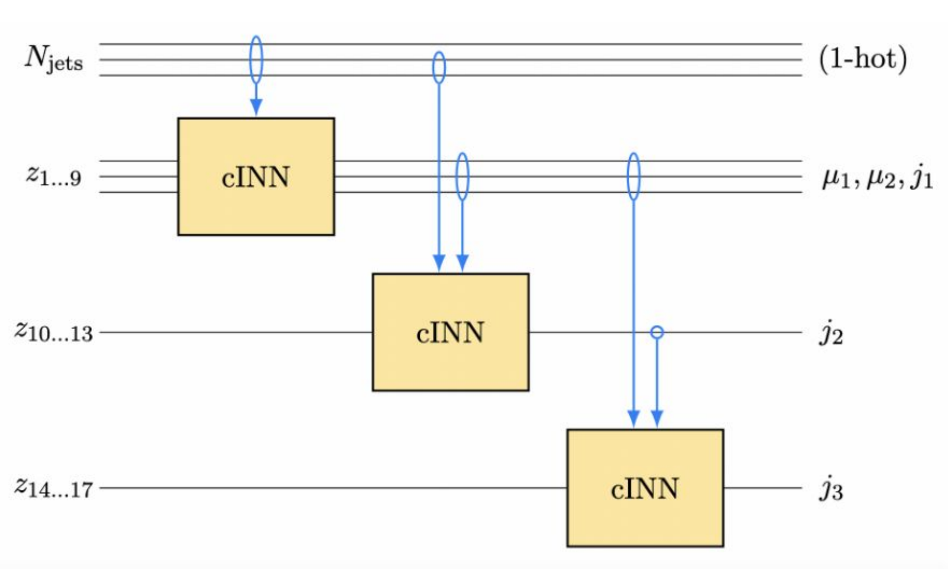
Adapted from S Diefenbacher

Flows

Well suited to low dimensionality, good at learning density dist. (read correlations!)

Flows have fixed dimensionality

- Want to Generate processes with variable multiplicities
- Be clever with architecture



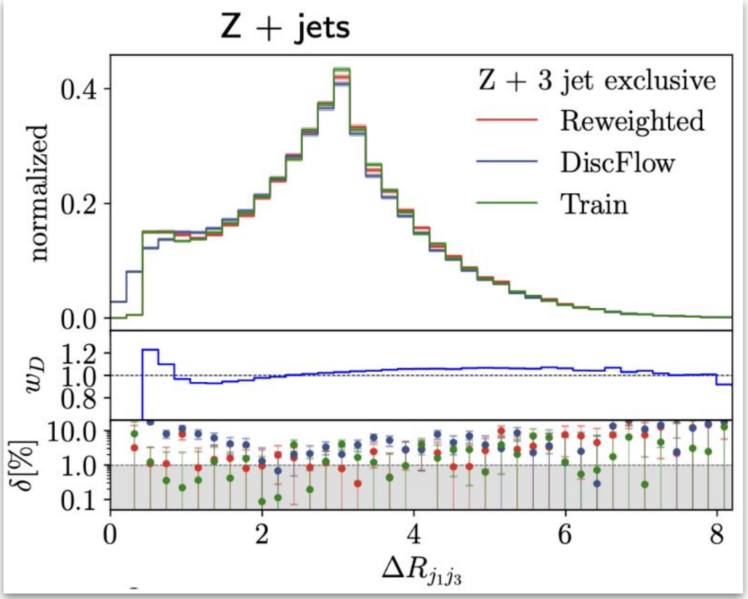
Adapted from S Diefenbacher

Flows

Well suited to low dimensionality, good at learning density dist. (read correlations!)

- Much better performance than GANs, but some residuals
- Correct with ML reweighting

Using Bayesian NNs doesn't capture mismodelling with an uncertainty!



Brainstorming

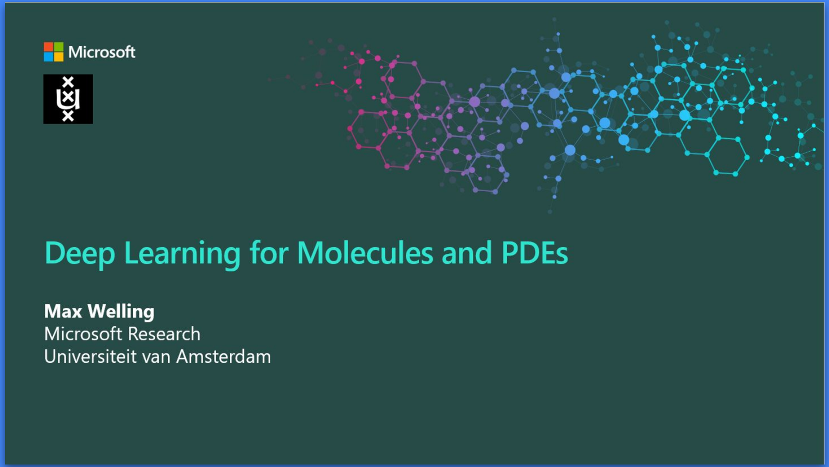
Many similarities across detector simulation and event generation

- Focus on wide range of distributions constructed from outputs
- Need to be smart with architecture design and losses!

How much do we need to replace, where can we speed up current implementations with ML

- Getting **final closure with reweighting** seen already with Flows, PDEs
- Speed up and **replace slow components** e.g. Vegas integration
- **MCMC with generative modelling** shown to be powerful

Molecule Generation



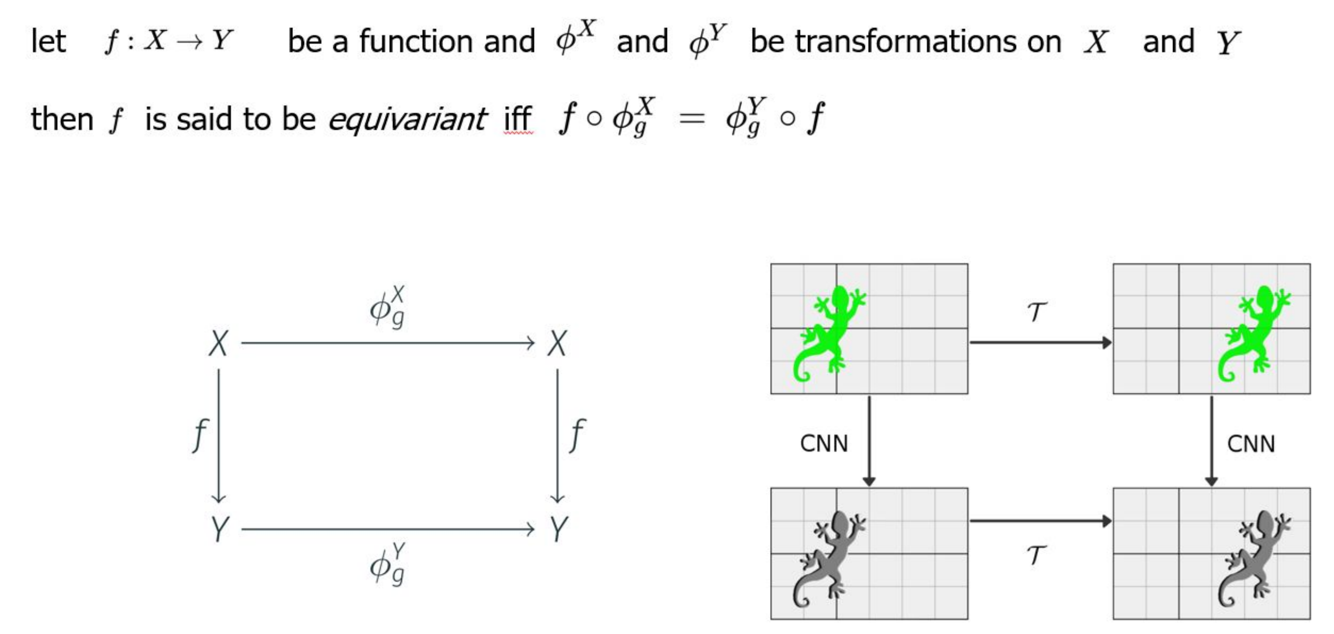
Microsoft

Deep Learning for Molecules and PDEs

Max Welling
Microsoft Research
Universiteit van Amsterdam

Adapted from M Welling

Equivariances in Networks



let $f : X \to Y$ be a function and ϕ^X and ϕ^Y be transformations on X and Y

then f is said to be *equivariant* iff $f \circ \phi_g^X = \phi_g^Y \circ f$

$$\begin{tikzcd}
X \arrow[r, "\phi_g^X"] \arrow[d, "f"'] & X \arrow[d, "f"] \\
Y \arrow[r, "\phi_g^Y"'] & Y
\end{tikzcd}$$

Adapted from M Welling

Equivariances in Networks

Updating nodes and rotating are equivarient operations
Need to make sure updates in GNN are equivariant

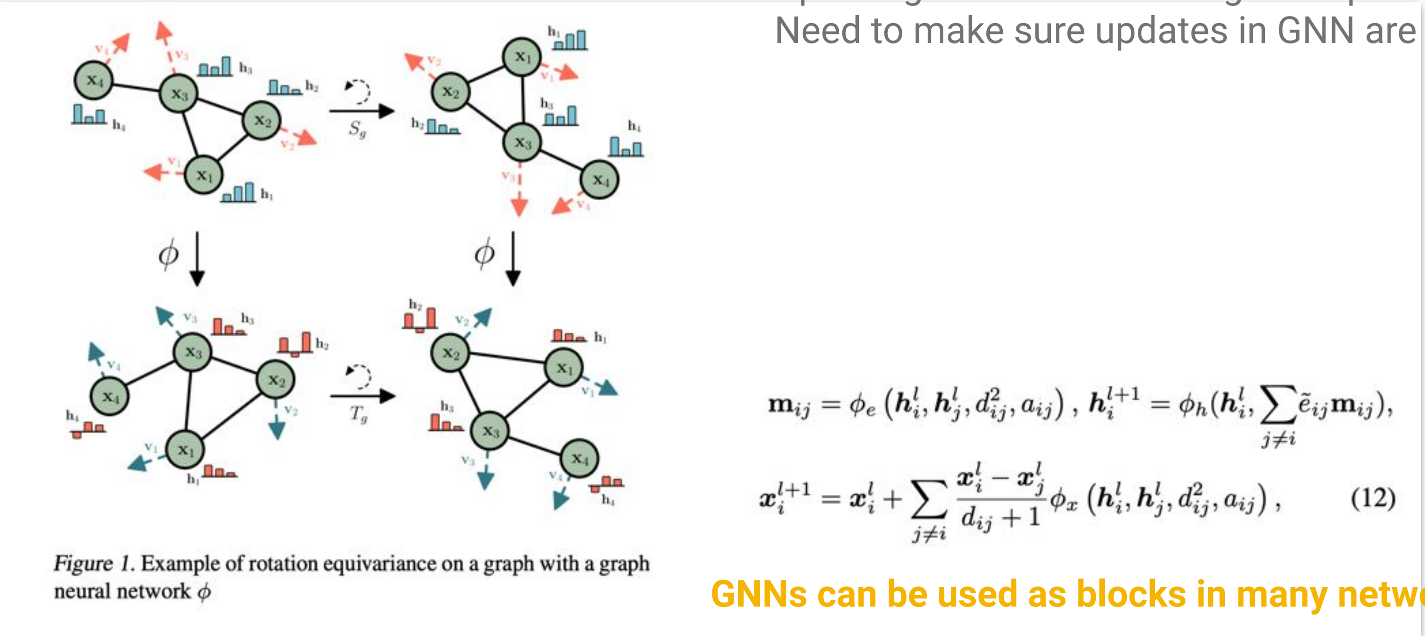


Figure 1. Example of rotation equivariance on a graph with a graph neural network ϕ

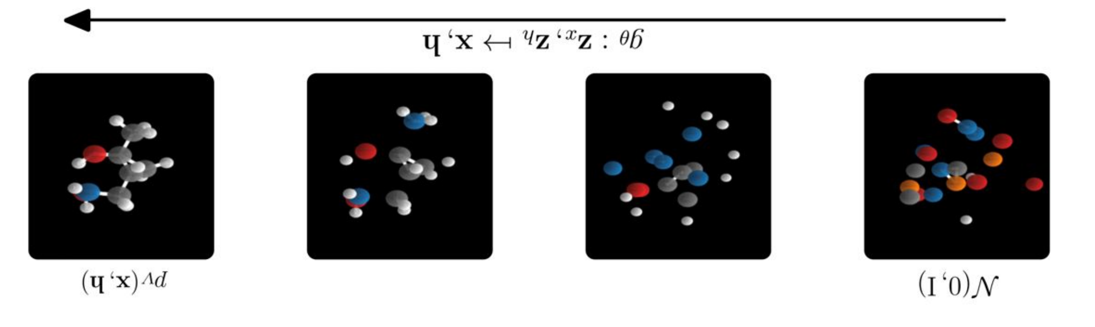
$$\mathbf{m}_{ij} = \phi_e\left(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, d_{ij}^2, a_{ij}\right), \; \boldsymbol{h}_i^{l+1} = \phi_h\Big(\boldsymbol{h}_i^l, \sum_{j\neq i} \tilde{e}_{ij}\mathbf{m}_{ij}\Big),$$

$$\boldsymbol{x}_i^{l+1} = \boldsymbol{x}_i^l + \sum_{j\neq i} \frac{\boldsymbol{x}_i^l - \boldsymbol{x}_j^l}{d_{ij}+1}\phi_x\left(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, d_{ij}^2, a_{ij}\right), \qquad (12)$$

GNNs can be used as blocks in many network architectures

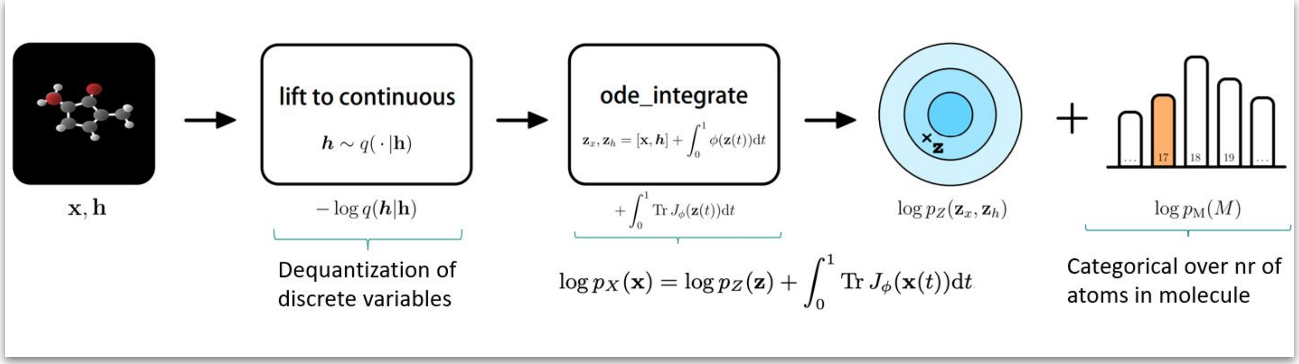
Adapted from M Welling

Molecule generation with equivariant flows


$g_\theta : \mathbf{z}_x, \mathbf{z}_h \to \mathbf{x}, \mathbf{h}$
$p_V(\mathbf{x}, \mathbf{h})$
$\mathcal{N}(0, \mathrm{I})$

Apply normalising flows to molecule generation

Transformation parametrised with equivariant operations


$\mathbf{x}, \mathbf{h}$
lift to continuous
$\boldsymbol{h} \sim q(\cdot|\mathbf{h})$
ode_integrate
$\mathbf{z}_x, \mathbf{z}_h = [\mathbf{x}, \boldsymbol{h}] + \int_0^1 \phi(\mathbf{z}(t))\mathrm{d}t$
$\mathbf{z}$
...
17
18
19
...
$-\log q(\boldsymbol{h}|\mathbf{h})$
$+\int_0^1 \mathrm{Tr}\, J_\phi(\mathbf{z}(t))\mathrm{d}t$
$\log p_Z(\mathbf{z}_x, \mathbf{z}_h)$
$+$
$\log p_{\mathrm{M}}(M)$

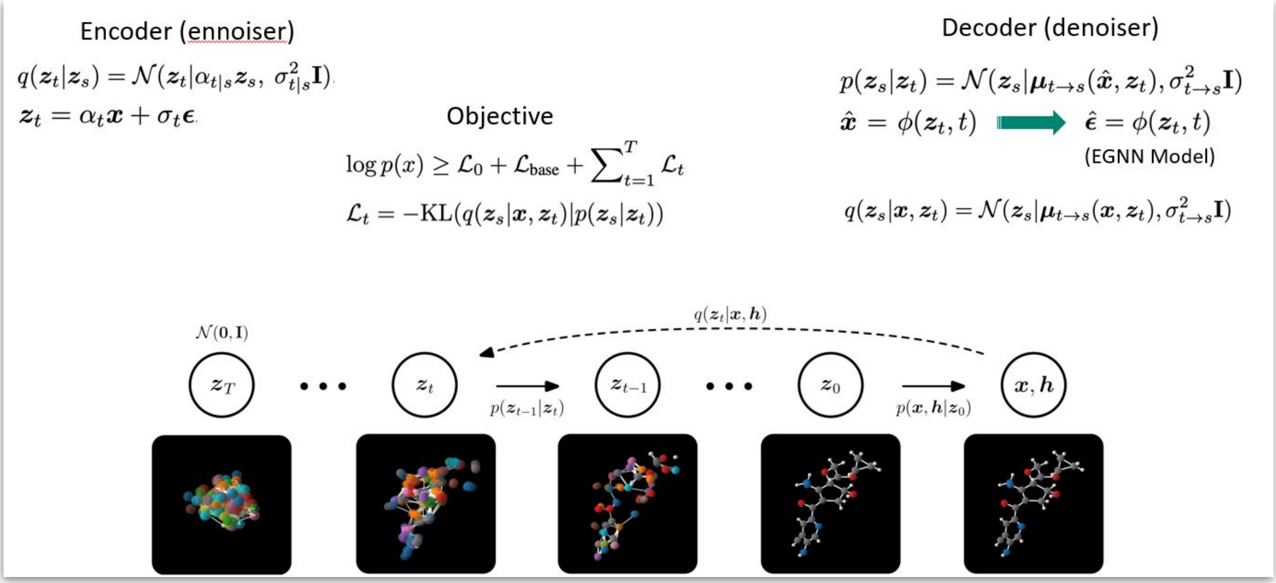
Dequantization of discrete variables

$$\log p_X(\mathbf{x}) = \log p_Z(\mathbf{z}) + \int_0^1 \mathrm{Tr}\, J_\phi(\mathbf{x}(t))\mathrm{d}t$$

Categorical over nr of atoms in molecule

Adapted from M Welling

Molecule generation with equivariant flows



Encoder (ennoiser)

$q(\boldsymbol{z}_t|\boldsymbol{z}_s) = \mathcal{N}(\boldsymbol{z}_t|\alpha_{t|s}\boldsymbol{z}_s,\ \sigma^2_{t|s}\mathbf{I})$

$\boldsymbol{z}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$

Objective

$$\log p(x) \geq \mathcal{L}_0 + \mathcal{L}_{\text{base}} + \sum\nolimits_{t=1}^{T} \mathcal{L}_t$$

$$\mathcal{L}_t = -\mathrm{KL}(q(\boldsymbol{z}_s|\boldsymbol{x}, \boldsymbol{z}_t)|p(\boldsymbol{z}_s|\boldsymbol{z}_t))$$

Decoder (denoiser)

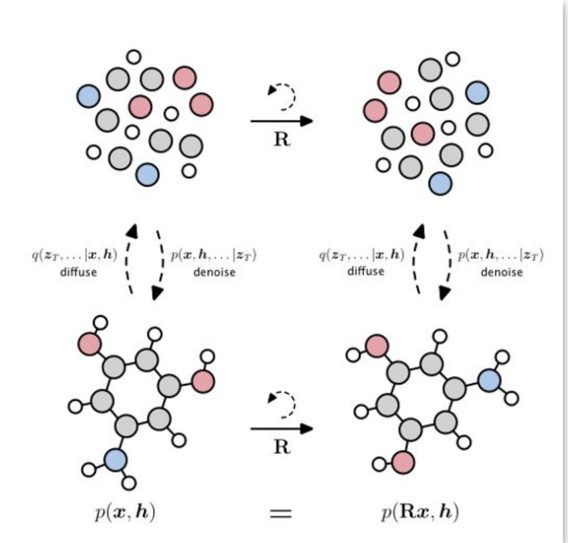
$p(\boldsymbol{z}_s|\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_s|\boldsymbol{\mu}_{t\to s}(\hat{\boldsymbol{x}}, \boldsymbol{z}_t), \sigma^2_{t\to s}\mathbf{I})$

$\hat{\boldsymbol{x}} = \phi(\boldsymbol{z}_t, t) \longrightarrow \hat{\boldsymbol{\epsilon}} = \phi(\boldsymbol{z}_t, t)$

(EGNN Model)

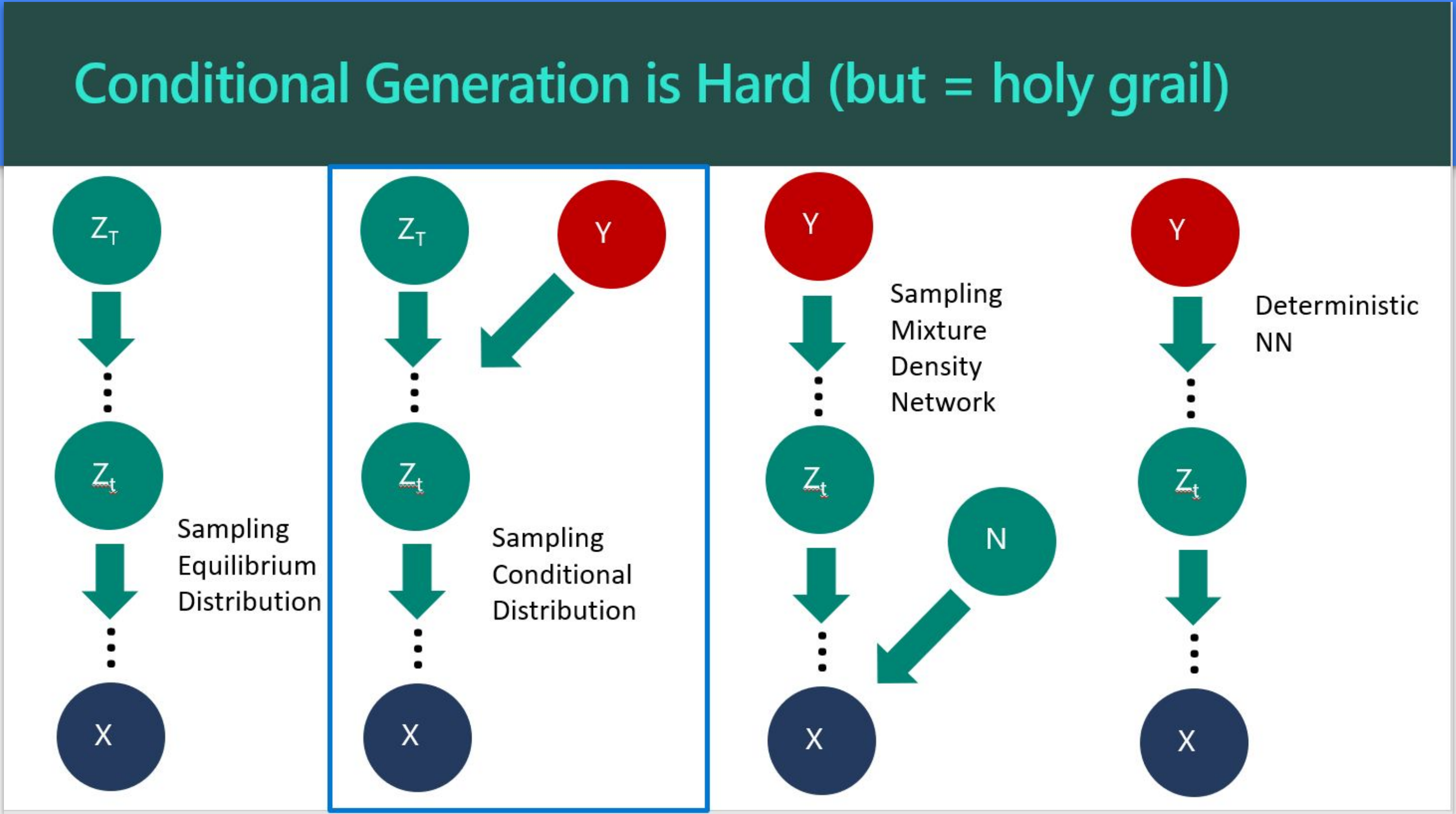
$q(\boldsymbol{z}_s|\boldsymbol{x}, \boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_s|\boldsymbol{\mu}_{t\to s}(\boldsymbol{x}, \boldsymbol{z}_t), \sigma^2_{t\to s}\mathbf{I})$

$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\boldsymbol{z}_T$
$\boldsymbol{z}_t$
$p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$
$\boldsymbol{z}_{t-1}$
$\boldsymbol{z}_0$
$p(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{z}_0)$
$\boldsymbol{x}, \boldsymbol{h}$
$q(\boldsymbol{z}_t|\boldsymbol{x}, \boldsymbol{h})$



R
$q(\boldsymbol{z}_T, \ldots|\boldsymbol{x}, \boldsymbol{h})$
diffuse
$p(\boldsymbol{x}, \boldsymbol{h}, \ldots|\boldsymbol{z}_T)$
denoise
$q(\boldsymbol{z}_T, \ldots|\boldsymbol{x}, \boldsymbol{h})$
diffuse
$p(\boldsymbol{x}, \boldsymbol{h}, \ldots|\boldsymbol{z}_T)$
denoise
R
$p(\boldsymbol{x}, \boldsymbol{h})$
=
$p(\mathbf{R}\boldsymbol{x}, \boldsymbol{h})$

Adapted from M Welling



Conditional Generation is Hard (but = holy grail)

Sampling Equilibrium Distribution

Sampling Conditional Distribution

Sampling Mixture Density Network

Deterministic NN

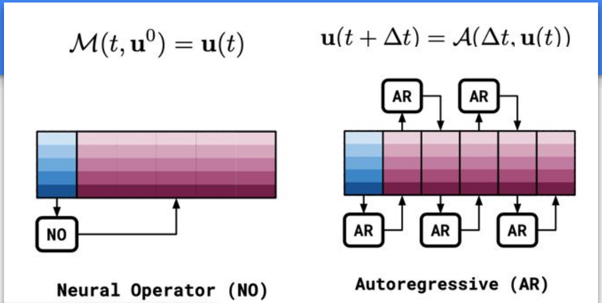
VAE

Adapted from M Welling

PDEs

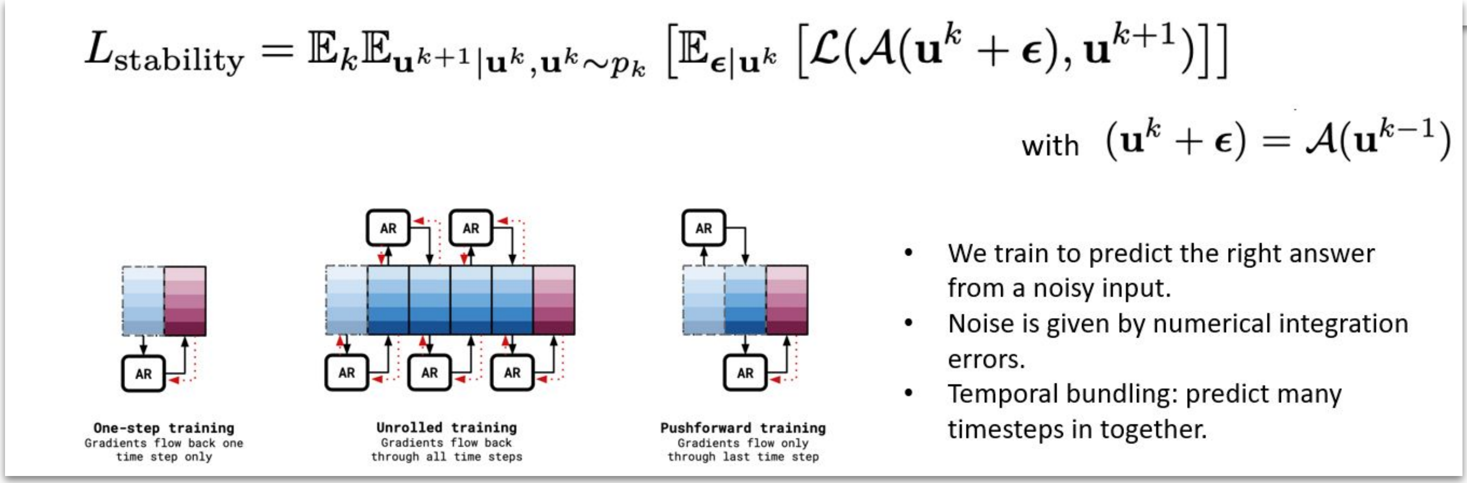
Describe evolution of systems

Great talk from Max Welling



$$\mathcal{M}(t, \mathbf{u}^0) = \mathbf{u}(t)$$

$$\mathbf{u}(t + \Delta t) = \mathcal{A}(\Delta t, \mathbf{u}(t))$$



$$L_{\text{stability}} = \mathbb{E}_k \mathbb{E}_{\mathbf{u}^{k+1}|\mathbf{u}^k, \mathbf{u}^k \sim p_k} \left[\mathbb{E}_{\epsilon|\mathbf{u}^k} \left[\mathcal{L}(\mathcal{A}(\mathbf{u}^k + \epsilon), \mathbf{u}^{k+1}) \right] \right]$$

with $(\mathbf{u}^k + \epsilon) = \mathcal{A}(\mathbf{u}^{k-1})$

AR
One-step training
Gradients flow back one time step only
Unrolled training
Gradients flow back through all time steps
Pushforward training
Gradients flow only through last time step

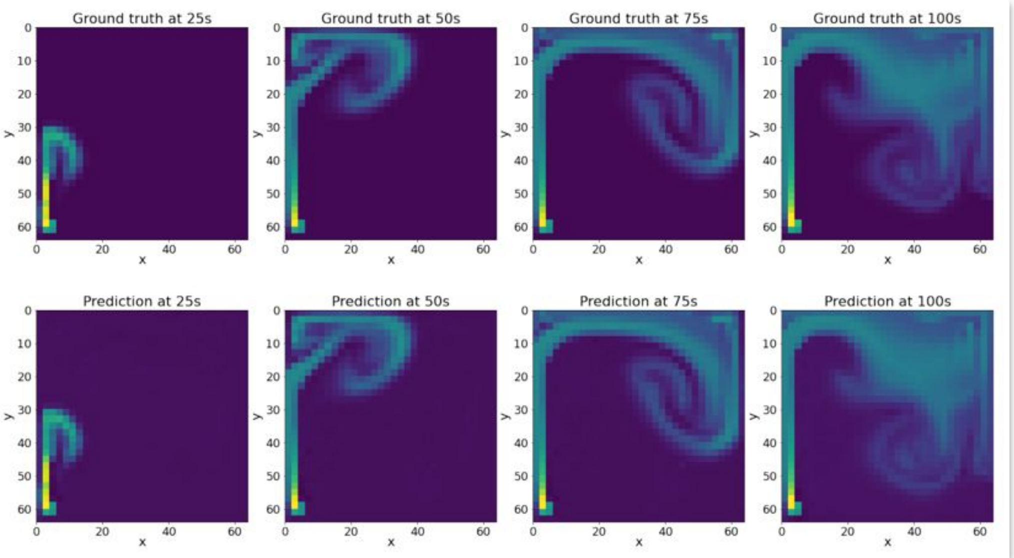
- We train to predict the right answer from a noisy input.
- Noise is given by numerical integration errors.
- Temporal bundling: predict many timesteps in together.

Adapted from M Welling

PDEs

Great predictions for time evolution of systems

- Smoke with 2D Navier Stokes
- Shock waves with Burgers equation
- Modelling waves with KdV equation



Summary

Summary

Great workshop with lots of discussion - big focus on detector simulation

- Very active area in HEP with generative modelling
- By no means a solved problem, but moving to deployment

Success in wide range of physical sciences and applications

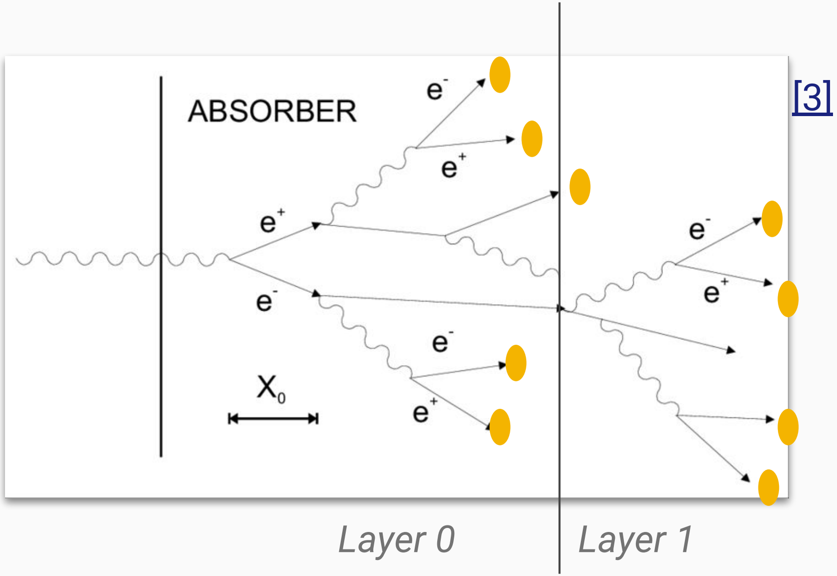
Room for lots more synergy and inspiration between models across physics domains, leveraging symmetries and invariances

Backup

Point Clouds

But! - Don’t record history of shower development

- Each point (E, x) is end of a branch!
- Multiplicity can be incredibly high
- Points aren’t directly connected
- Density of points between layers correlated but cannot easily build a physical graph
- Typically no time information, just layer depth



[3]

And for sampling calorimeters dead material

- Low sampling fraction - lots of missed hits!

Current point cloud models aren’t designed to learn underlying structure - they focus on learning surfaces or connected structure! Difficult to use standard benchmarks!