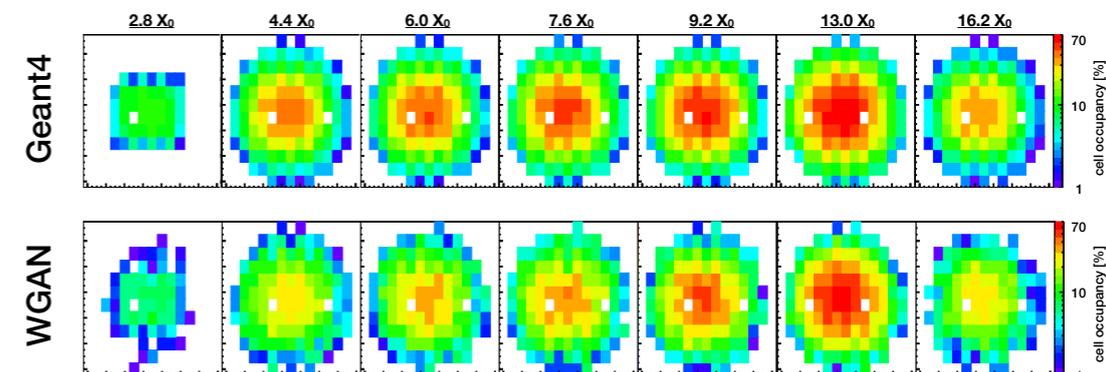




CLICdp Collaboration Meeting, 29th August 2018

Precise *simulation* of
electromagnetic calorimeter
showers using a Wasserstein
Generative Adversarial *Network*



Thorben Quast

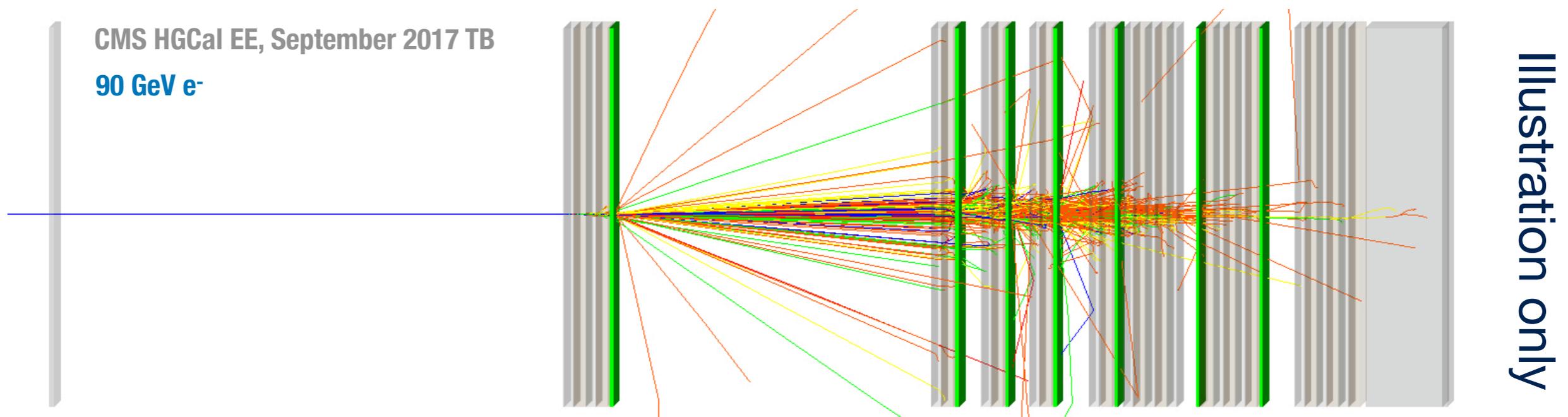
29.08.2018

Calorimeter simulation

- Computationally expensive: simulation of particles interacting with material.

Geant 4

- electromagnetic & hadronic physics, lists with increasing/decreasing accuracy.



Calorimeter “simulation” with generative models

- Computationally expensive: simulation of particles interacting with material.

Geant 4

- electromagnetic & hadronic physics, lists with increasing/decreasing accuracy.

- Grand goal: replace simulation steps by *ultra fast, accurate* generative methods.

➔ **Step 1: Focus on simulation of particles showers in calorimeters.**

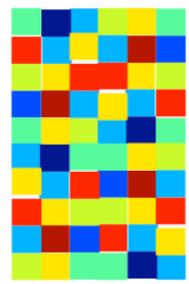
Proof-of-principle already demonstrated:

- e.g. at the 1st IML workshop in 2017 by L. Oliveira, M. Paganini and B. Nachman.
- or *arXiv:1701.05927v2*, *arXiv:1705.02355v2*, *arXiv:1711.08813v1*, S. Vallecorsa @ ACAT2017, *arXiv:1802.03325v1*, ...

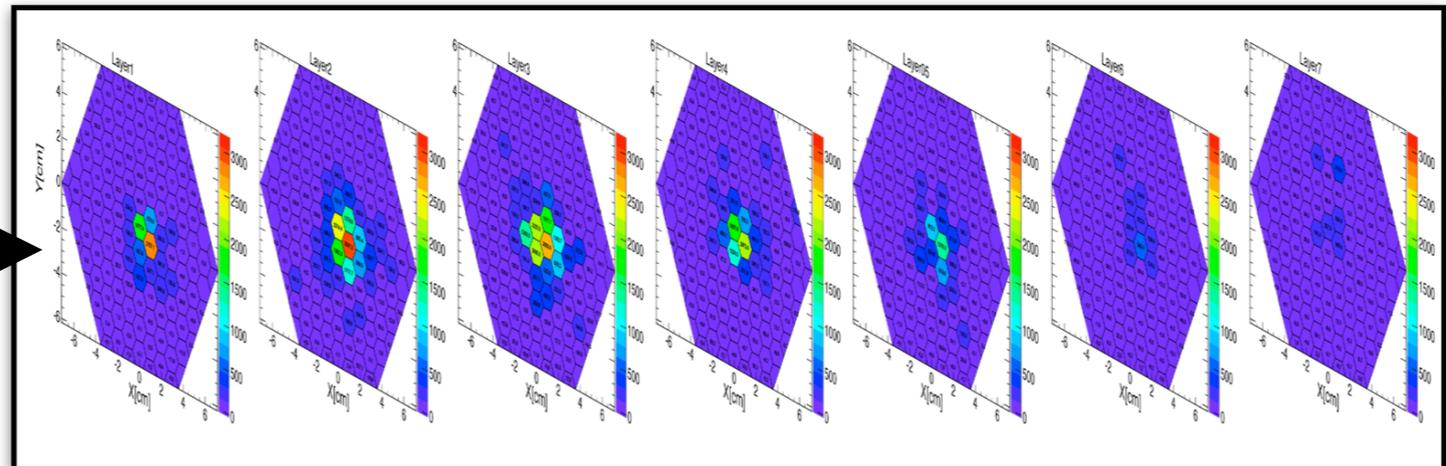
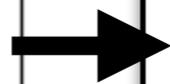
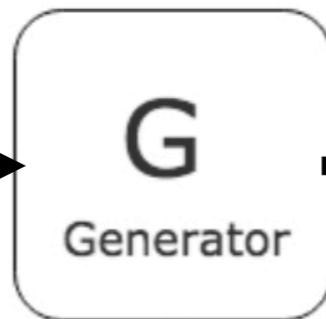
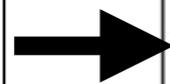
Goal formulation

We want G: Neural network
(Function, $O(10^5)$ free parameters)

“Labels”:
Energy,
impact
(X,Y)



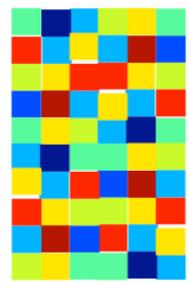
noise



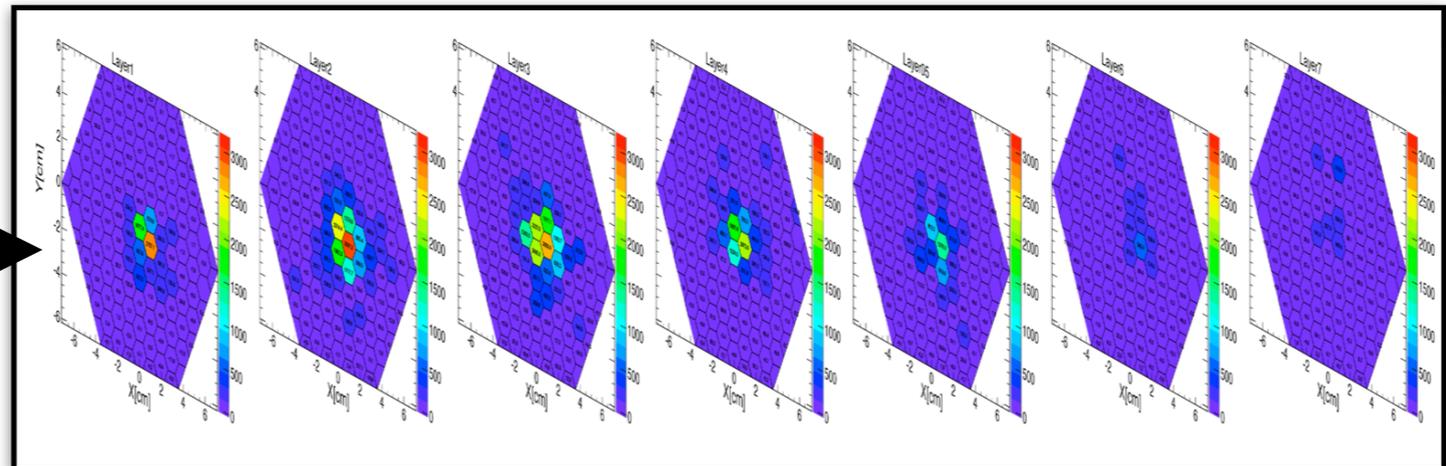
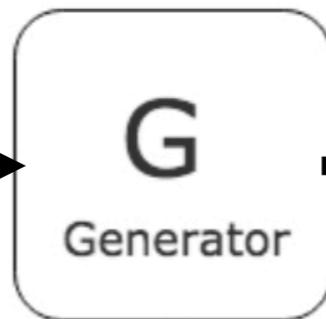
Goal formulation

We want G:

“Labels”:
Energy,
impact
(X,Y)



noise



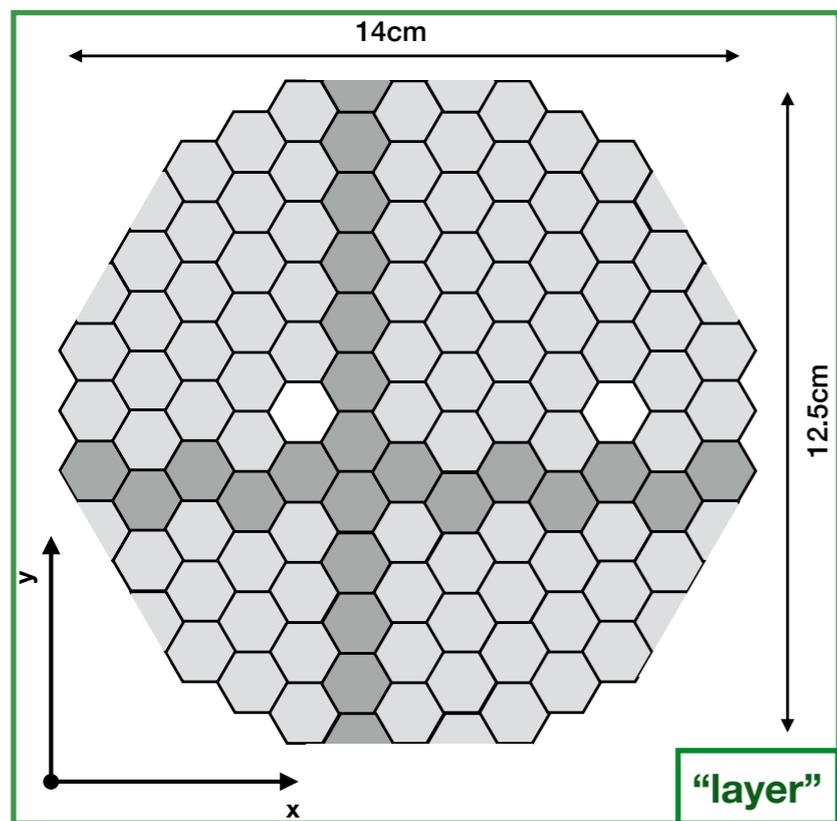
?



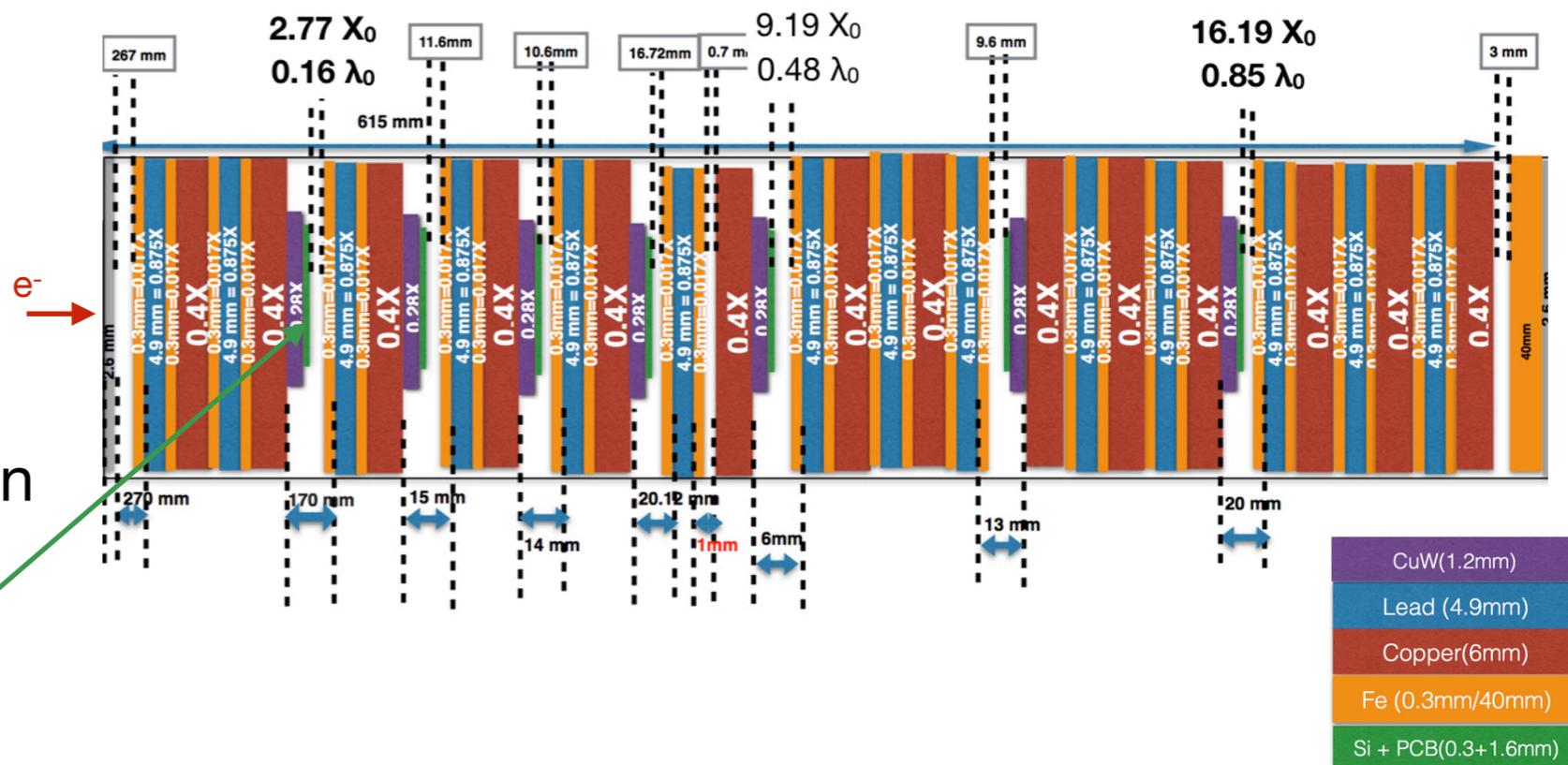
HGCAL prototype in September 2017

Features:

- ▶ **Sampling calorimeter.**
- ▶ **7 sensitive silicon layers.**
- ▶ **2.7 - 16.2 X_0** in depth.
- ▶ **Hexagonal pixels with ~ 1.2 cm in diameter** (128 pixels per layer).



Above: constant x and y coordinates



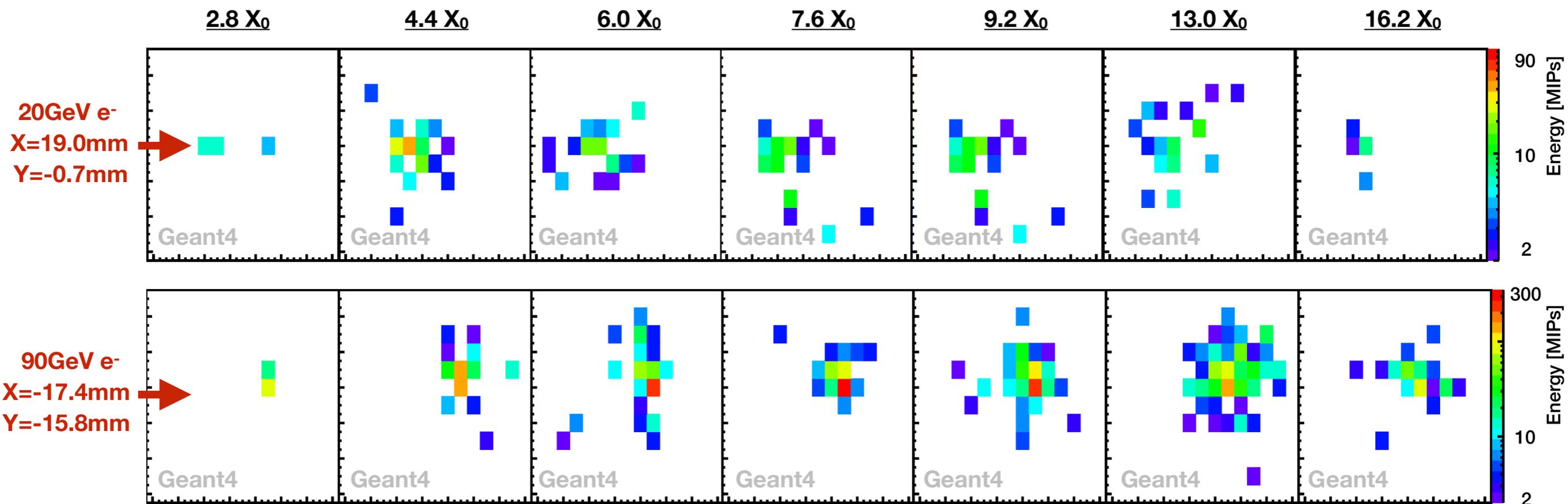
Prototype has been tested with beam...

... but the available statistics of electron showers is likely too low for training a generative model.

➔ Using **Geant 4** simulated electron samples generated *with* beam test conditions.

Exemplary Geant4 showers

- **20, 32, 50, 80 & 90 GeV electrons** with 1% energy spread.
- Sample size: **O(100k)** showers for **each energy** bin.
- Additional **70 GeV electron** sample not used in the training.



➔ **Challenge #1:**
Sparsity of pixel occupations

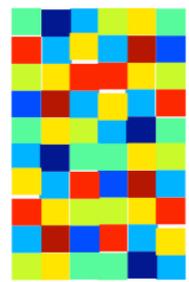
➔ **Challenge #2:**
range of per-pixel energies

➔ **Challenge #3:**
external dependence on
incident energy and position

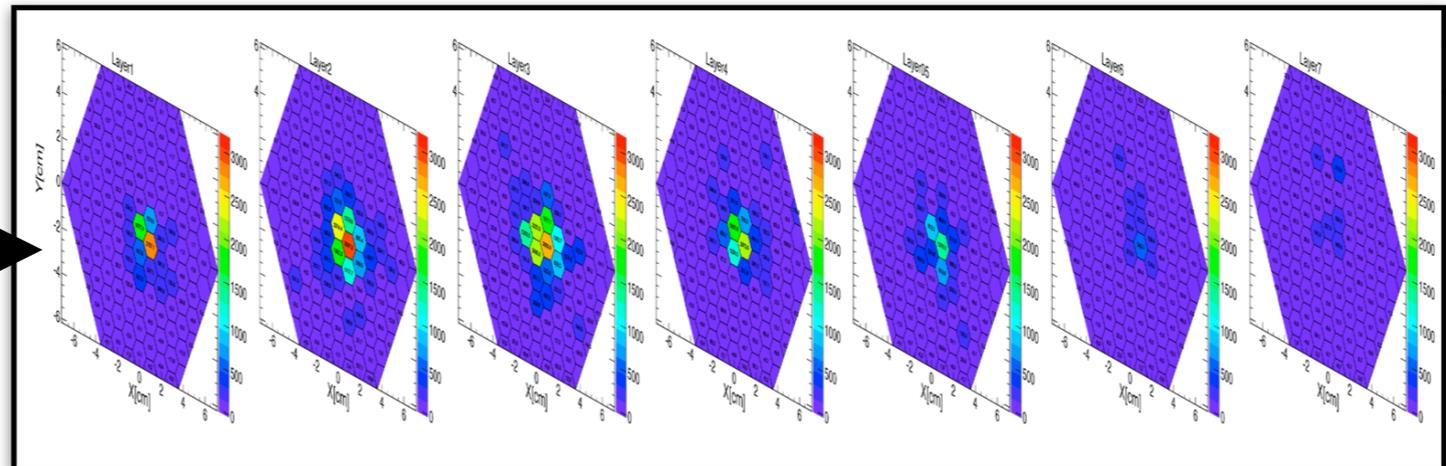
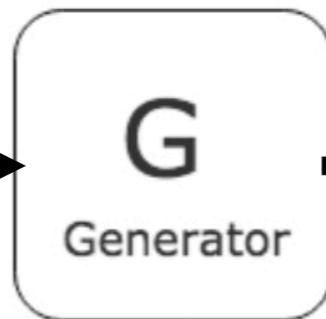
Goal formulation

We want G:

“Labels”:
Energy,
impact
(X,Y)



noise



How to train?

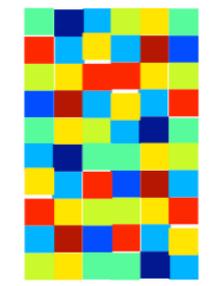
Concept of Generative Adversarial Networks

modified from: <https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

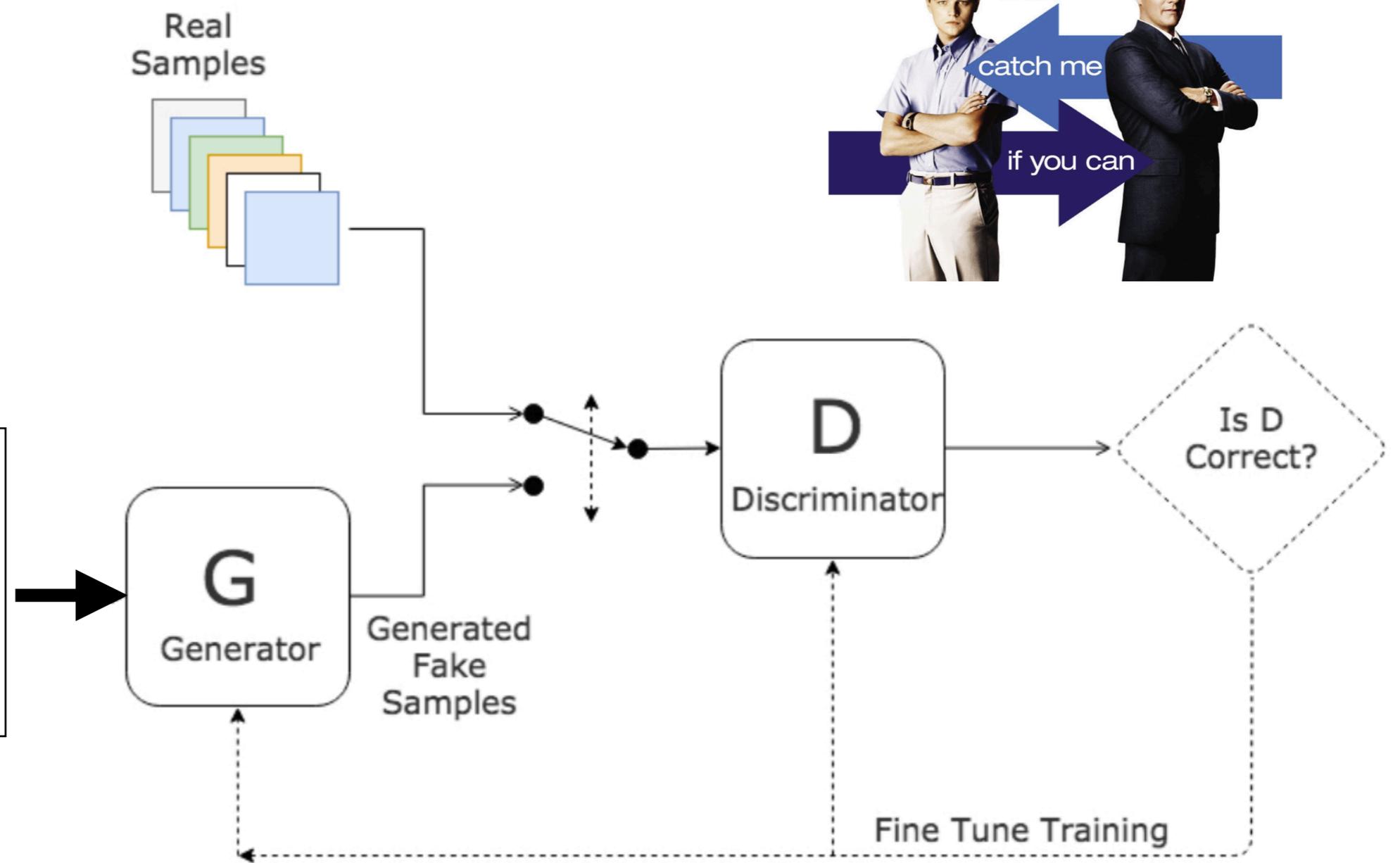
Original
(2014)



“Labels”:
Energy,
impact
(X,Y)



noise



Ian J. Goodfellow's (2014)

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

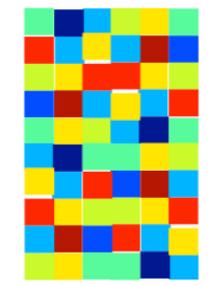
Concept of Generative Adversarial Networks

modified from: <https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

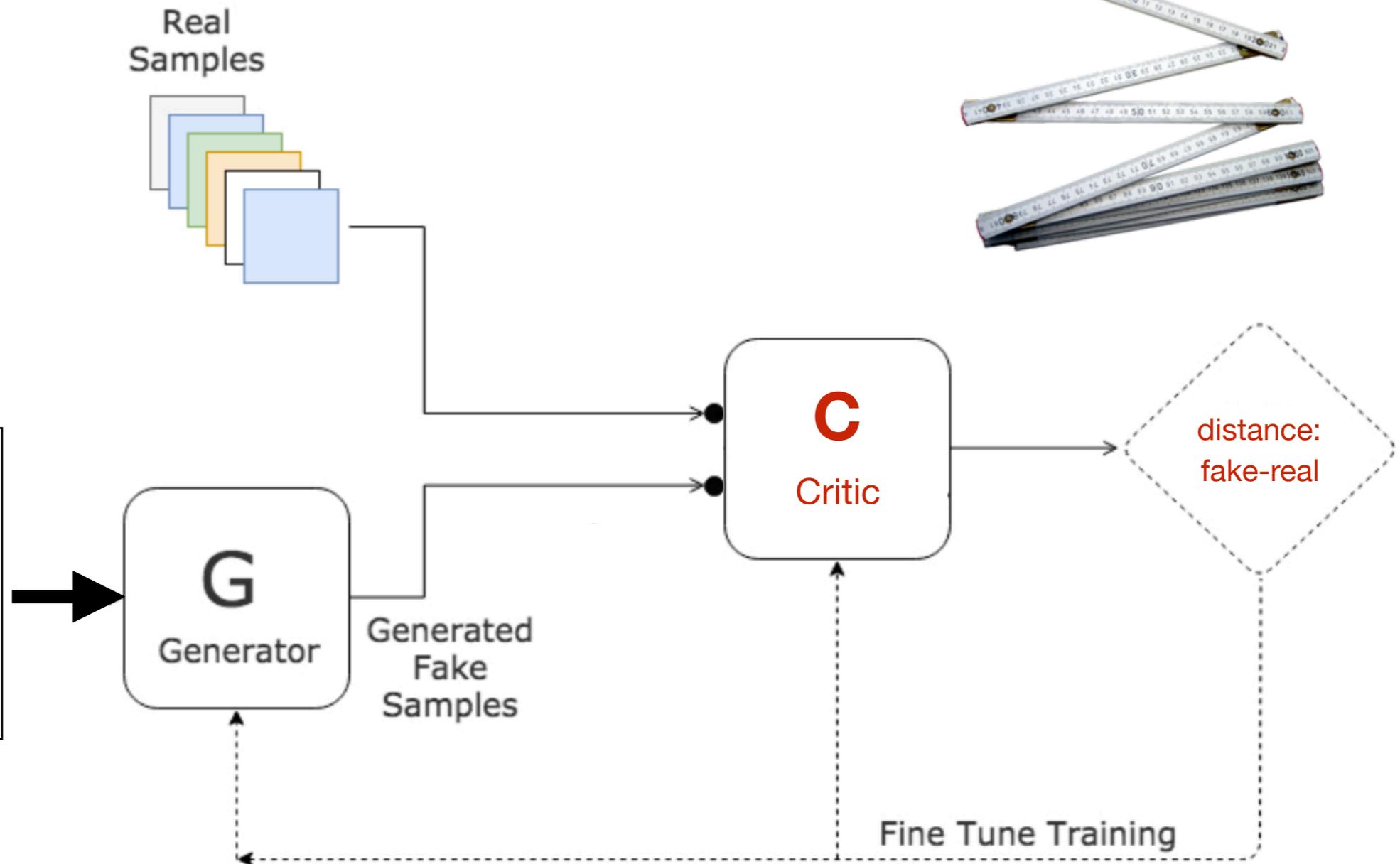
New
(2017)



“Labels”:
Energy,
impact
(X,Y)



noise



EARTH MOVER DISTANCE

arXiv:1704.00028v3

$$L = \underbrace{\mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} [D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} [D(\mathbf{x})]}_{\text{Original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]}_{\text{Our gradient penalty}}.$$

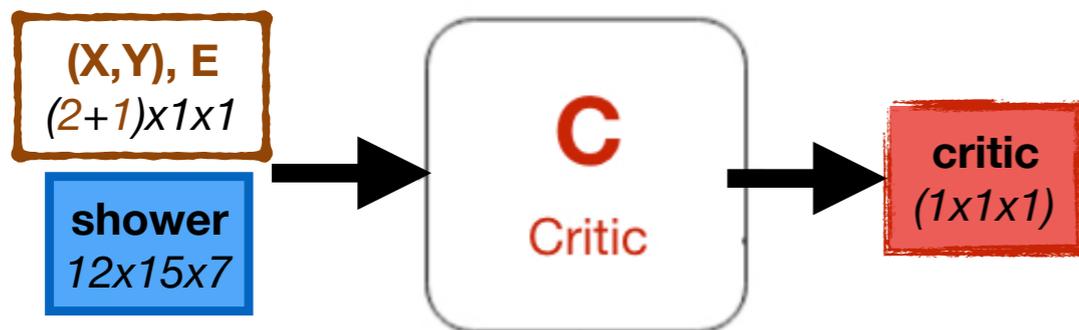
Training strategy using WGANs

- ▶ **Generator network (WGAN)** maps (noise, E_{fakes} , $\text{position}_{\text{fakes}}$) to generated showers.



- ▶ Set of upsampling and convolutions.
- ▶ Batch normalisation.
- ▶ Leaky Relu activation functions except for last step.
- ▶ 672k parameters to be trained.

- ▶ **Critic network (C)** estimates the *Earth Mover* distance btw. generated & real showers.



- ▶ Labels as additional input.
- ▶ Set of convolutions & fully connected layers.
- ▶ Layer normalisation.
- ▶ 477k parameters to be trained.

Figures of merit for training:

Critic loss:

$$C_{\text{loss}} = -\mathbf{C}(\text{showers}_{\text{real}}, E_r, \text{pos.}_{\text{real}}) + \mathbf{C}(\text{showers}_{\text{fakes}}, E_{\text{fakes}}, \text{pos.}_{\text{fakes}}) + \lambda \times \text{gradient penalty},$$

Generator loss w.r.t. critic:

$$g_{\text{loss, c}} = -\mathbf{C}(\text{showers}_{\text{fakes}}, E_{\text{fakes}}, \text{pos.}_{\text{fakes}})$$

$\lambda := 5$

Training strategy to include the conditions, “labels”

- ▶ **2 constrainer networks** for energy- (**E**) and position regression (**P**) on shower images.

Energy regression network E



- ▶ 3D convolutions+FCN, Leaky Relu.
- ▶ Batch normalisation.
- ▶ 96k trainable parameters.

Position regression network P



- ▶ Mostly identical to **E**.
- ▶ Last FCN with two outputs.
- ▶ 96k trainable parameters.

- ▶ **E** and **P** trained using “real” showers - no effect from generated “fake” showers.

Energy and position regression losses:

$$\mathbf{e}_{\text{loss, real}} = (\mathbf{E}(\text{showers}_{\text{real}}) - E_{\text{real}})^2, \quad \mathbf{p}_{\text{loss, real}} = (\mathbf{P}(\text{showers}_{\text{real}}) - \text{pos.}_{\text{real}})^2$$

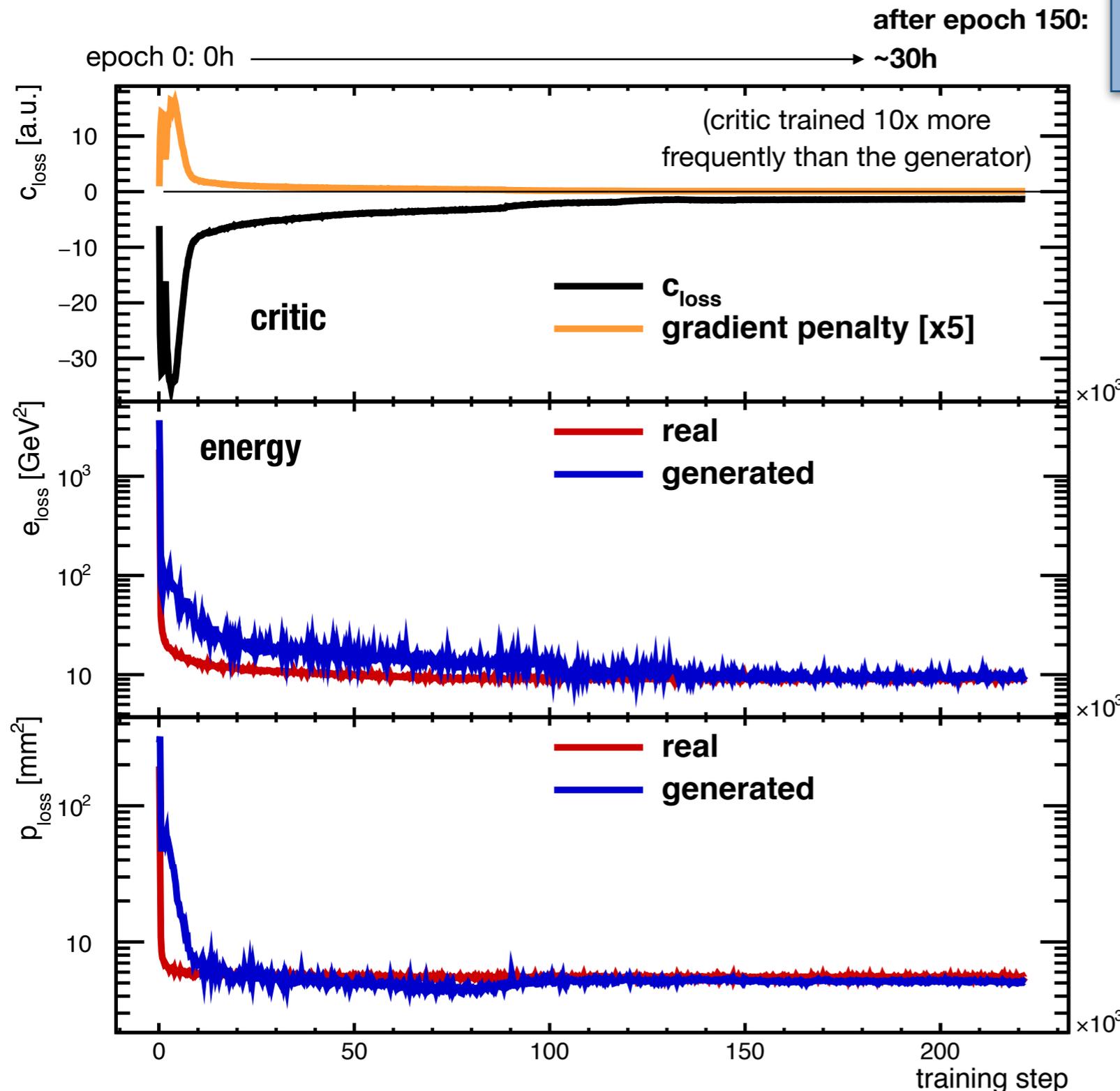
- ▶ **Generator is additionally trained to minimise the regression errors.**

→ Total generator loss combines generator related losses.

$$\mathbf{g}_{\text{loss, tot}} = \mathbf{g}_{\text{loss, c}} + K_e \times |\mathbf{e}_{\text{loss, real}} - \mathbf{e}_{\text{loss, fakes}}| + K_p \times |\mathbf{p}_{\text{loss, real}} - \mathbf{p}_{\text{loss, fakes}}|, \\ K_e := K_p := 0.01$$

System of networks trained for one day

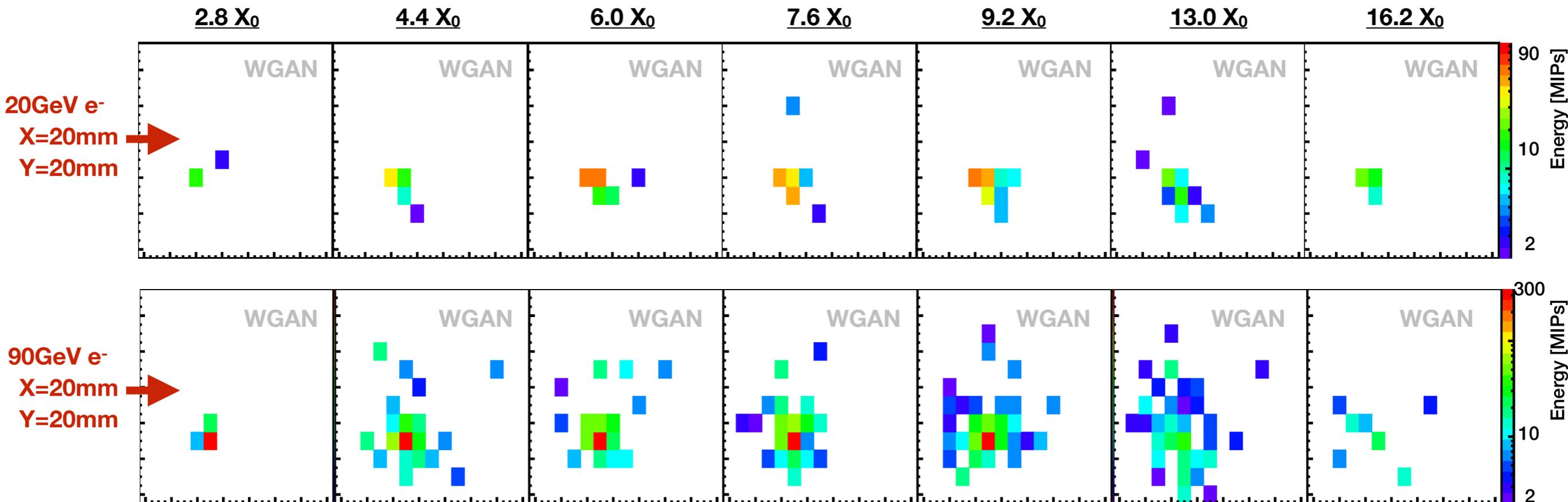
Software: Tensorflow v1.5.
Hardware: NVIDIA GTX1080 GPU.



- ✓ Critic loss converging to 0.
- ✓ Vanishing gradient penalty.
- ✓ Energy regression loss converging fast.
- ✓ Loss on generated images converging.
- ✓ Position regression loss converging.
- ✓ Loss on generated images converging.

→
▸ 1 step → 256 batch of showers.
▸ 1479 steps / epoch.

Generated electron showers look reasonable

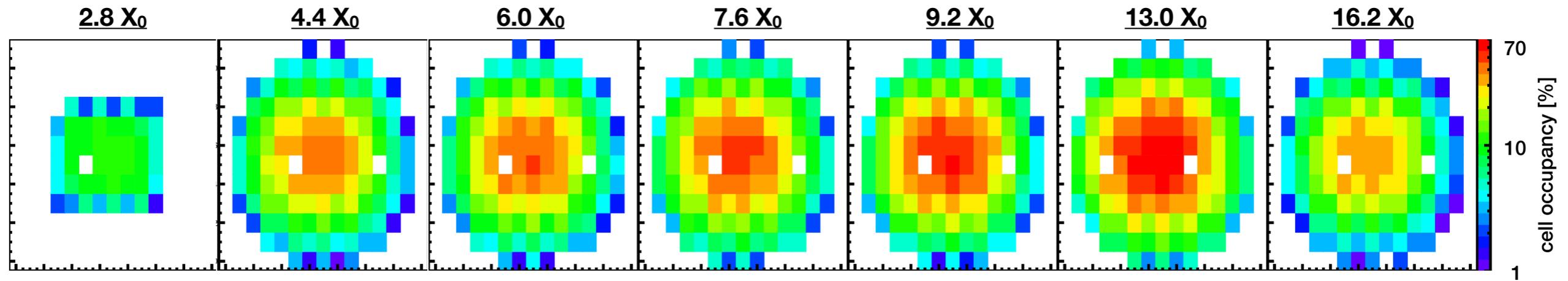


Side note:

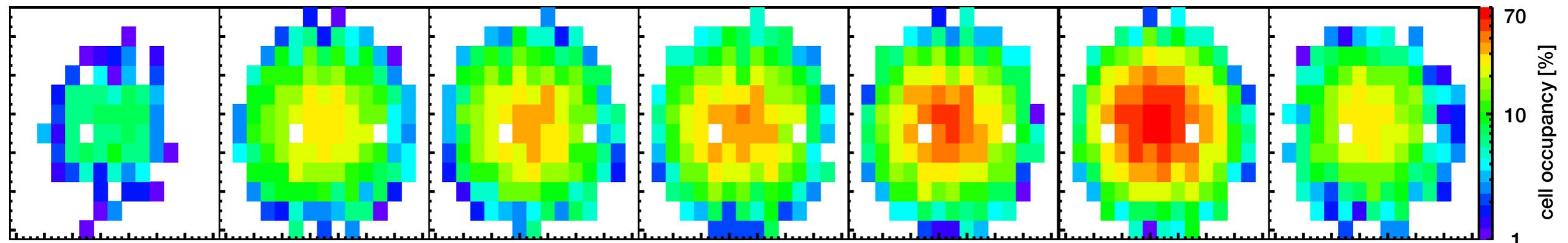
- Reasonable shower images are already obtained after a few training epochs.

WGAN has learnt: Pixel occupancy

Geant4



WGAN



note: masking of regions outside the acceptance in the WGAN

✓Radial development.

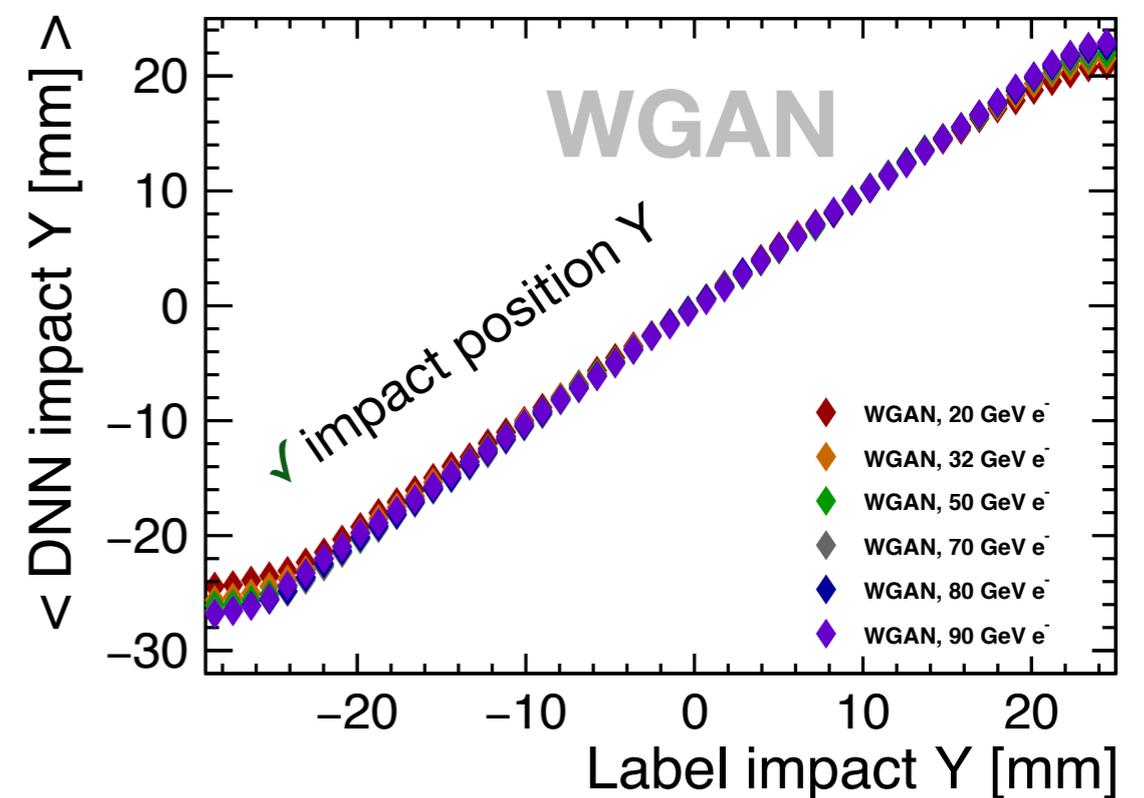
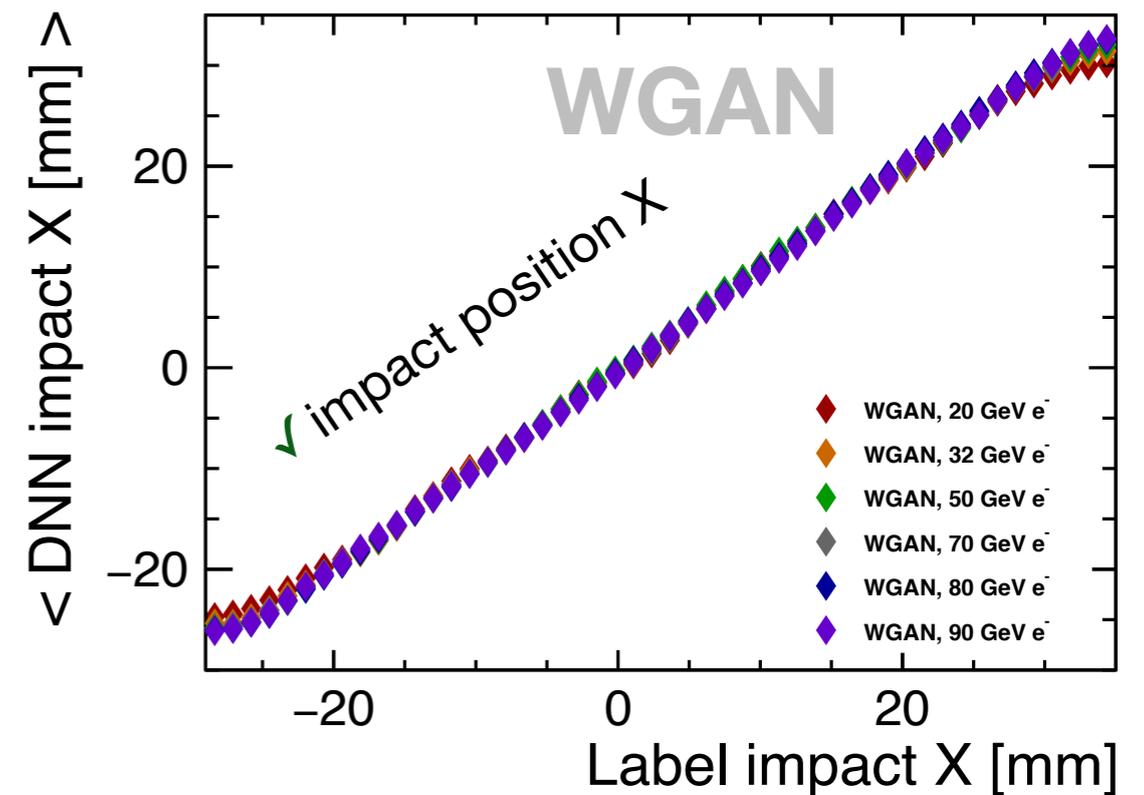
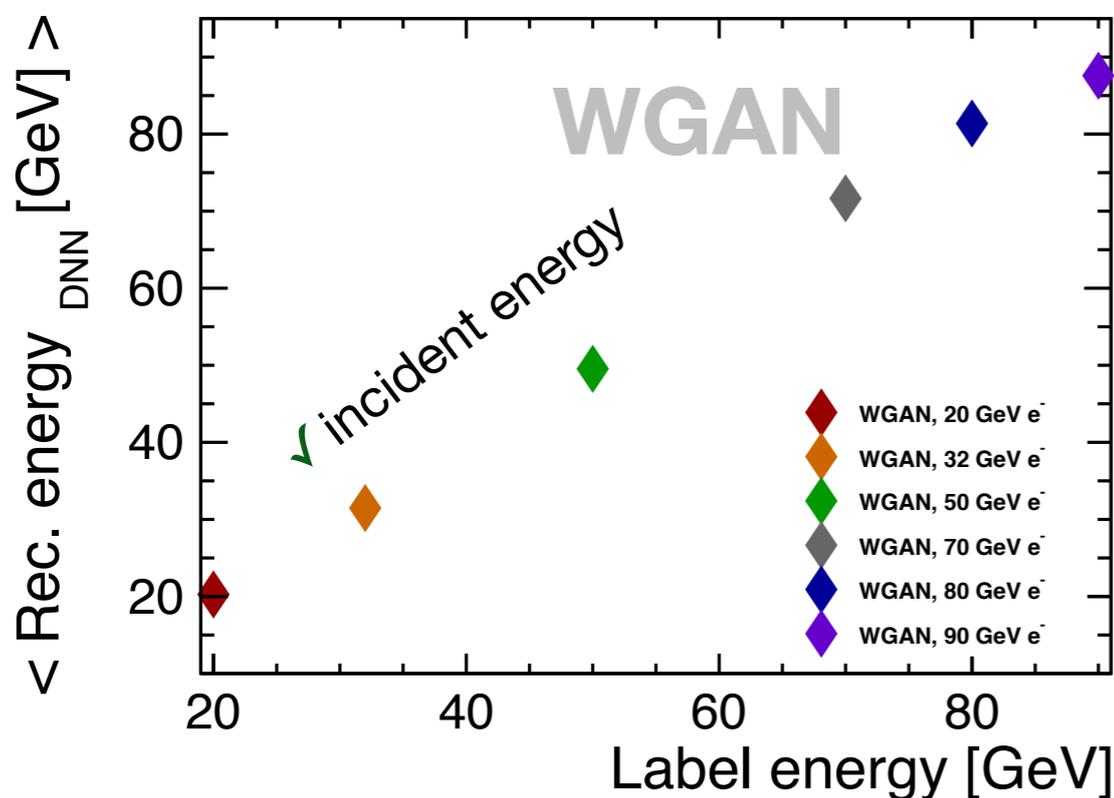
x WGAN: Overall scale slightly underestimated.

Generated events: Dependence on labels

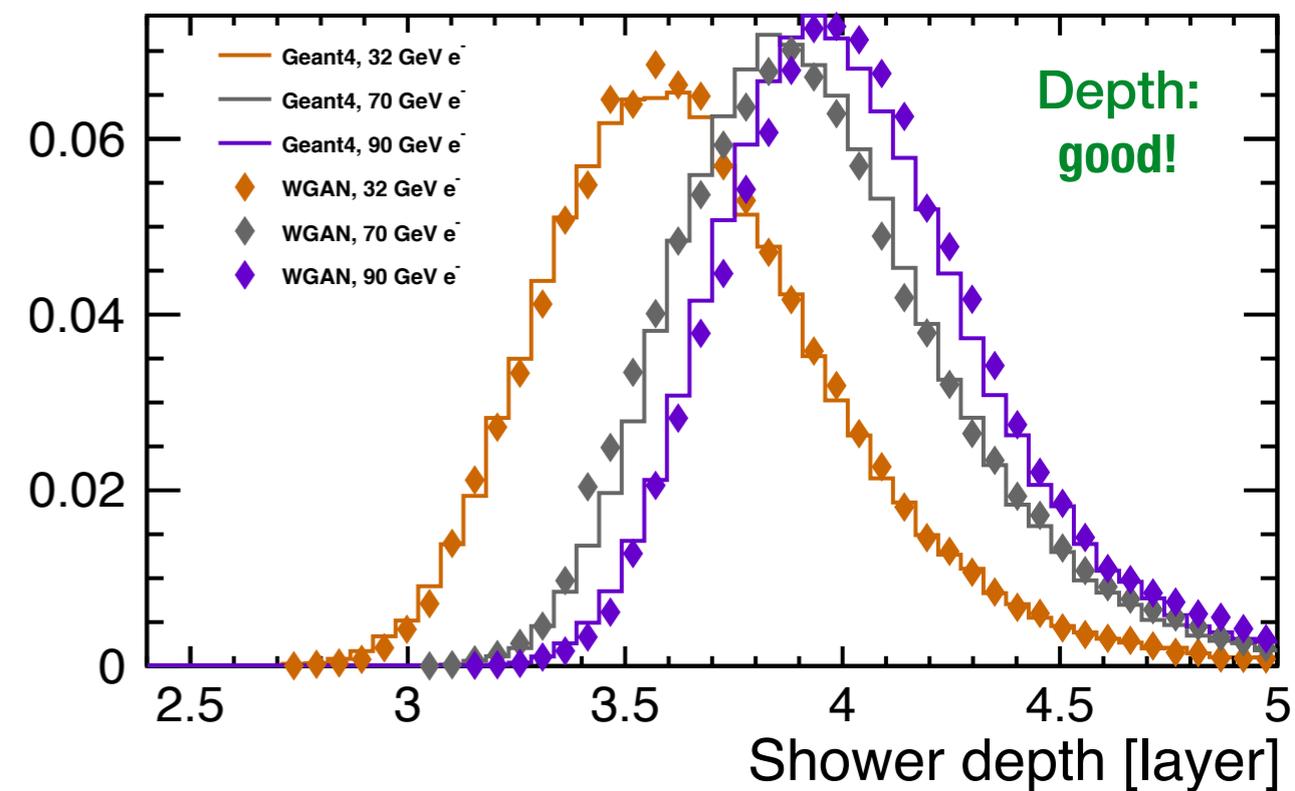
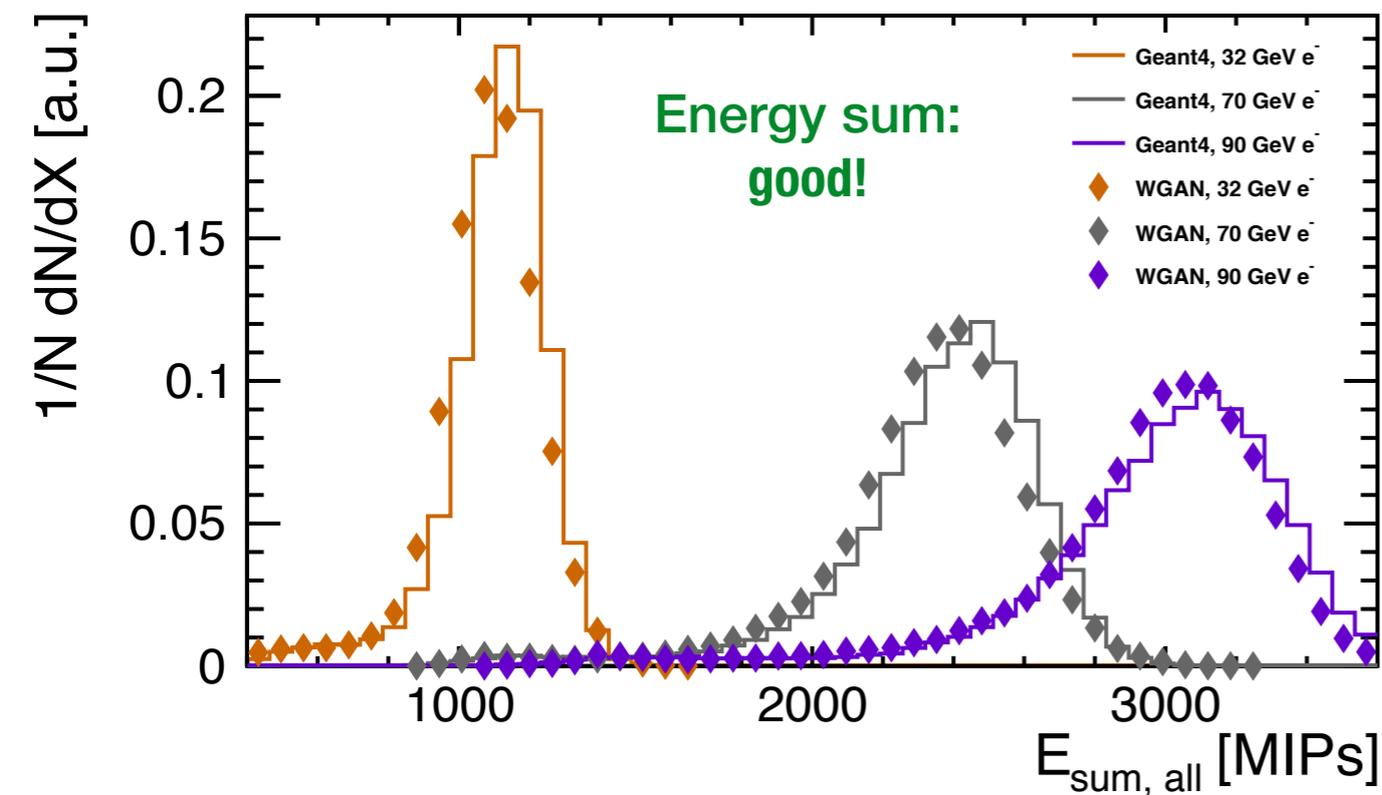
If WGAN has learnt to respect labels:

Reconstructed quantities of generated showers correlate with true label.

Note: 70GeV sample not used in training.

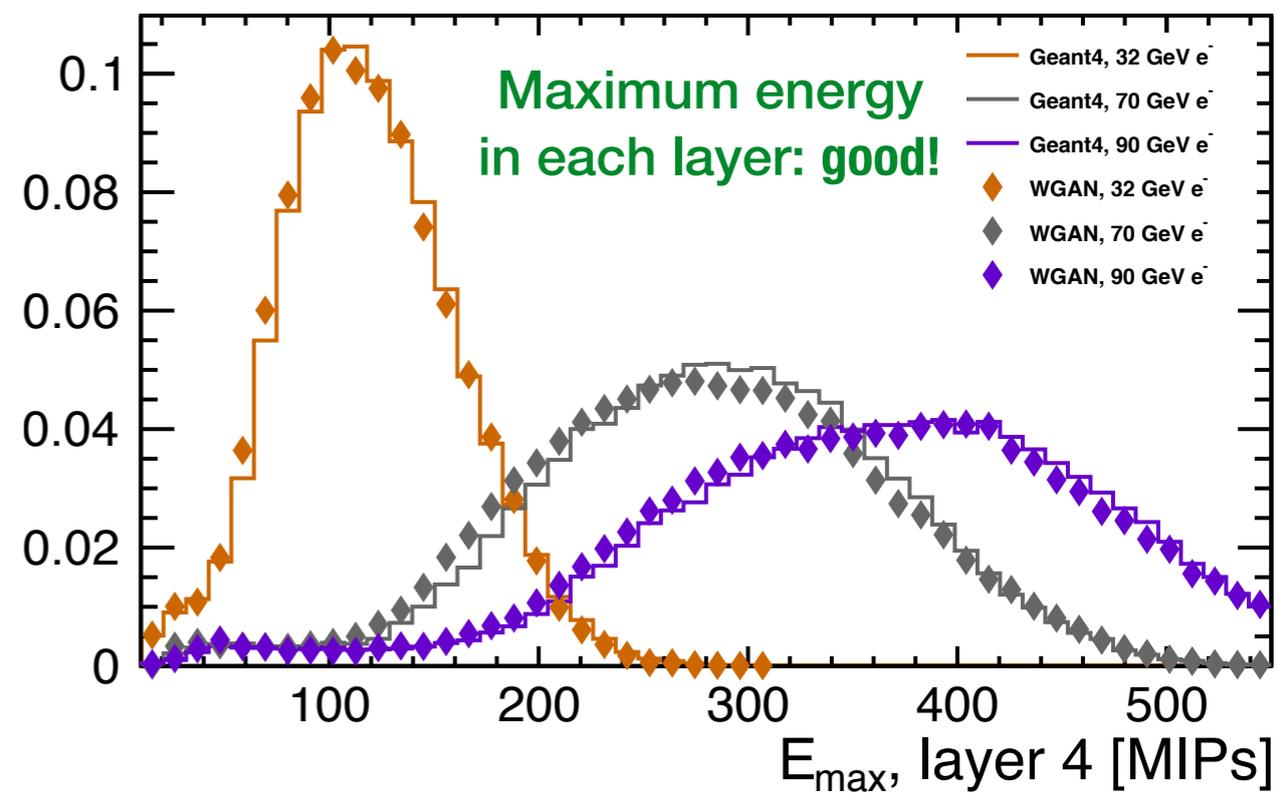
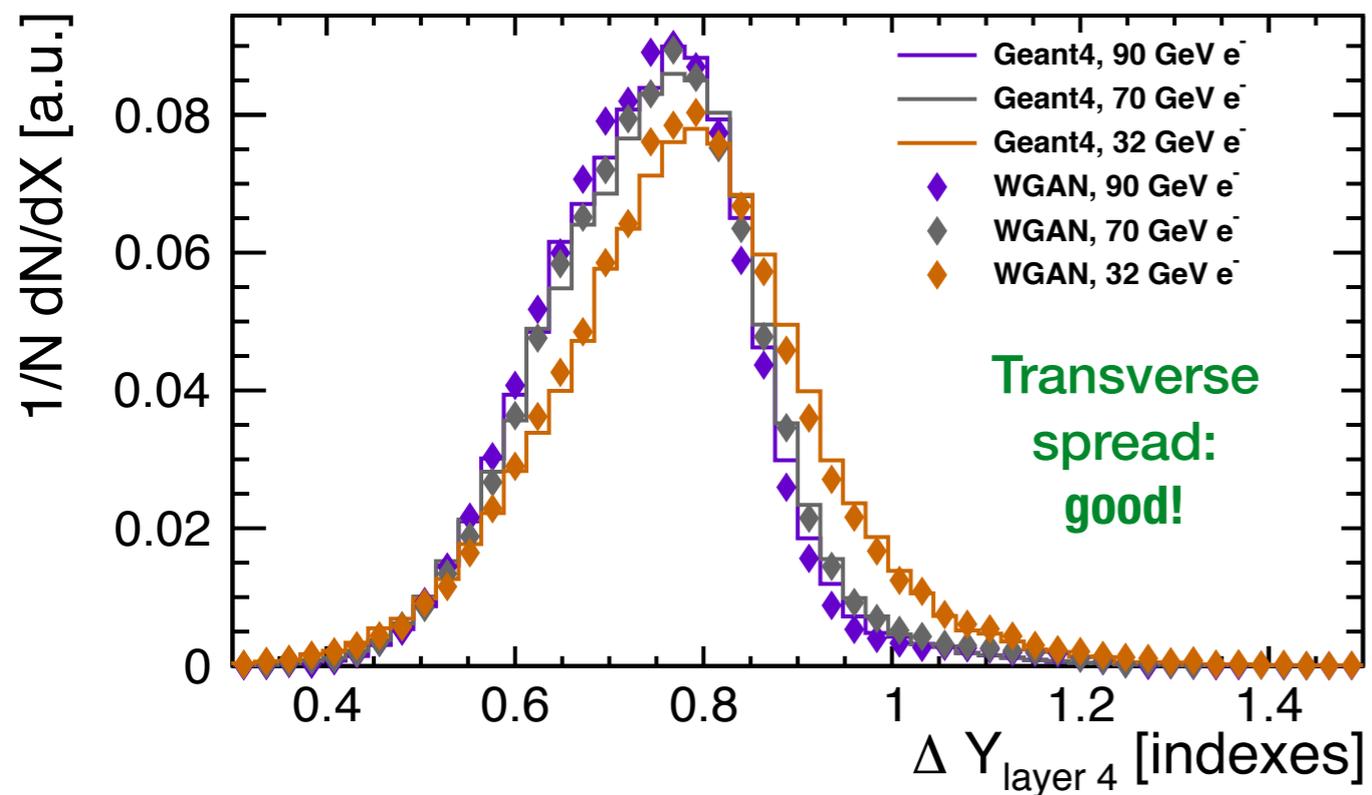


Comparison: Distributions of 1D observables



Note: 70GeV sample not used in training.

Comparison: Distributions of 1D observables

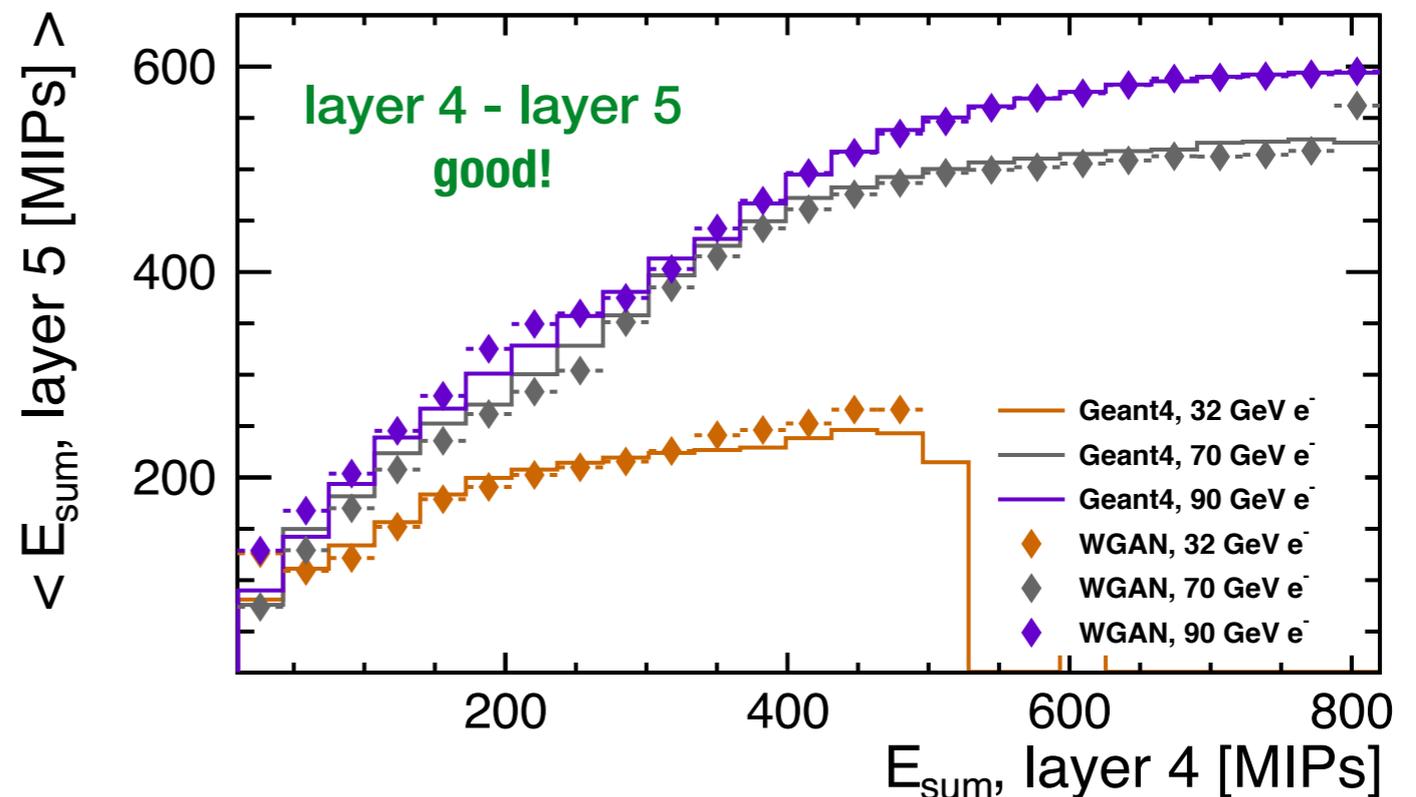
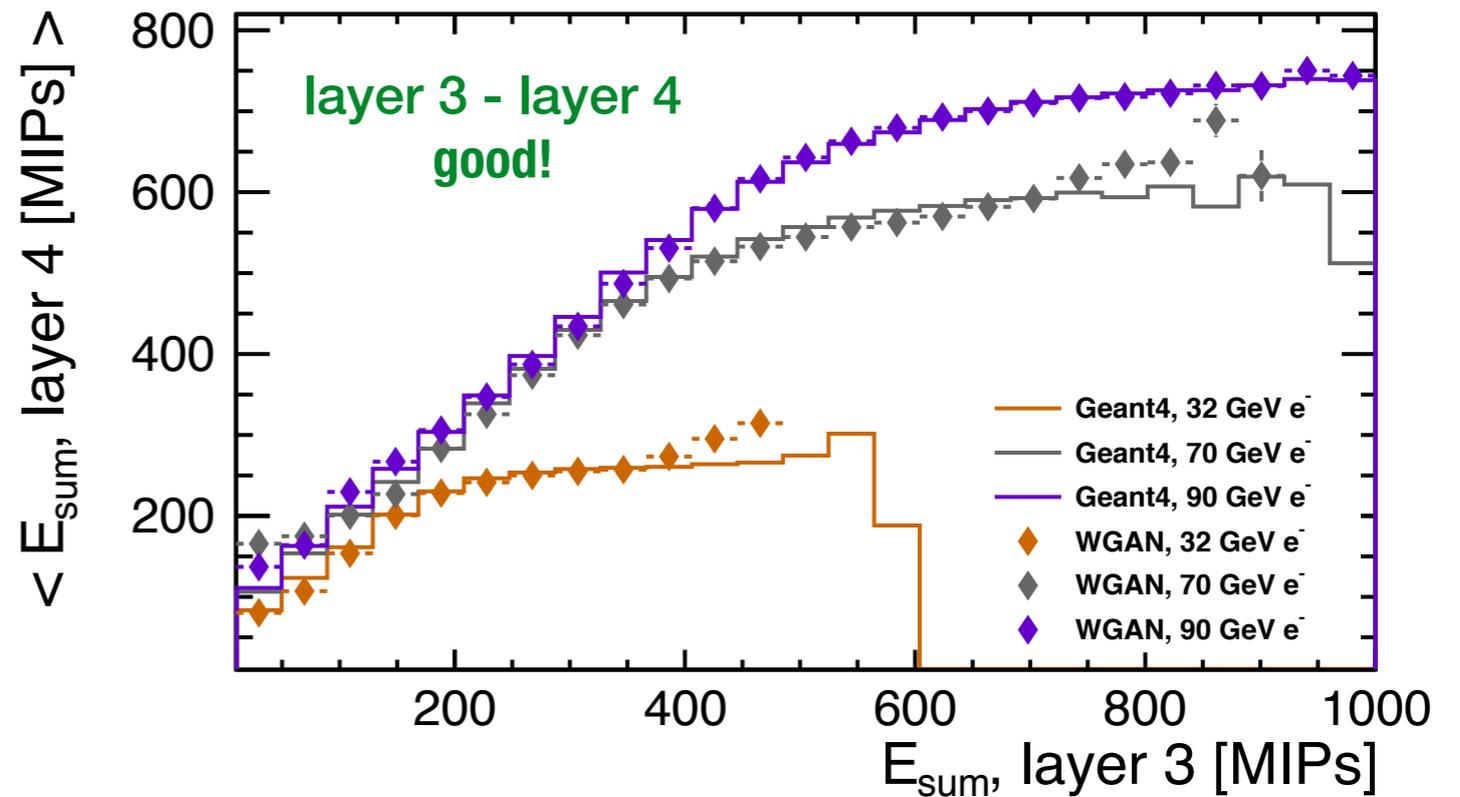


Note: 70GeV sample not used in training.

Correlation between layers

- ▶ Summed energy in one layer $\langle \rightarrow \rangle$ sum in previous layer.

Note: 70GeV sample not used in training.



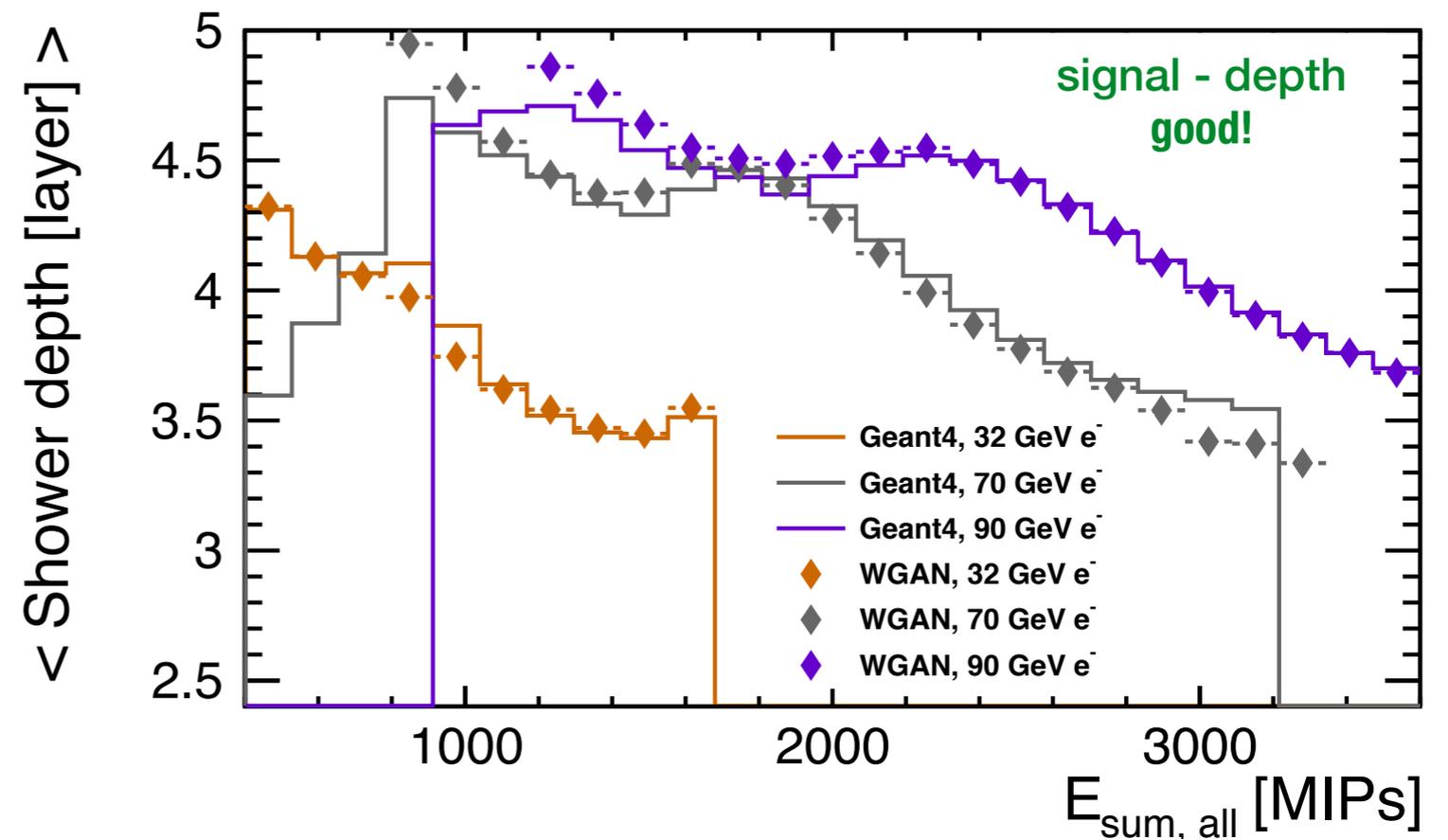
Correlation of depth and signal sum

► Specific sampling configuration:

↑ shower depth

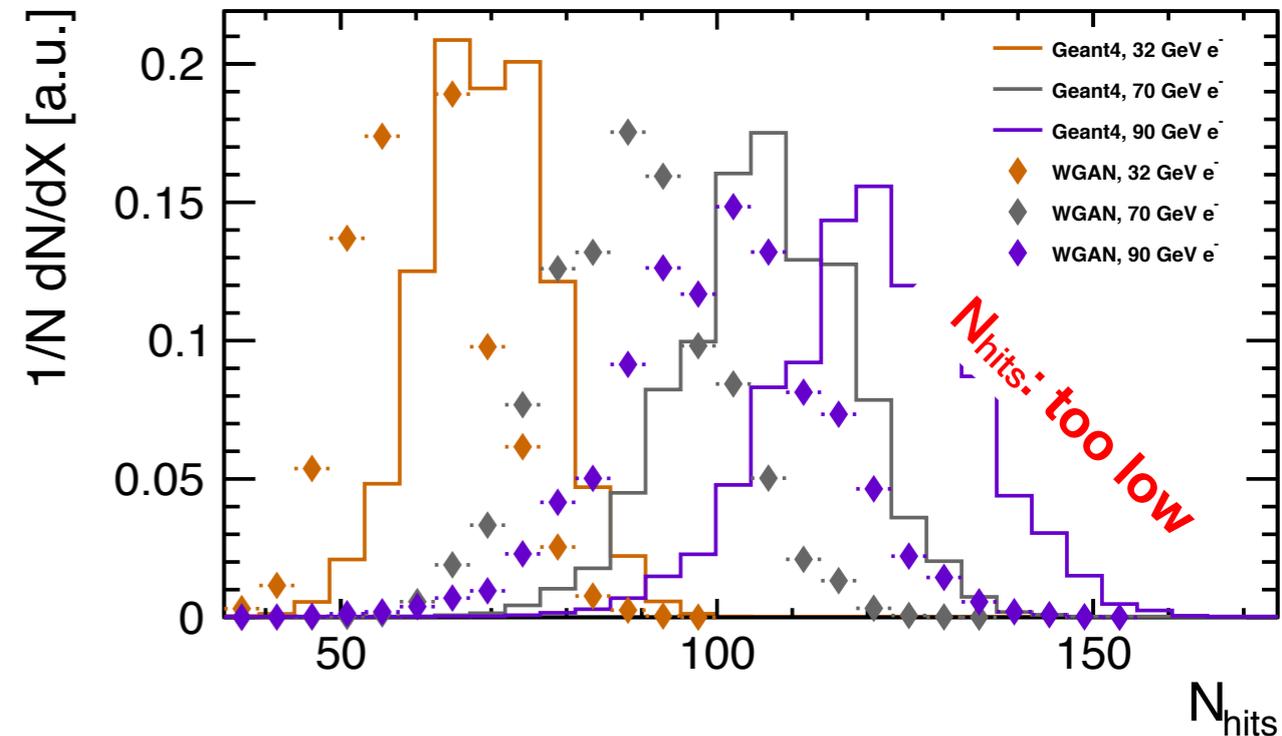
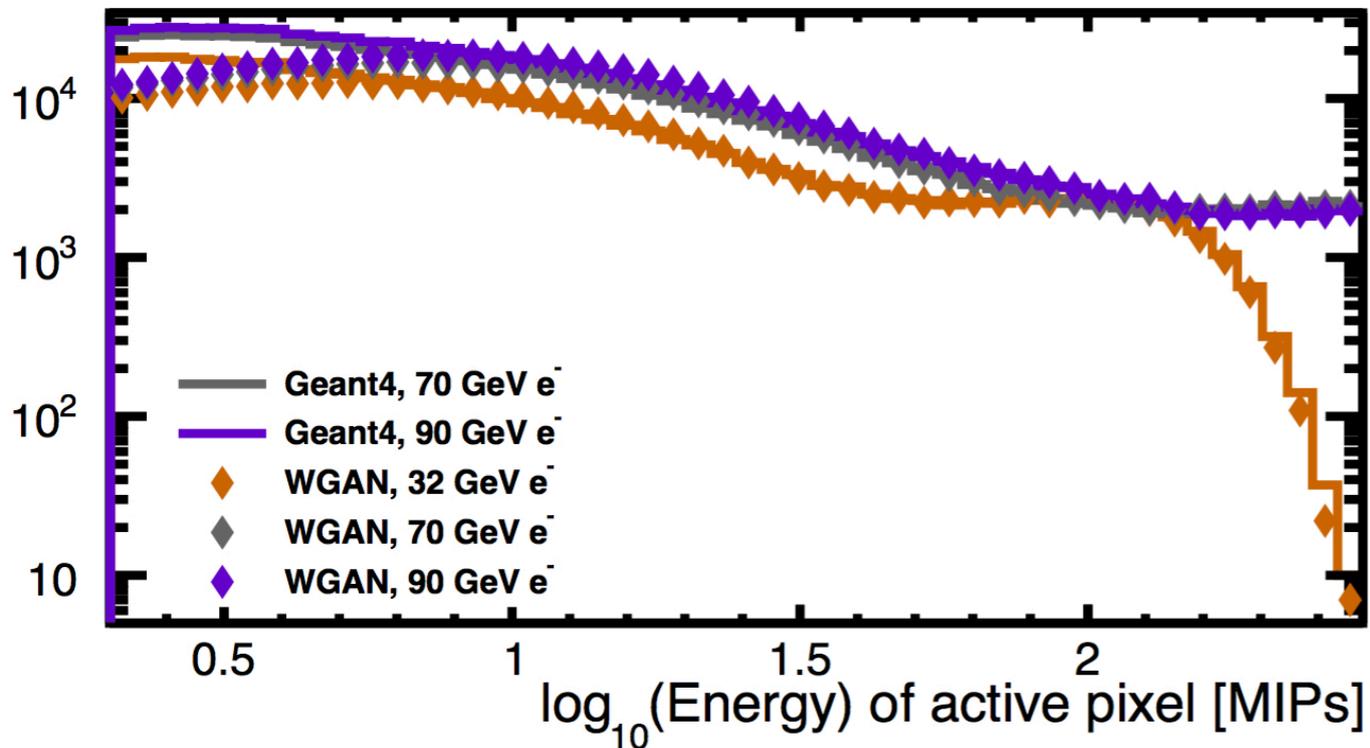
<—>

↓ summed signal



Note: **70GeV sample**
not used in training.

Discrepancy at low energy densities



- ▶ **Underrepresented:** Energy densities below 10MIPs/pixel.
- ▶ $\rho_E < 10$ MIPs/pixel: **Only ~10%** contribution to the total shower signal.
- ▶ Supplementary benchmark with $\rho_E > 10$ MIPs/pixel in the backup.

O(x1000) faster calorimeter simulations possible

- Typical 20-90GeV e⁻ shower generated within **0.5-2 seconds** using **Geant 4**.

Different hardware setups, fixed generator network architectures

Method	Computing Setup	20 GeV e ⁻	Speed-up	90 GeV e ⁻	Speed-up
Geant 4	<i>any</i>	O(500ms)	-	O(2000ms)	-
WGAN	Intel® Xeon® CPU E5-1620	52 ms	x10	52 ms	x40
WGAN	NVIDIA® Quadro® K2000	3.6 ms	x140	3.6 ms	x560
WGAN	NVIDIA® GTX™ 1080	0.3 ms	x1660	0.3 ms	x6660

- ➔ **O(x1000) faster** than full simulation.
- ➔ Evaluation time: **No energy dependence.**

Summary: Calorimeter WGAN

- Generative models: promising **fast simulation tools** for particles' passage through matter.

This study:

- **Wasserstein GAN** concept instead of traditional GANs.
- **Conditioning** impact **position** & **incident** energy shower generating electrons.
- **CMS HGCal prototype** as real-life calorimeter assumed.
(Training with beam test data is possible.)

Key observations:

- ➡ Many **reconstructed quantities** & key **correlations** of generated showers appear in many aspects surprisingly close to **Geant 4** simulation.
- ➡ Discrepancy for low energy densities.
- ➡ Here: Inference step **O(1000)x faster** than **Geant 4**.

<https://arxiv.org/abs/1807.01954>

Backup

Wasserstein GANs

- Concept of Wasserstein loss (Arjovsky et al. 2017) is used.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \xleftrightarrow{\text{Kantorovich duality}} W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r} [f(x)] - \mathbb{E}_{x \sim \mathbb{P}_g} [f(x)]$$

Mathematically
motivated approach.
Relevant for the
application is this.

Proposition 1. Let \mathbb{P}_r and \mathbb{P}_g be two distributions in \mathcal{X} , a compact metric space. Then, there is a 1-Lipschitz function f^* which is the optimal solution of $\max_{\|f\|_L \leq 1} \mathbb{E}_{y \sim \mathbb{P}_r} [f(y)] - \mathbb{E}_{x \sim \mathbb{P}_g} [f(x)]$. Let π be the optimal coupling between \mathbb{P}_r and \mathbb{P}_g , defined as the minimizer of: $W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\pi \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \pi} [\|x - y\|]$ where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of joint distributions $\pi(x, y)$ whose marginals are \mathbb{P}_r and \mathbb{P}_g , respectively. Then, if f^* is differentiable[†], $\pi(x = y) = 0$ [§], and $x_t = tx + (1 - t)y$ with $0 \leq t \leq 1$, it holds that $\mathbb{P}_{(x,y) \sim \pi} \left[\|\nabla f^*(x_t)\| = \frac{\|y - x\|}{\|y - x\|} \right] = 1$.

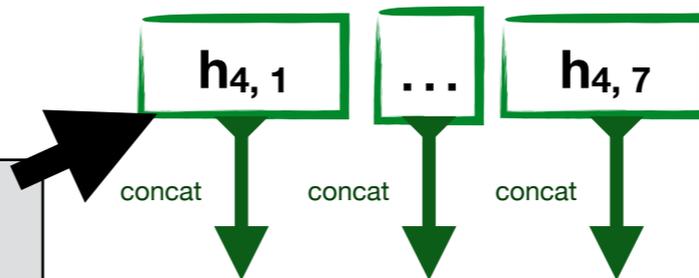
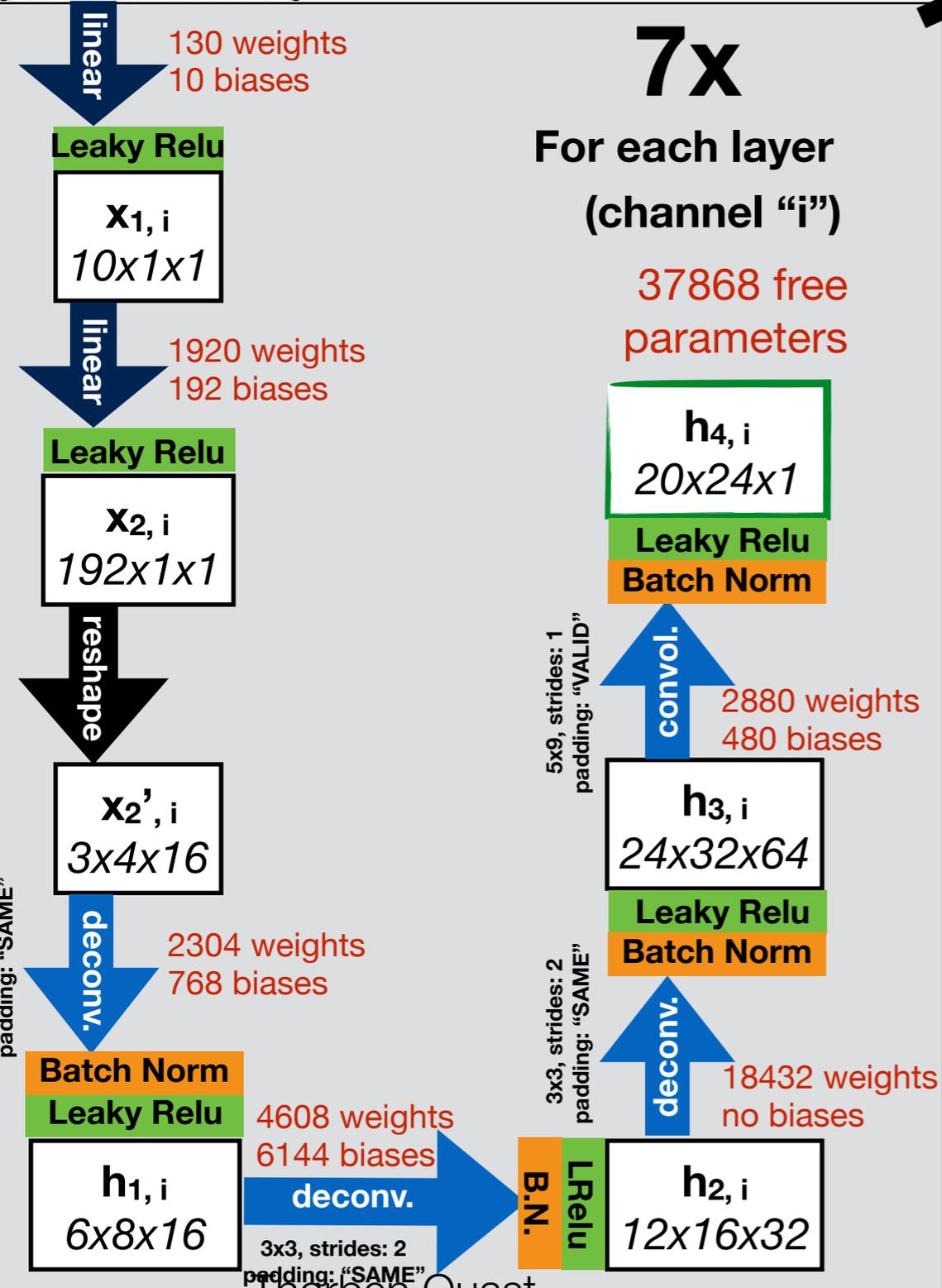
Corollary 1. f^* has gradient norm 1 almost everywhere under \mathbb{P}_r and \mathbb{P}_g .

$$L = \underbrace{\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)]}_{\text{Original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{Our gradient penalty}}.$$

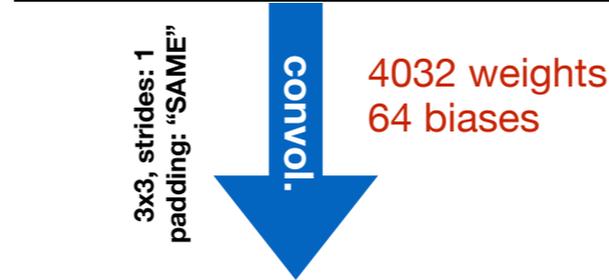
- ➔ **Critic $D(x)$ instead of a discriminator network.**
- ➔ L is a direct measure for the convergence of the training.

Generator network with ~672k free parameters

$z, (X, Y), E$
 $(10+2+1) \times 1 \times 1$

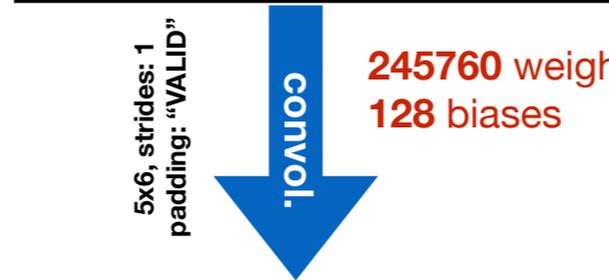


h_5
 $20 \times 24 \times 7$



Batch Norm
 Leaky Relu

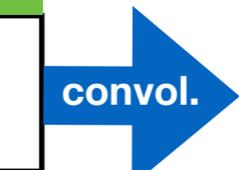
h_6
 $20 \times 24 \times 64$



Batch Norm
 Leaky Relu

h_7
 $16 \times 19 \times 128$

147456 weights, 128 biases



3x3, strides: 1, padding: "VALID"

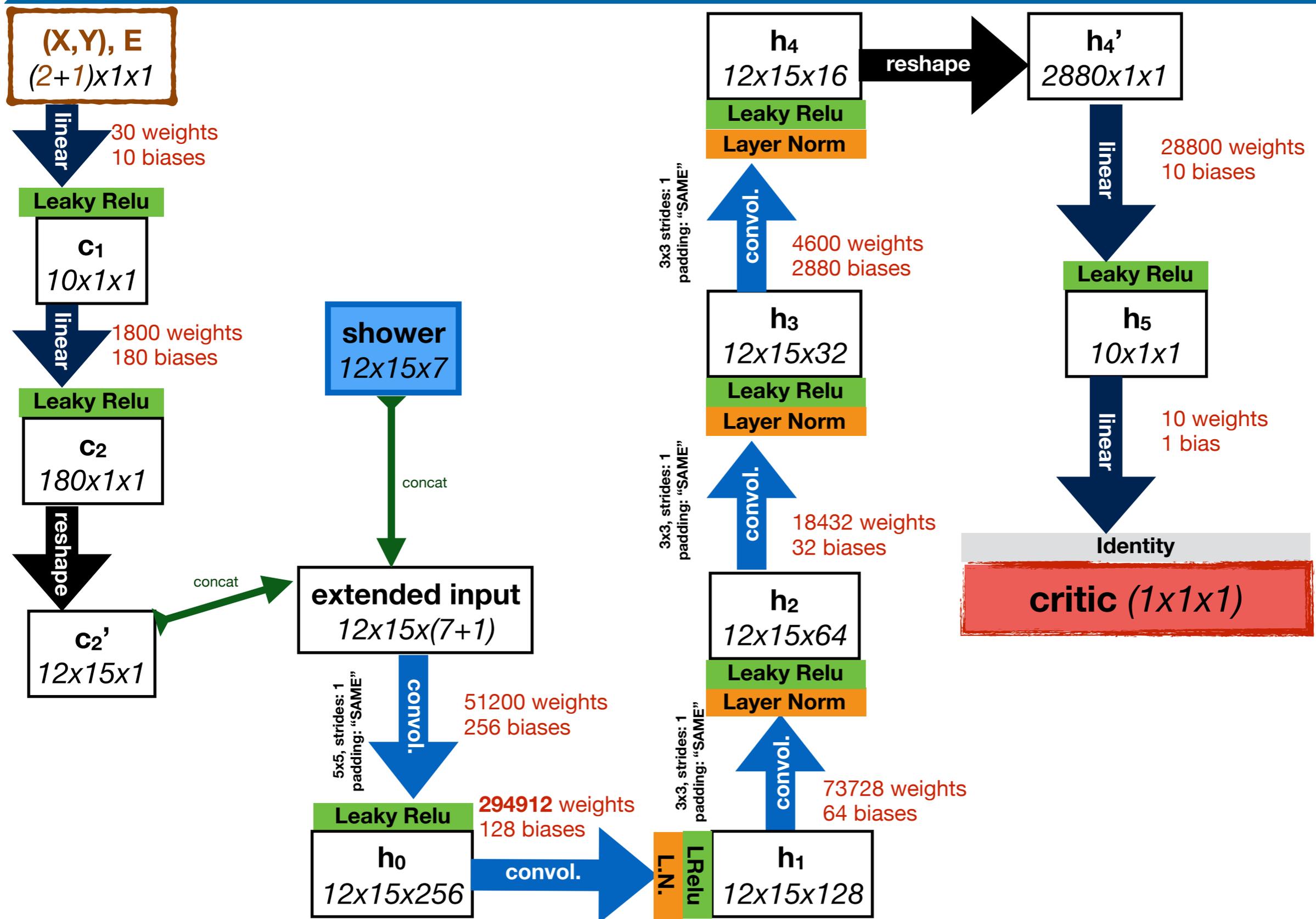
shower ($12 \times 15 \times 7$)
 Relu



B.N.
 LRelu

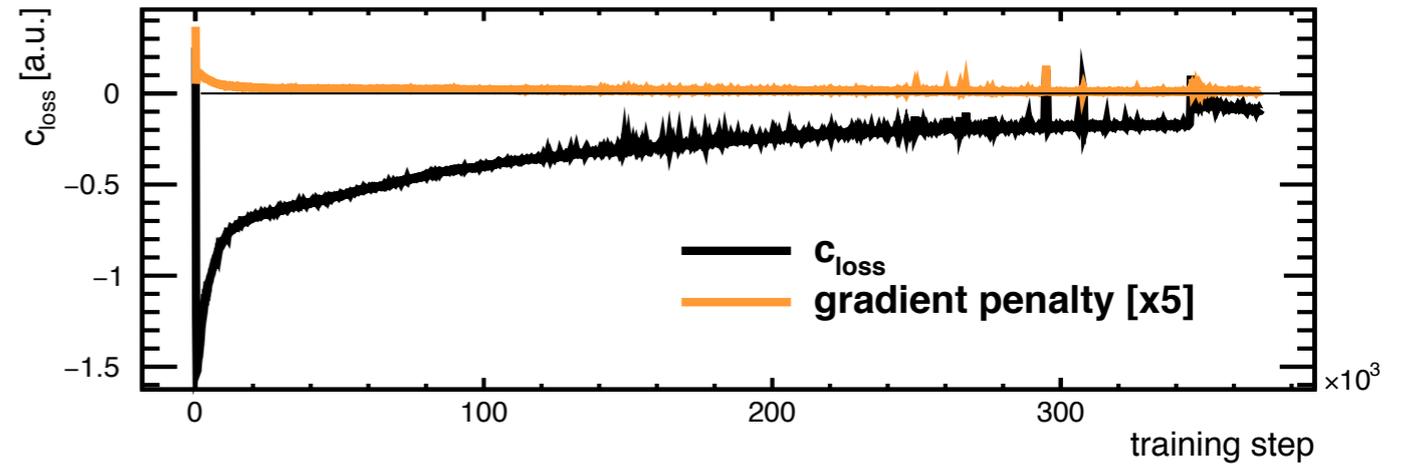
h_8
 $14 \times 17 \times 128$

Critic network with ~477k free parameters

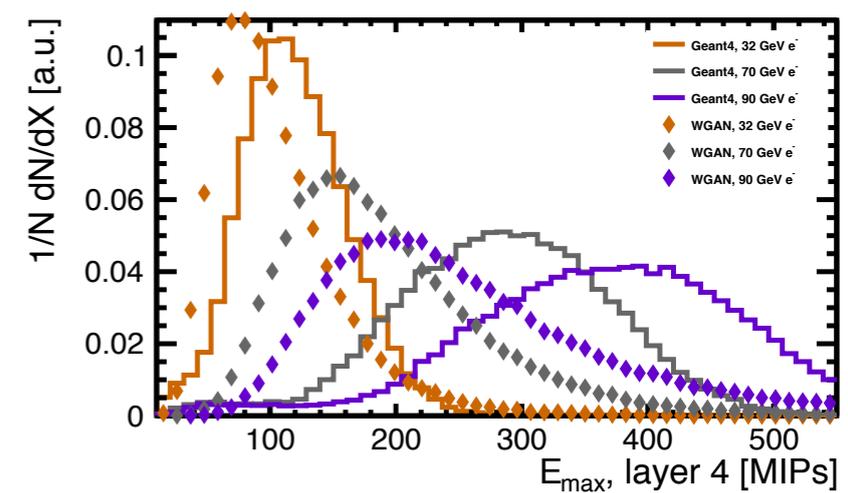
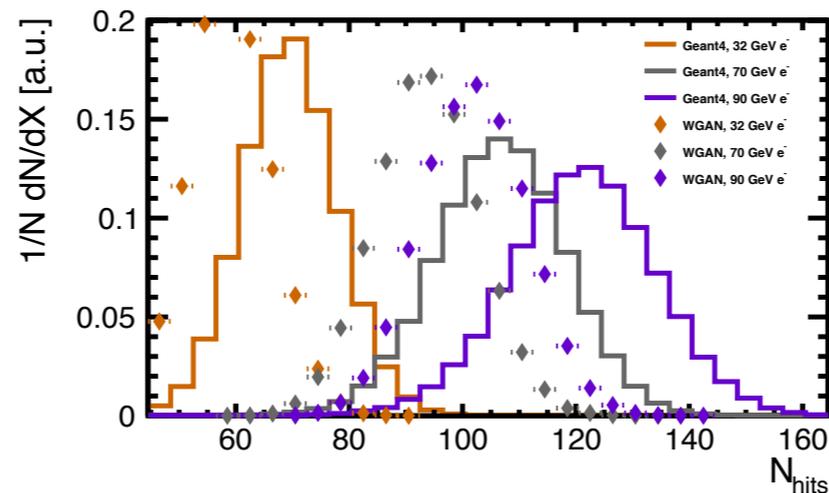
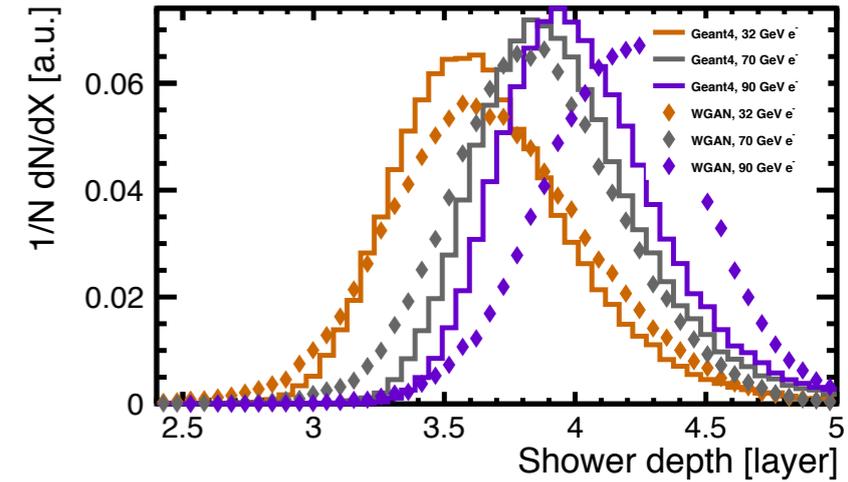
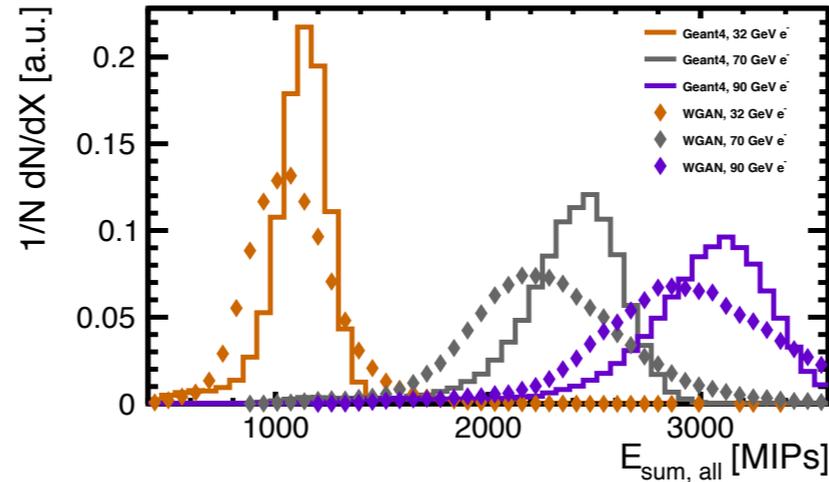


Logarithmic intensity: energy' = log(1+energy)

- ✓ Training converges.
- ...less smoothly.



- E_{max} mis-modeled!
- ...all others, too.

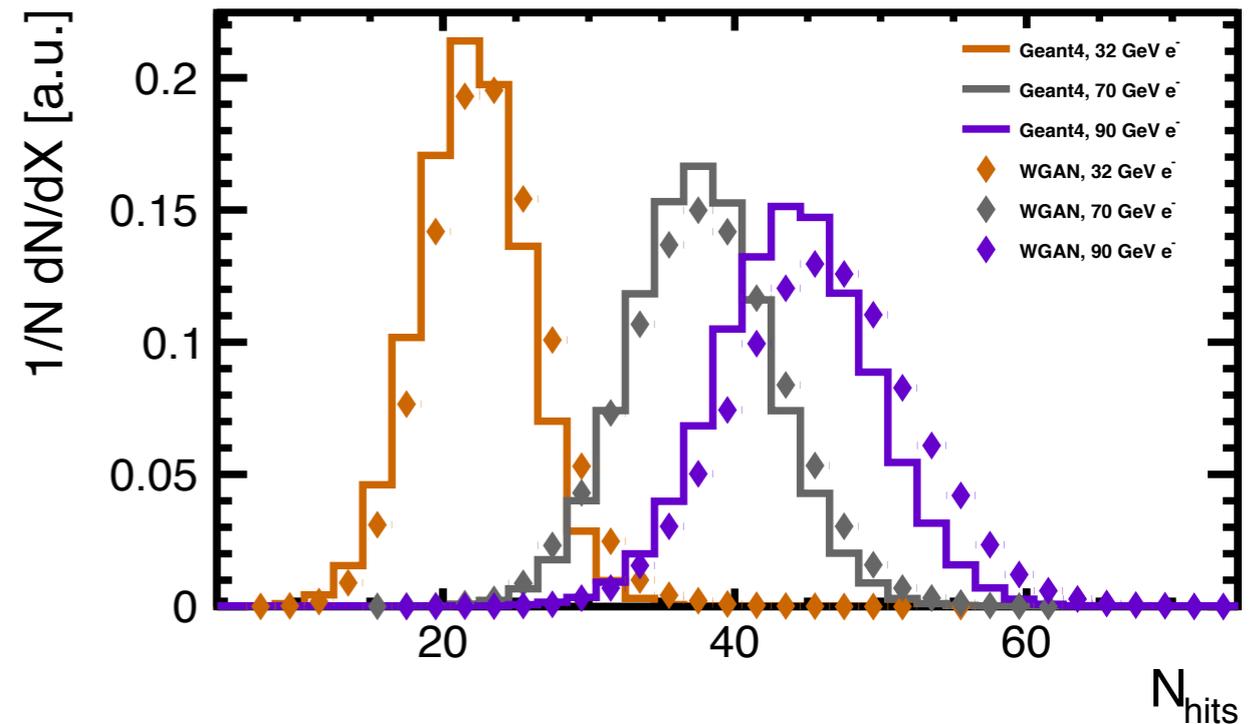
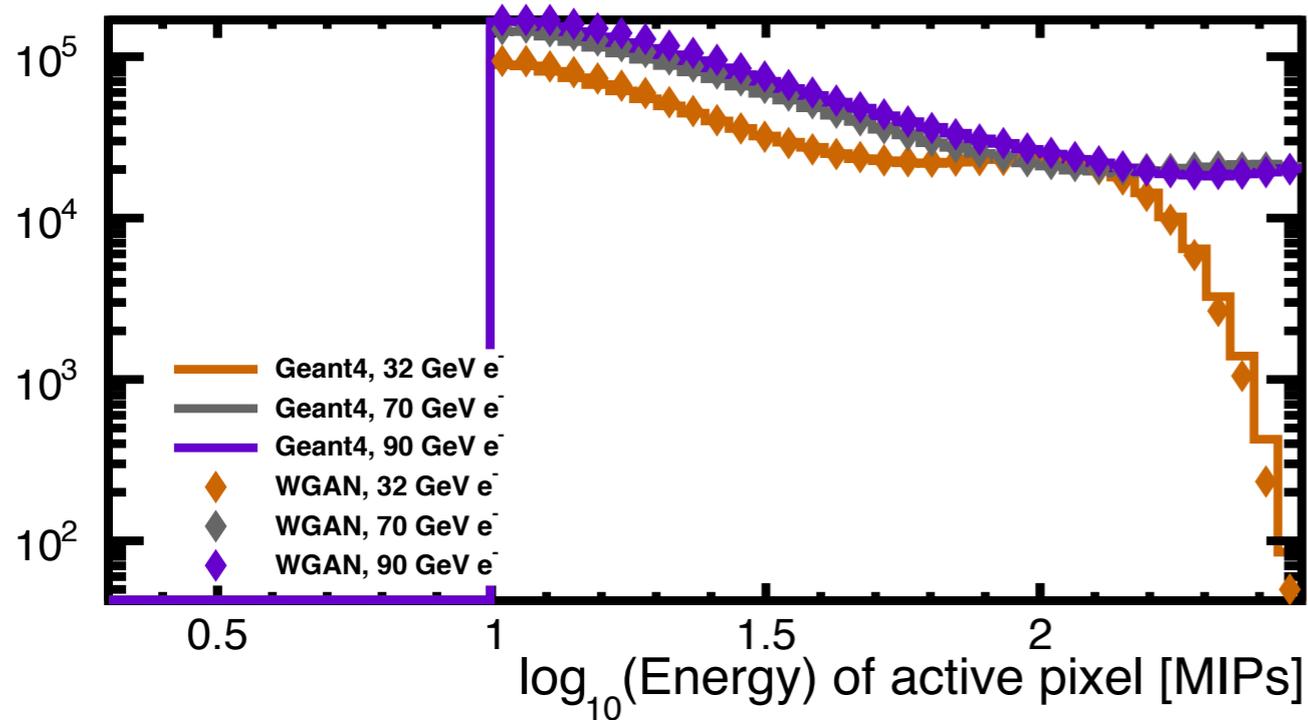


➔ No improvement.

*Supplementary benchmark:
10 MIP pixel cut at evaluation*

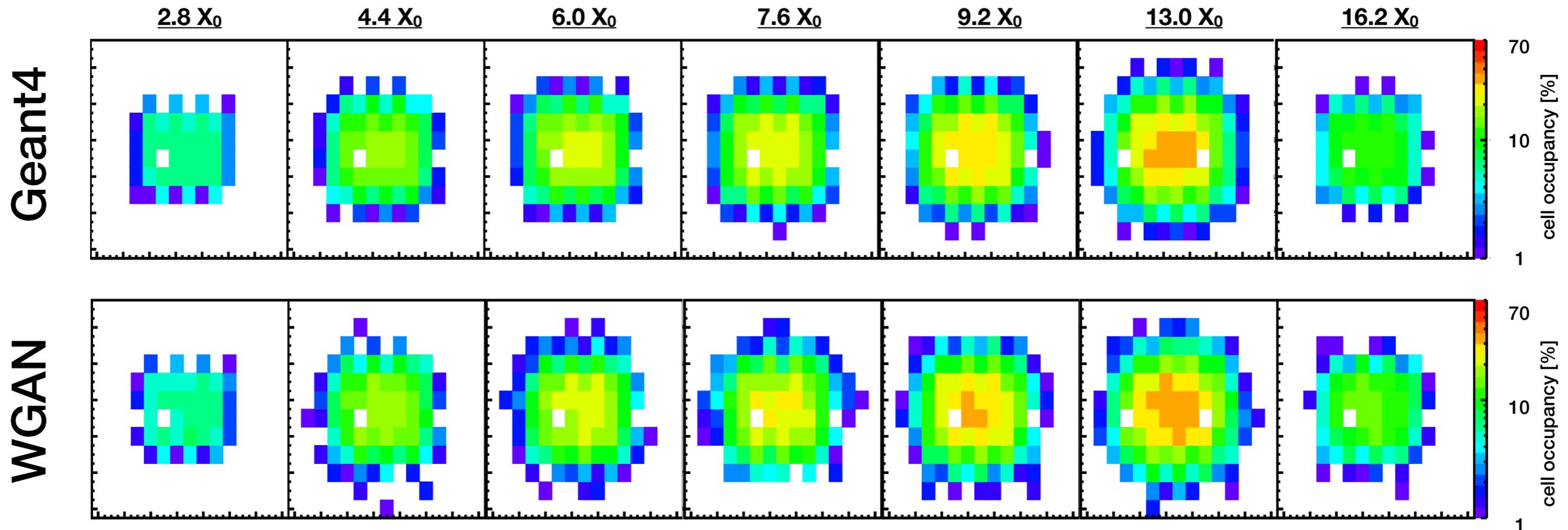
Supplementary benchmark: N_{hits}

10 MIP cut per pixel



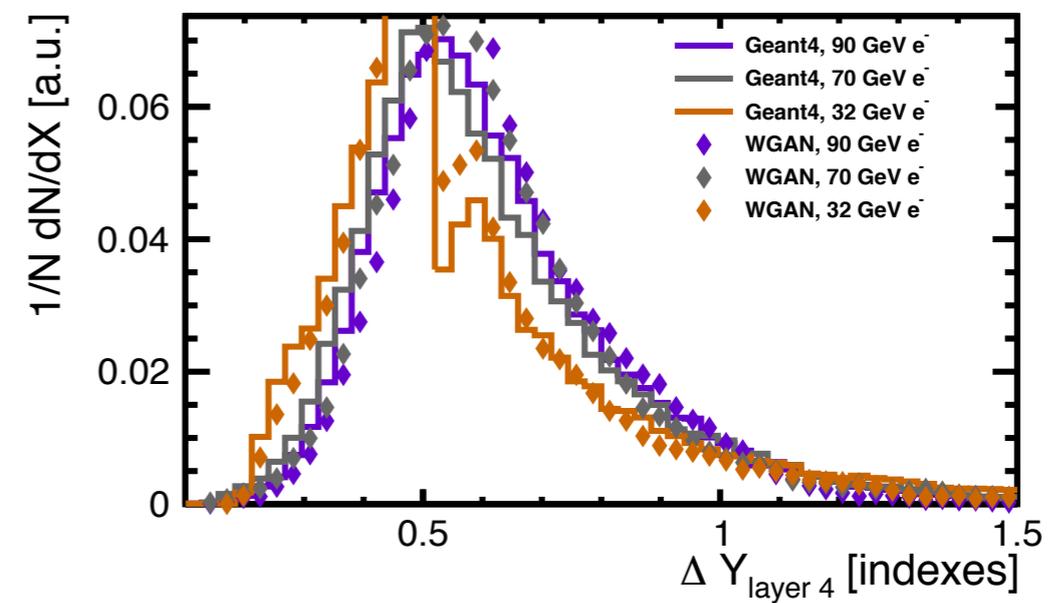
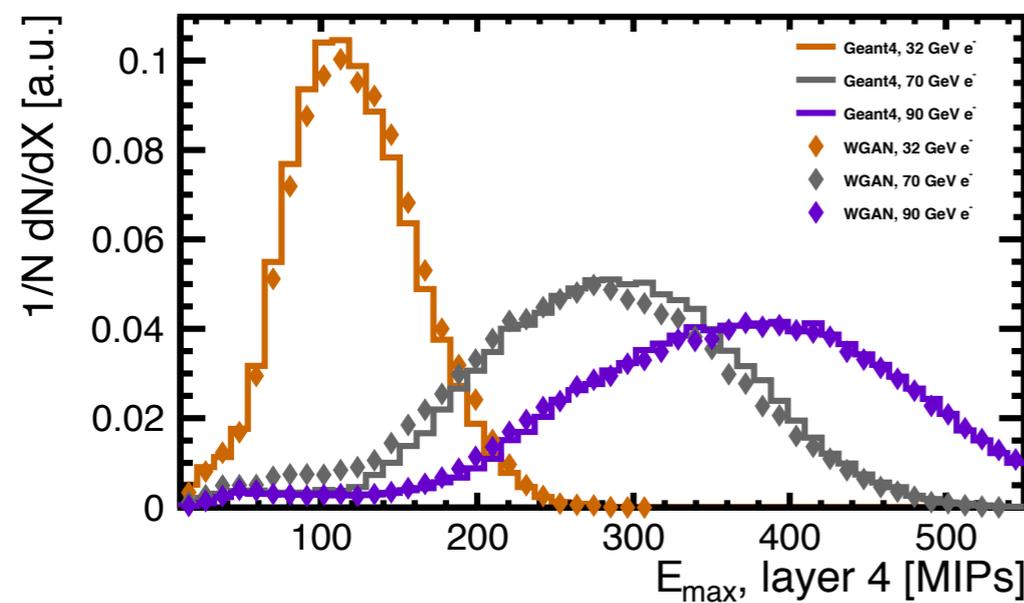
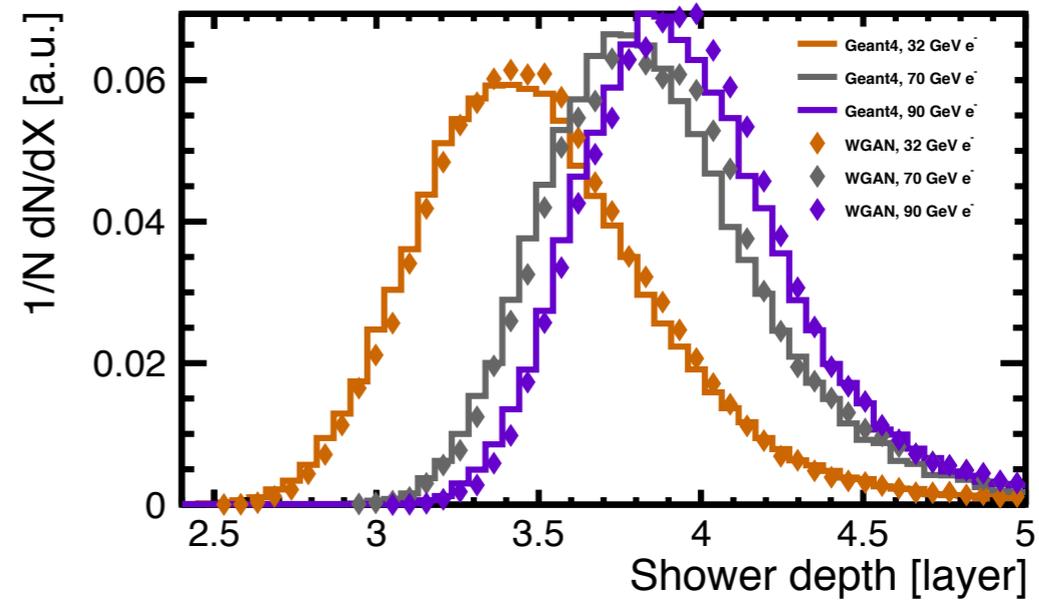
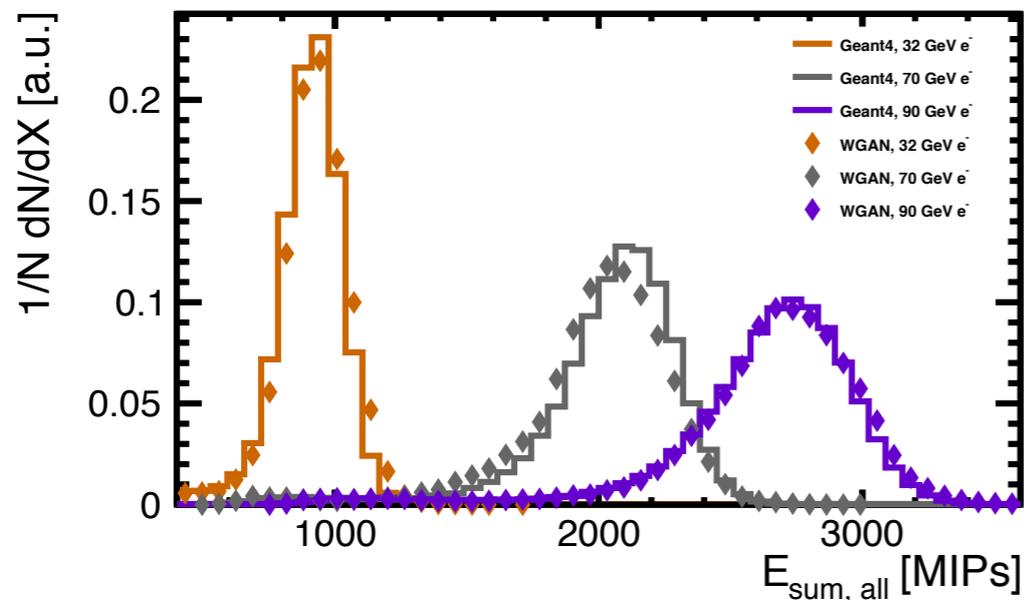
Supplementary benchmark: N_{hits}

10 MIP cut per pixel



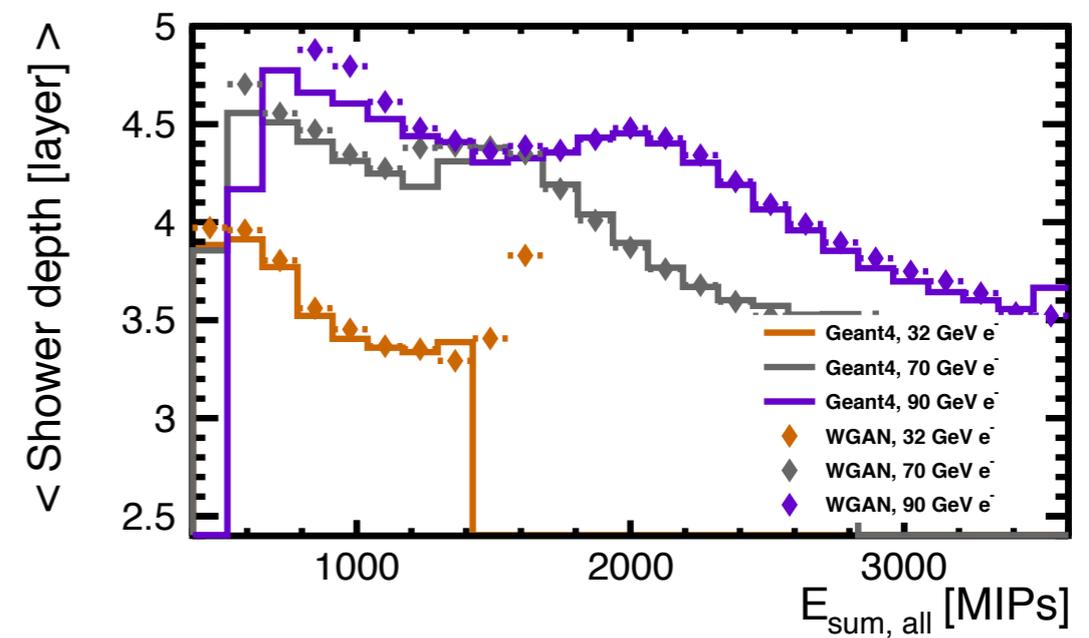
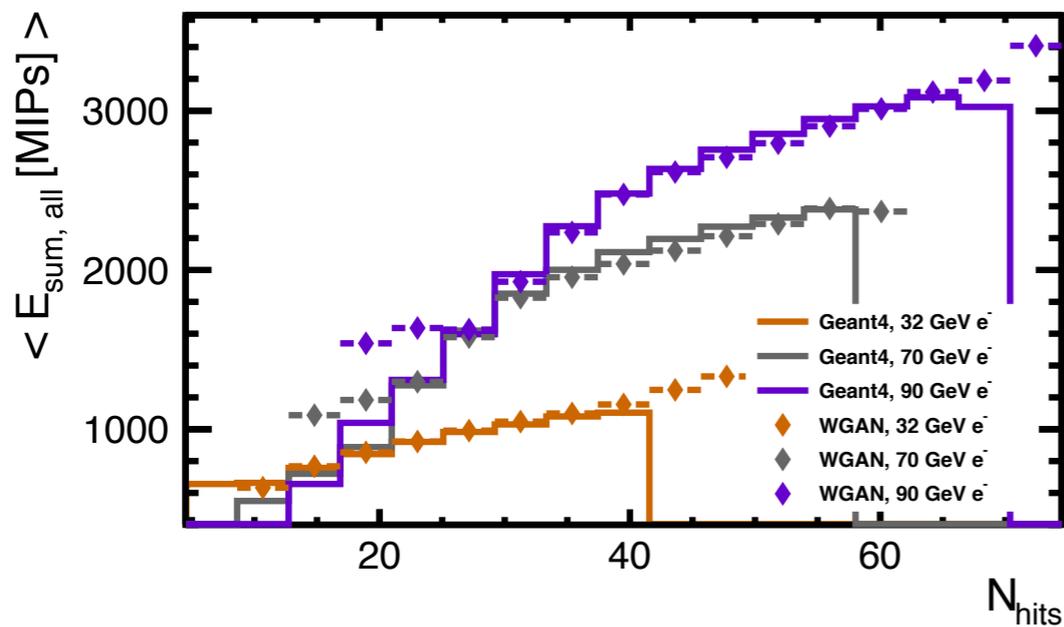
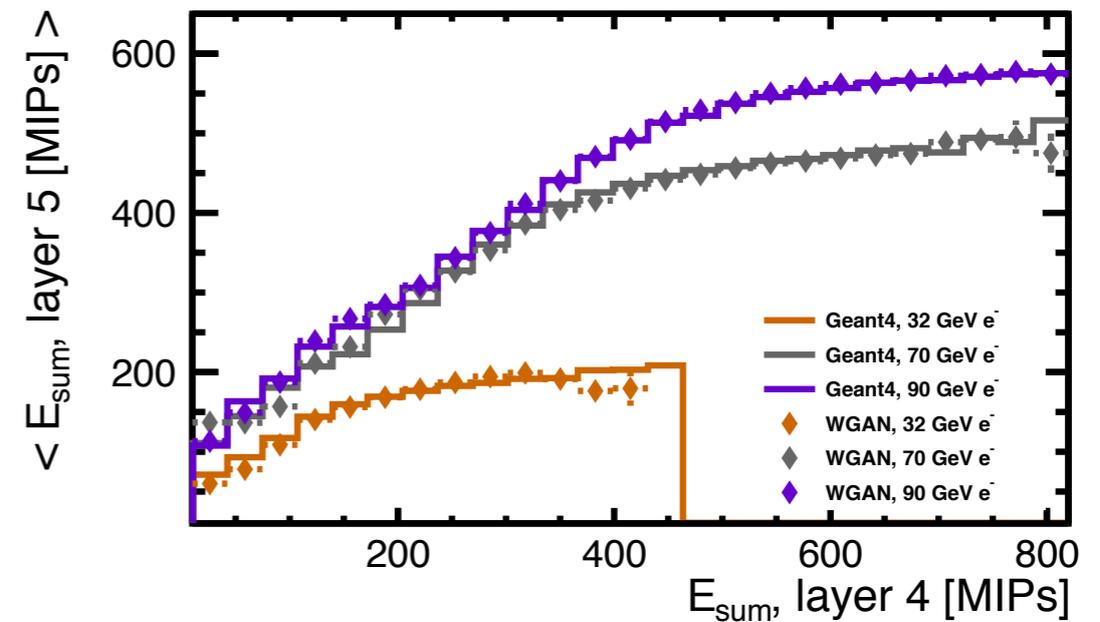
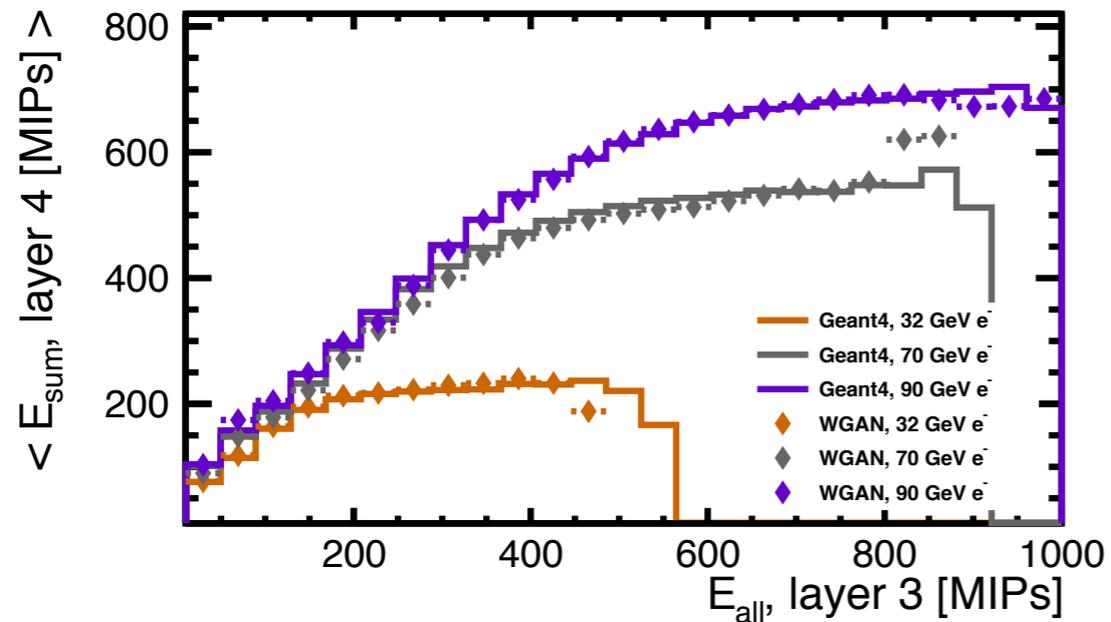
Supplementary benchmark: Observables

10 MIP cut per pixel



Supplementary benchmark: Correlations

10 MIP cut per pixel



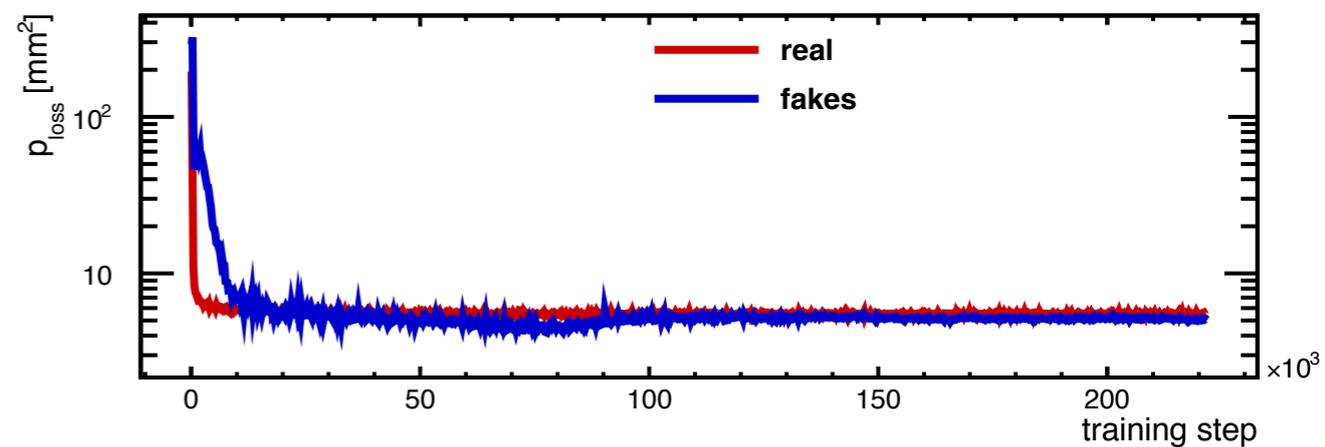
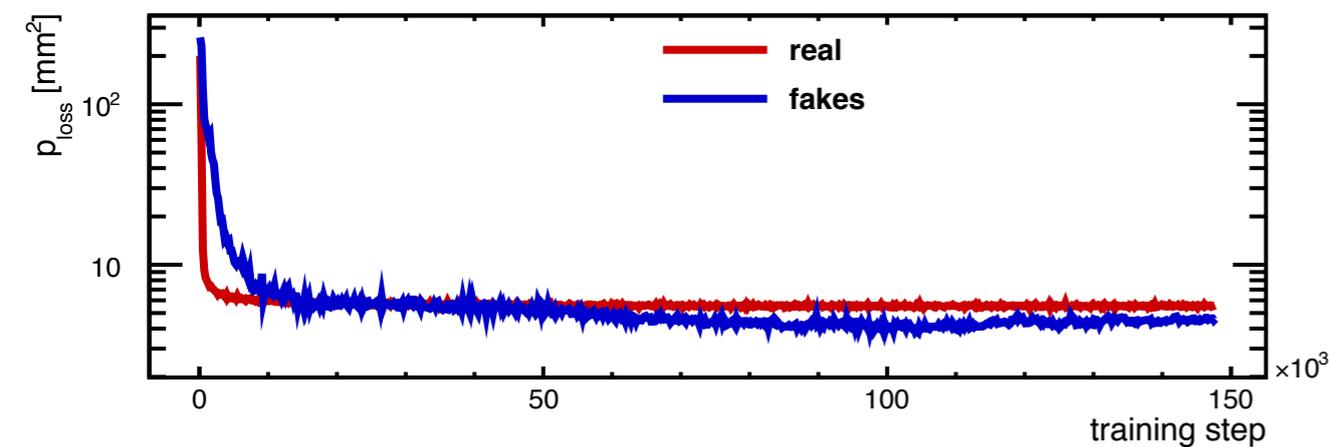
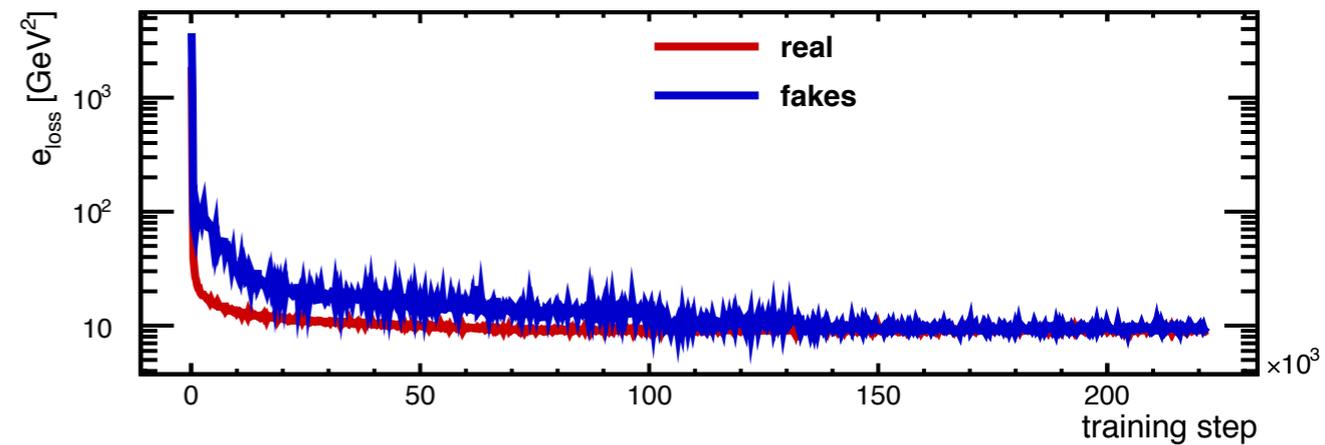
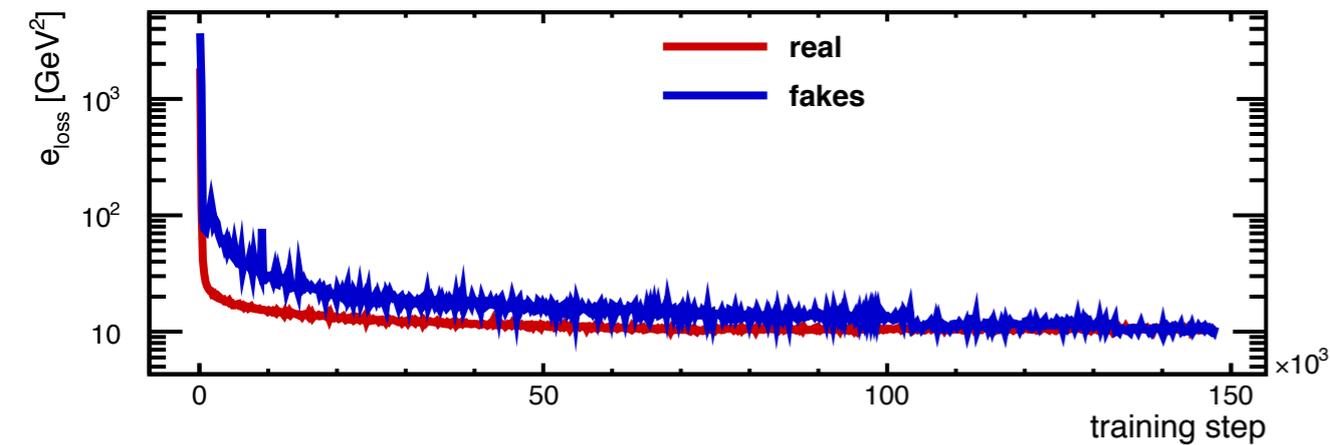
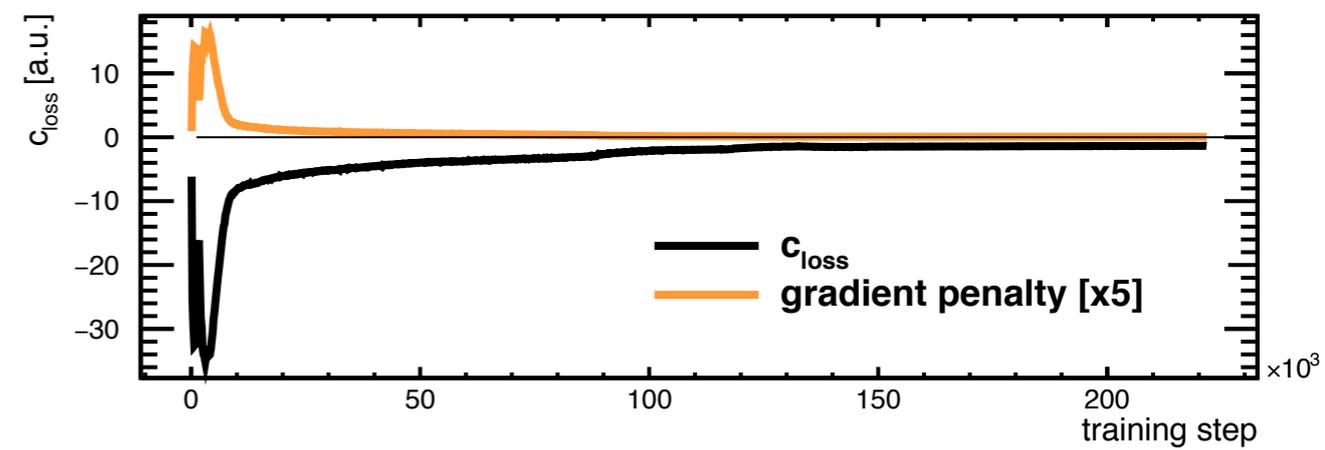
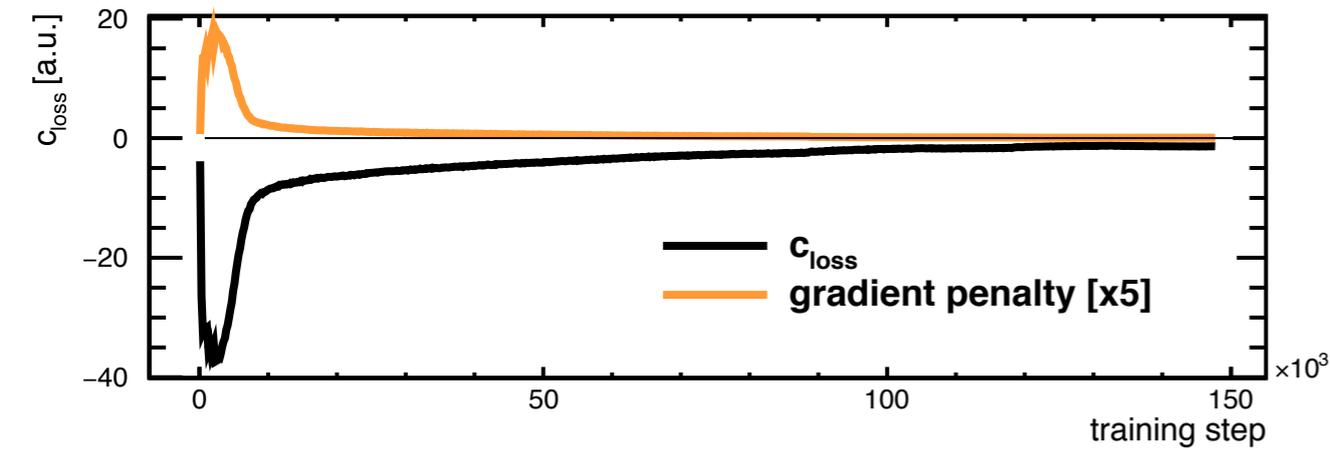
Do dead areas need to be masked?

Comparison: Costs

no masking of dead cells

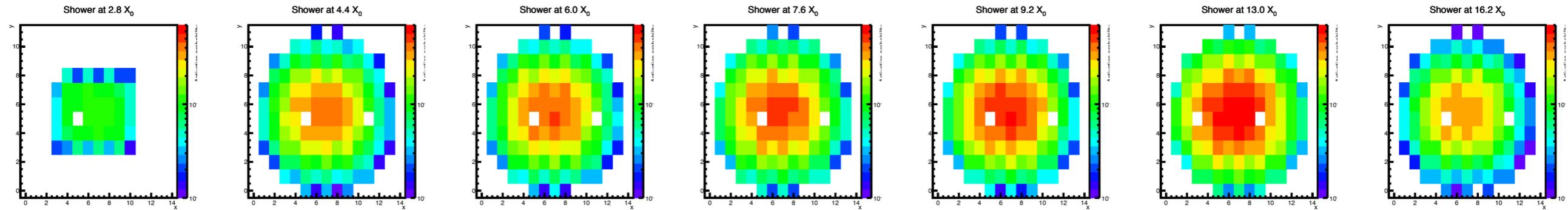
vs.

with masking of dead cells

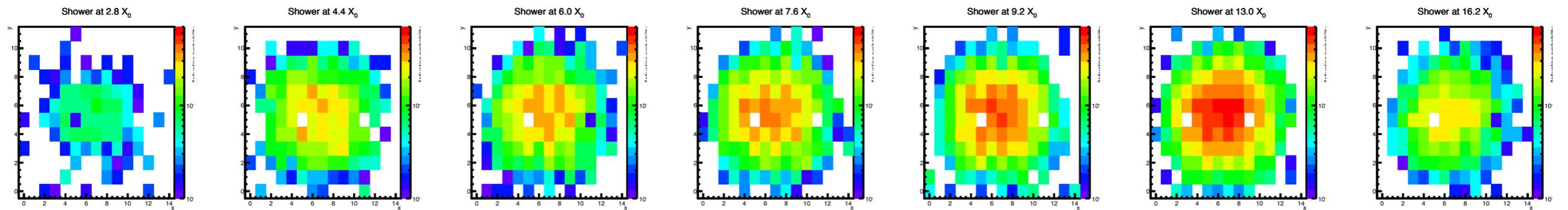


Comparison: Occupancy

90 GeV e- Geant4

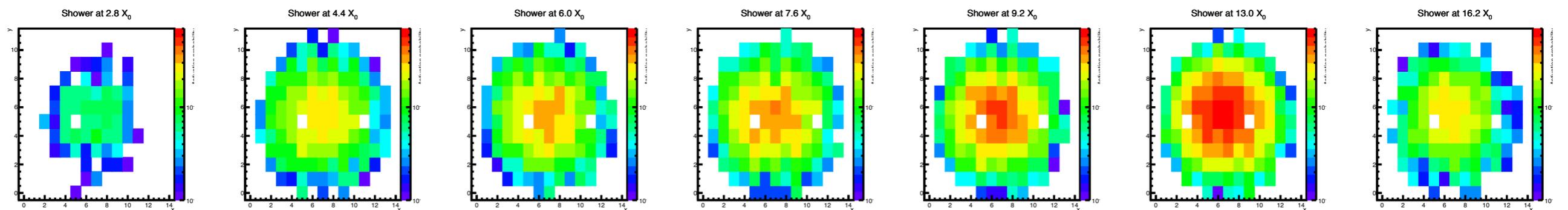


WGAN: no masking of dead cells



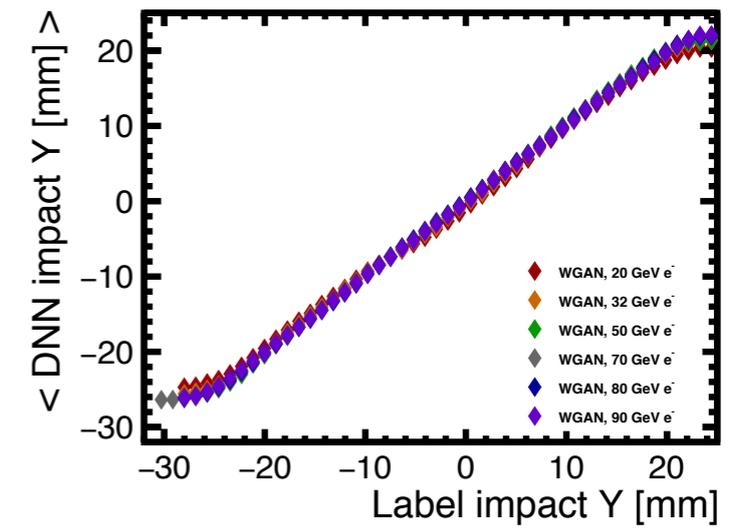
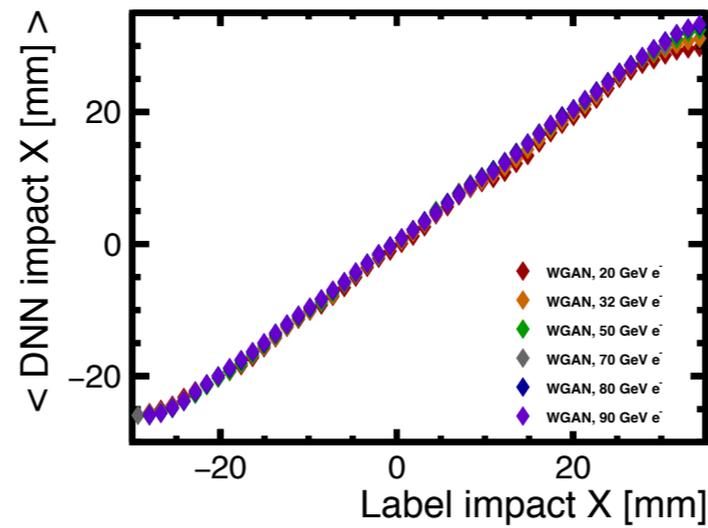
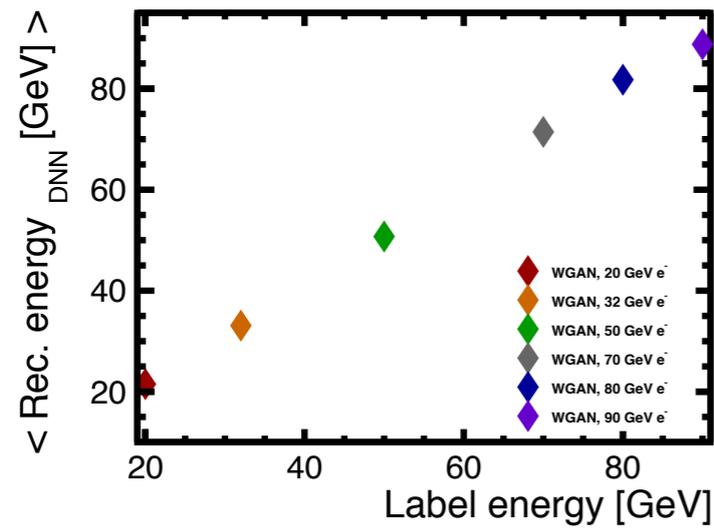
VS.

WGAN: with masking of dead cells



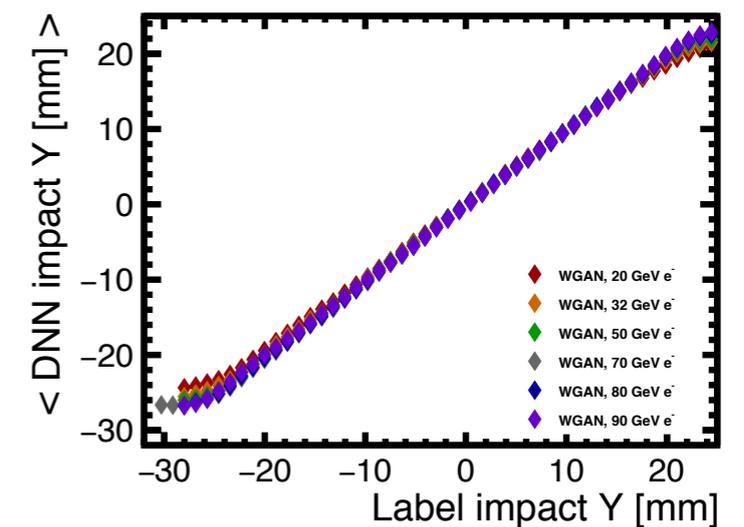
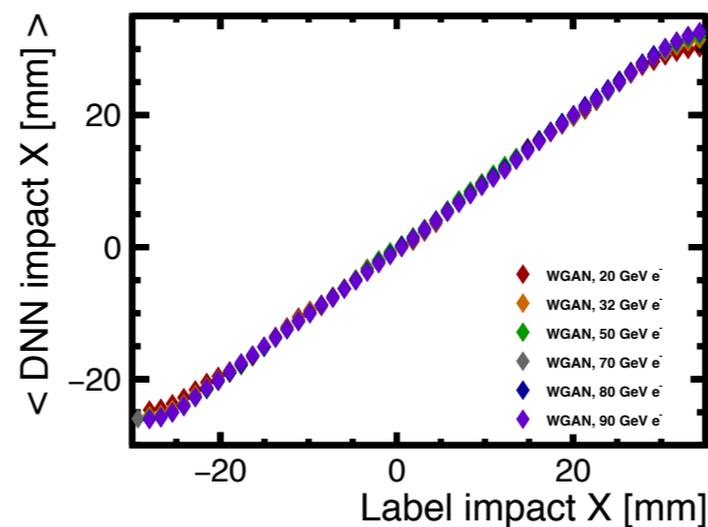
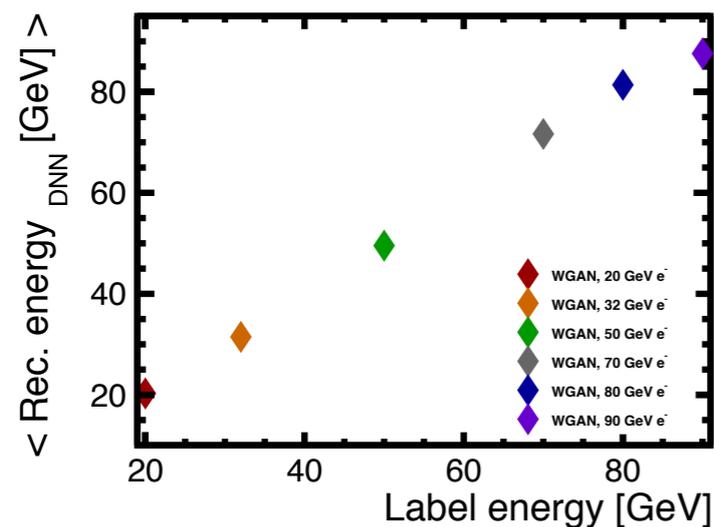
Comparison: Label dependence

no masking of dead cells

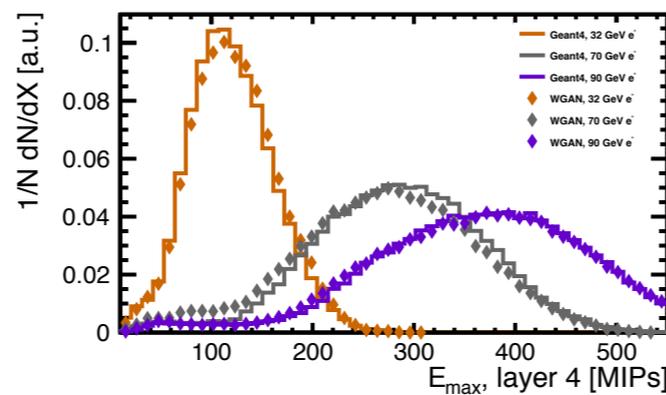
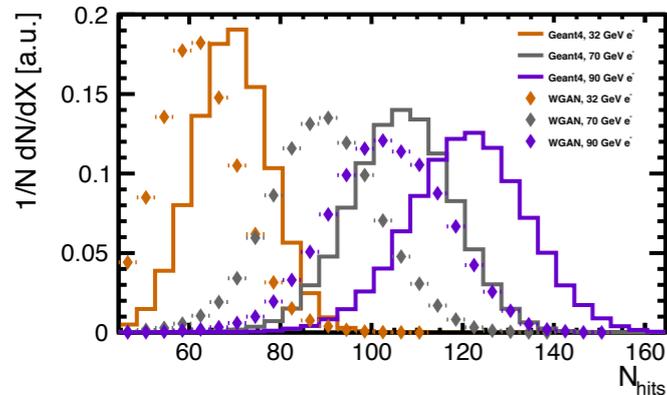
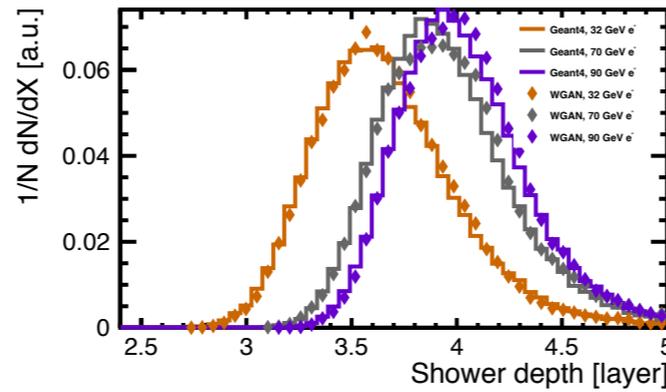
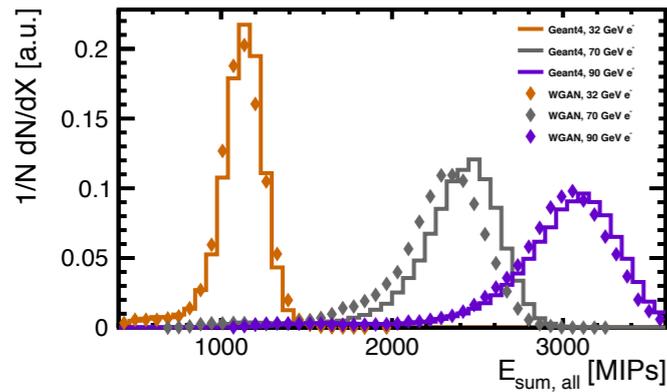


vs.

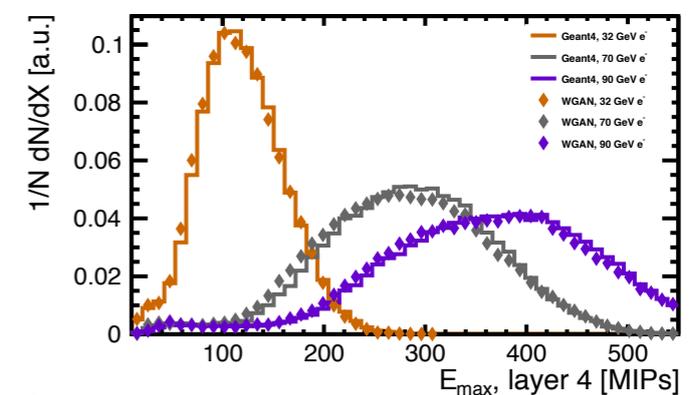
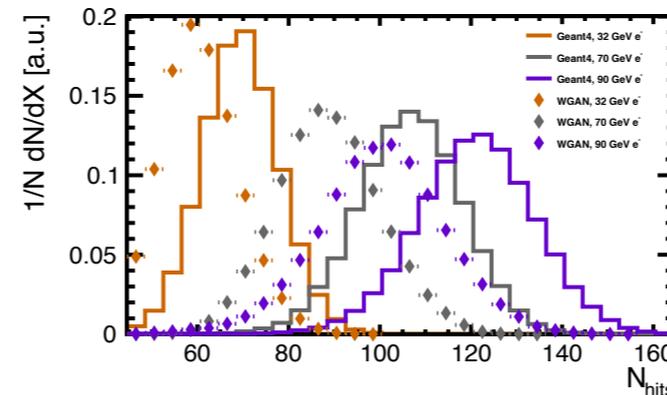
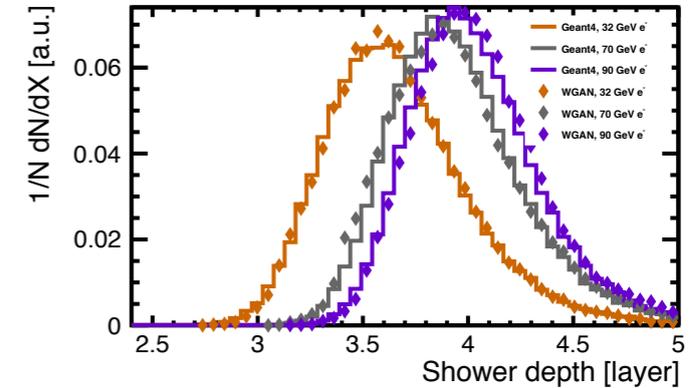
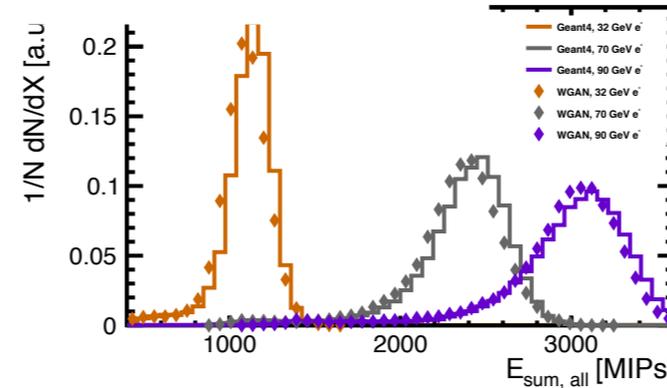
with masking of dead cells



Comparison: Observables

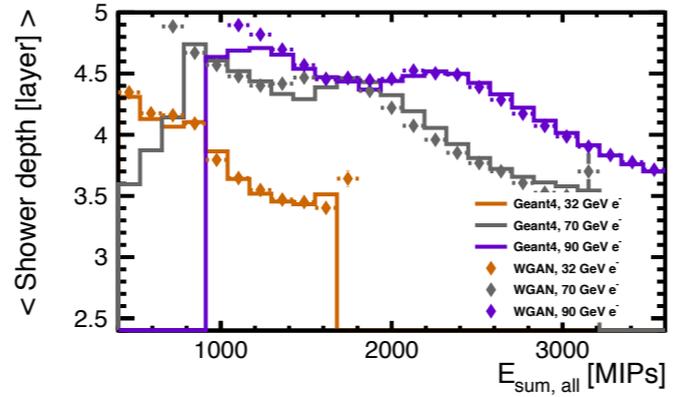
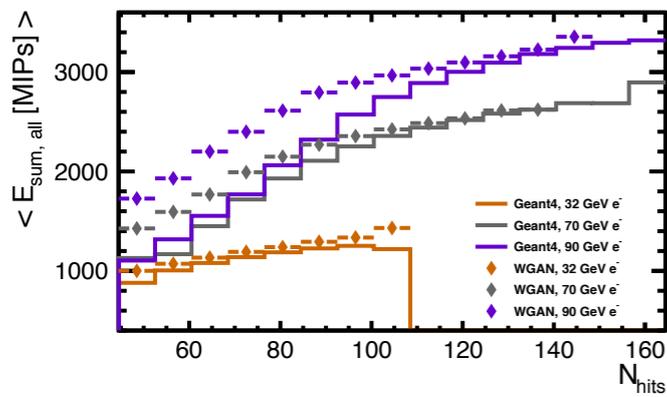
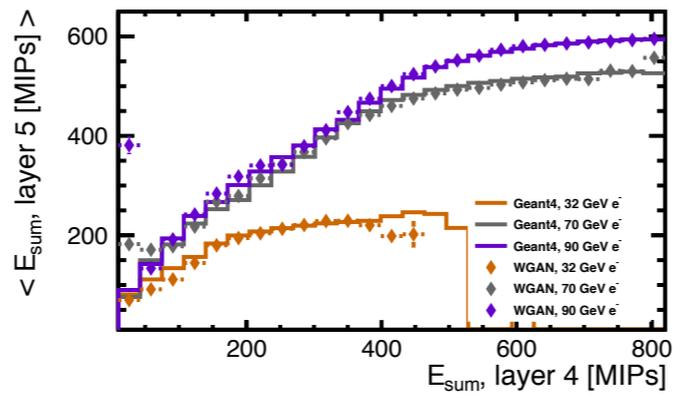
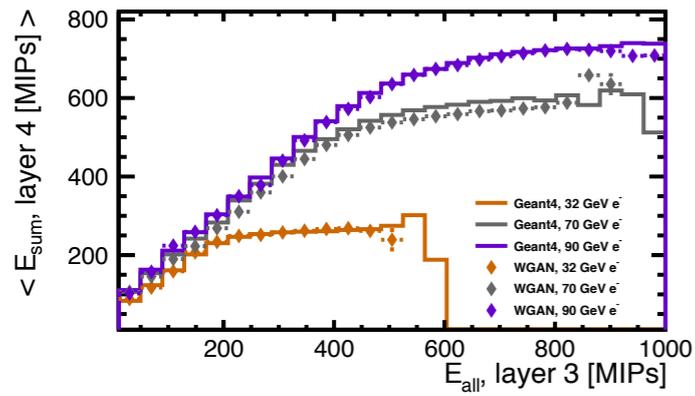


no masking of
dead cells

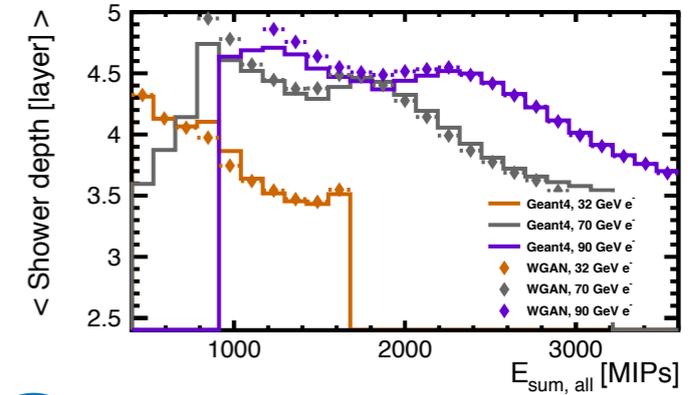
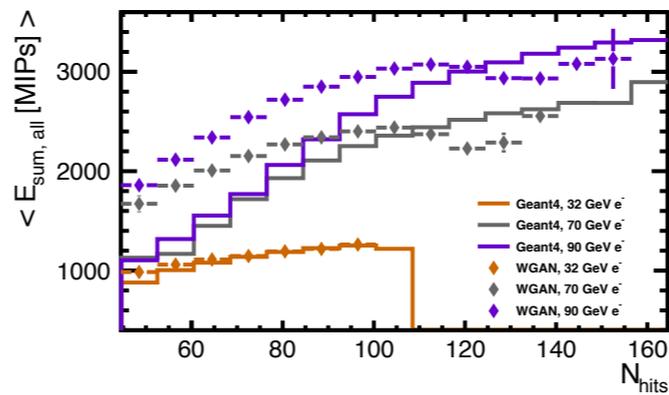
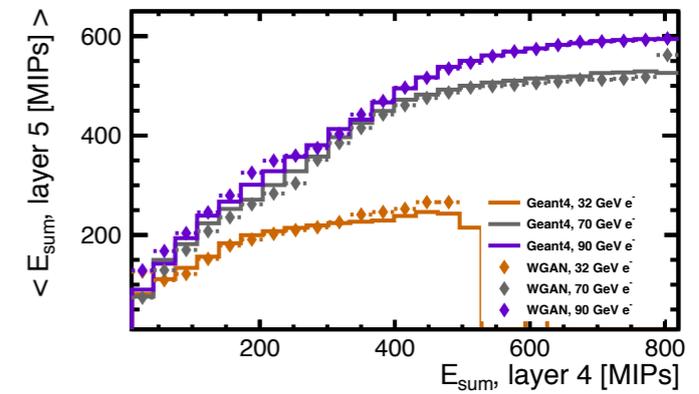
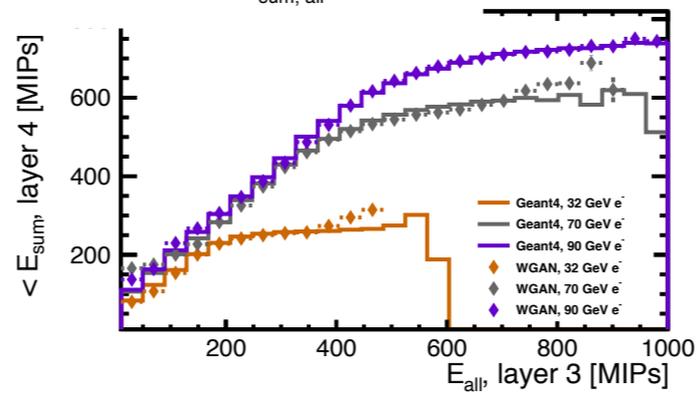


with masking of
dead cells

Comparison: Correlations



no masking of
dead cells



with masking of
dead cells