Generative Models for Fast Calorimeter Simulation

Вступ

Монте-Карло симуляції грають дуже важливу роль в дослідженнях та вимагають значних обчислювальних ресурсів.

Щорічно, симуляції для LHC вимагають мільярдів CPU-годин.

Симуляції калориметрів є найбільш обчислювально витратною частиною.

Вступ

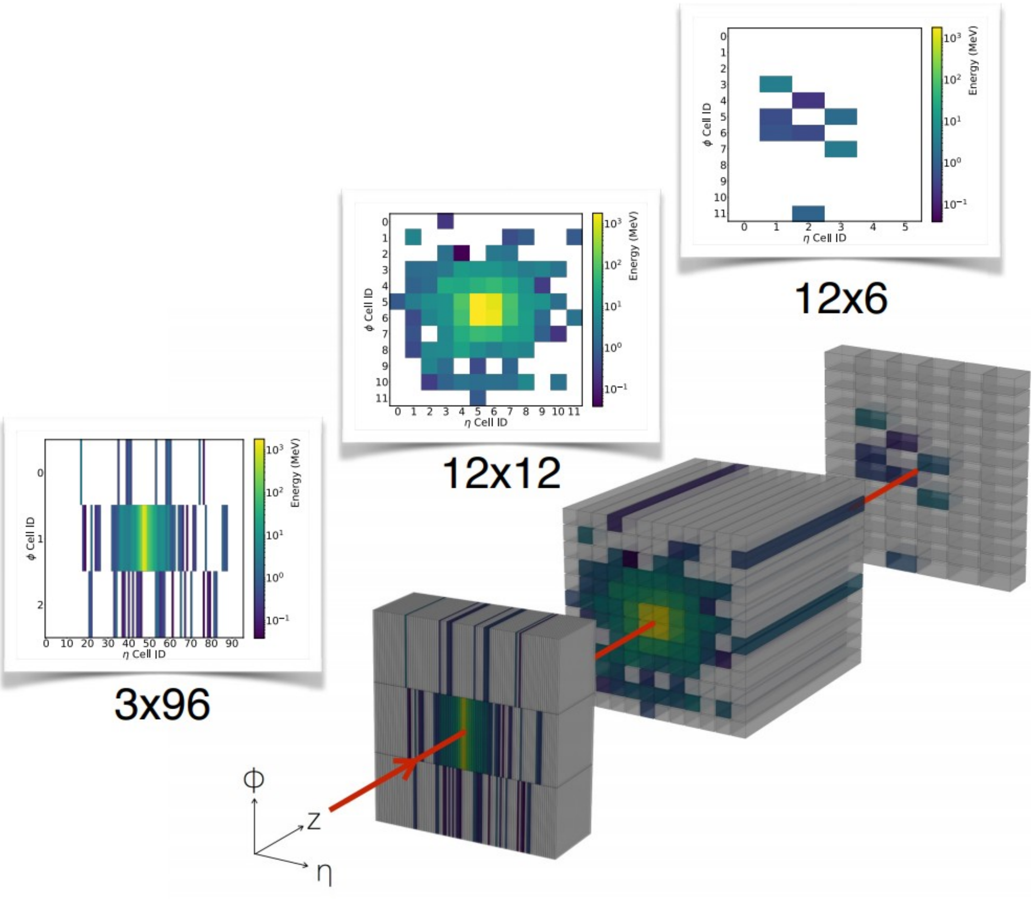
Параметричні моделі – пришвидшення ~10 разів.
Мінуси:

- Обмежені конкретним експериментом
- Гірша точність
- Недостатнє пришвидшення

CaloGAN

ArXiv:1705.02355 (2017)

Модель симулювала виділення енергії в калориметрі з гетерогенною повздовжньою та поперечною сегментацією.

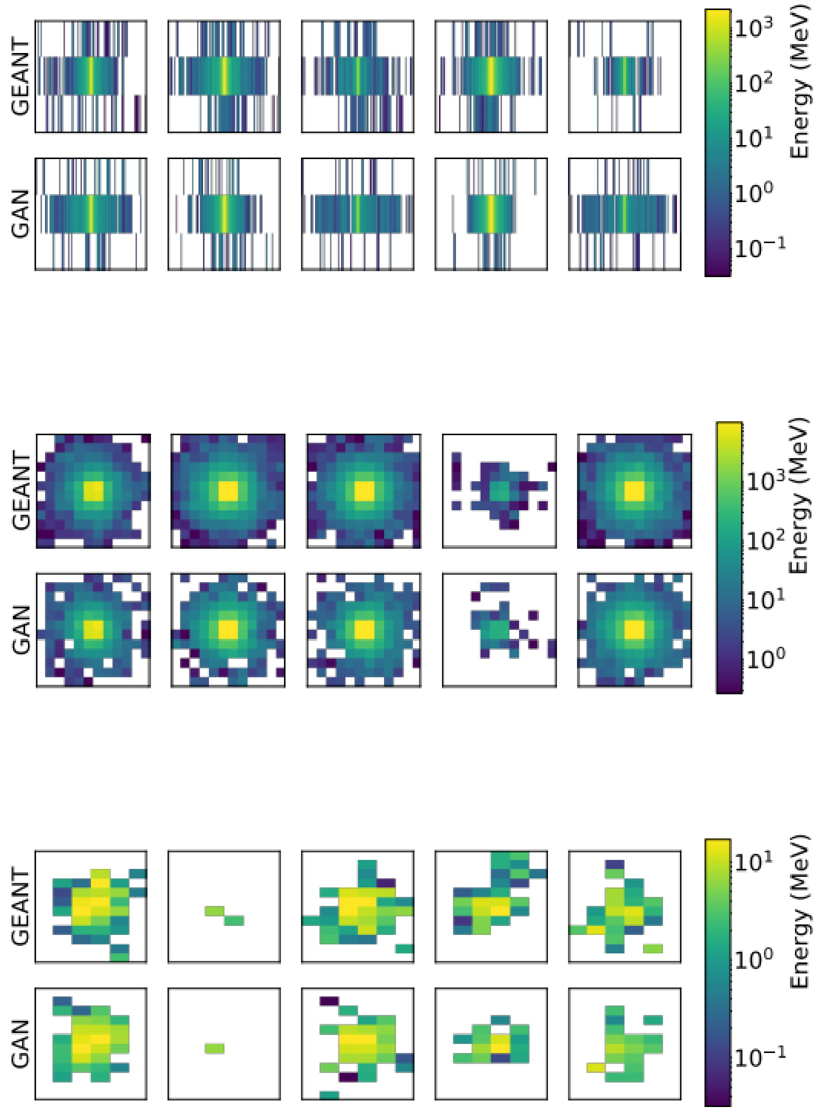


CaloGAN

Модель представляла собою DCGAN, де деякі згорткові шари були замінені на локально-зв’язані шари.

Модель була натренована на електронах, протонах та піонах з енергіями від 1 до 100 ГеВ.

CaloGAN



CaloGAN

- Найбільше отримане прискорення ~10^5
- Дуже простий сетап – симуляції параметризувались лише енергією.
- Спостерігались істотні відмінності з Geant4 симуляціями

Відстань між розподілами

Нехай *X* – компактна метрична множина та *Σ* – множина всіх підмножин Бореля *X*. Нехай *Prob(X)* позначає міру ймовірності на *X*, тоді можна ввести відстань між розподілами P_r та P_g:

- Total variation distance

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A \in \Sigma} |\mathbb{P}_r(A) - \mathbb{P}_g(A)|$$

- The Kullback-Leibler divergence

$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x)$$

Відстань між розподілами

- The Jensen-Shannon divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m)$$

де $P_m = (P_r + P_g)/2$

- The Earth-Mover distance or Wasserstein-1

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\|x - y\| \big]$$

$\Pi(P_r, P_g)$ - множина всіх об'єднаних розподілів $\gamma(x,y)$ чиї відособлені розподіли P_r, P_g

GAN

Генератор та дискримінатор почергово грають гру за наступним правилом:

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

Коли генератор зафіксований, оптимальне значення може бути записано аналітично:

$$\max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}[\log(1 - D(\mathbf{x}))] = \text{JS}(p_{\text{data}} \parallel p_G)$$

де JS – розбіжність Дженсена-Шеннона (Jensen–Shannon divergence)

GAN

Коли зафіксований дискримінатор, генератор буде підганяти свій розподіл:

$$\min_G \mathrm{JS}(p_{\mathrm{data}} \,\|\, p_G)$$

Гру дискримінатора та генератора можна переформулювати використовуючи іншу відстань:

$$\min_G \mathcal{D}(p_{\mathrm{data}} \,\|\, p_G)$$

Wasserstein GAN (WGAN)

arXiv:1701.07875

В 2017-му році було запропоновано використовувати відстань Вассерштайна. Її можна записати як:

$$W(p_{\text{data}} \,\|\, p_G) = \max_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_G}[f(\mathbf{x})]$$

де f - 1-Ліпшиць функції *f: X→ R*

Wasserstein GAN (WGAN)

Матимемо наступну гру дискримінатора та генератора:

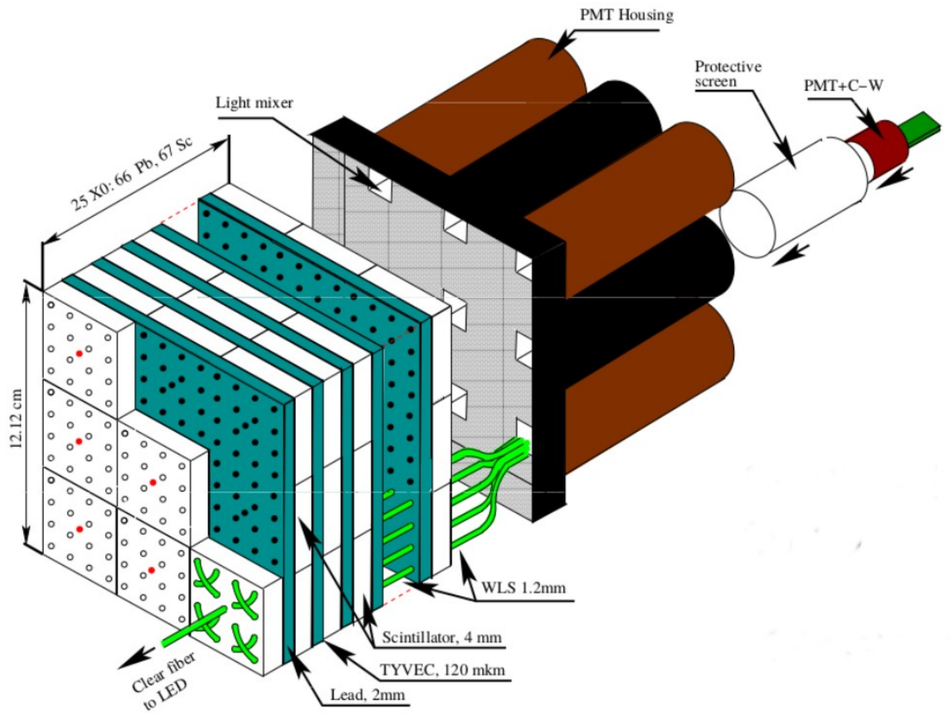
$$\min_G \max_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x})}[f(\mathbf{x})]$$

З властивостей функцій Ліпшиця:

$$\min_G \max_f \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p_G(\mathbf{x})} f(\mathbf{x}) + \lambda \mathbb{E}_{\mathbf{x} \sim p_G} (\|\nabla_{\tilde{\mathbf{x}}} D(\tilde{\mathbf{x}})\|_2 - 1)^2$$

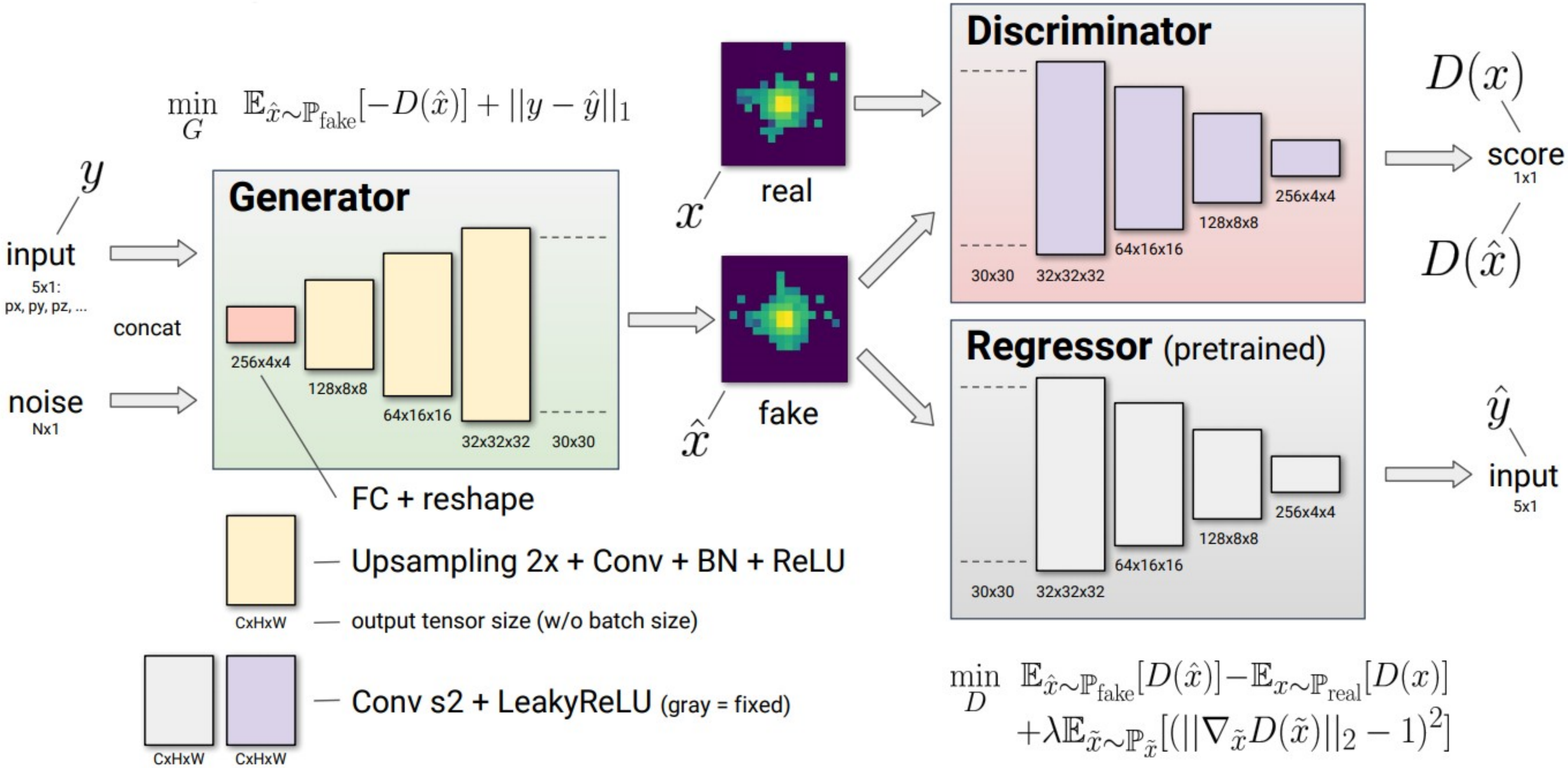
Геометрія симуляцій

5x5 блоків 12x12 см з розміром комірки 2x2 см



Геометрія LHCb

Модель



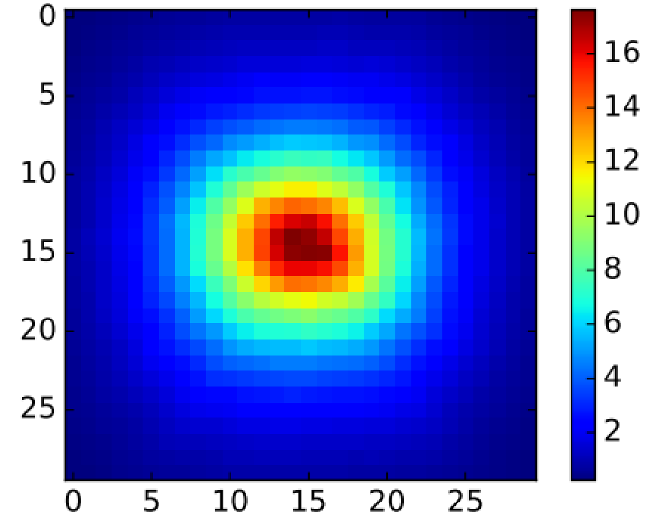
$\min_G \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[-D(\hat{x})] + ||y-\hat{y}||_1$
y
input
5x1:
px, py, pz, ...
concat
noise
Nx1
Generator
256x4x4
128x8x8
64x16x16
32x32x32
30x30
FC + reshape
Upsampling 2x + Conv + BN + ReLU
CxHxW
output tensor size (w/o batch size)
CxHxW
CxHxW
Conv s2 + LeakyReLU (gray = fixed)
x
real
$\hat{x}$
fake
Discriminator
30x30
32x32x32
64x16x16
128x8x8
256x4x4
$D(x)$
score
1x1
$D(\hat{x})$
Regressor (pretrained)
30x30
32x32x32
64x16x16
128x8x8
256x4x4
$\hat{y}$
input
5x1
$\min_D \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[D(\hat{x})] - \mathbb{E}_{x\sim\mathbb{P}_{\text{real}}}[D(x)]$
$+\lambda\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(||\nabla_{\tilde{x}}D(\tilde{x})||_2-1)^2]$

Тренування

Енергія: 1-100 ГеВ, (~1/E)

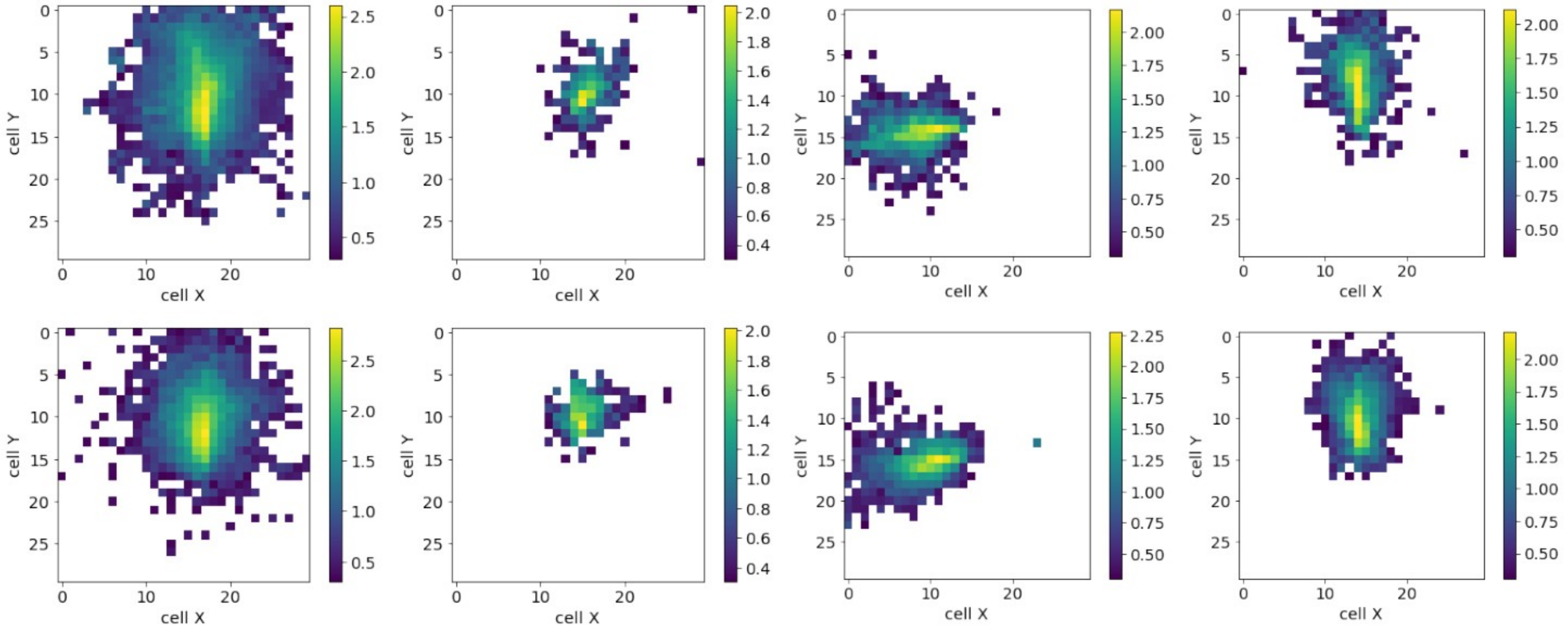
x,y: 1x1 см
Кути: 20 deg (XZ), 10 deg (YZ)

train set: 50000 events
test set: 10000 events



Щоб згладити дані та зробити оптимізацію більш стабільною, робиться перетворення Бокса-Кокса над данними.

Порівняння



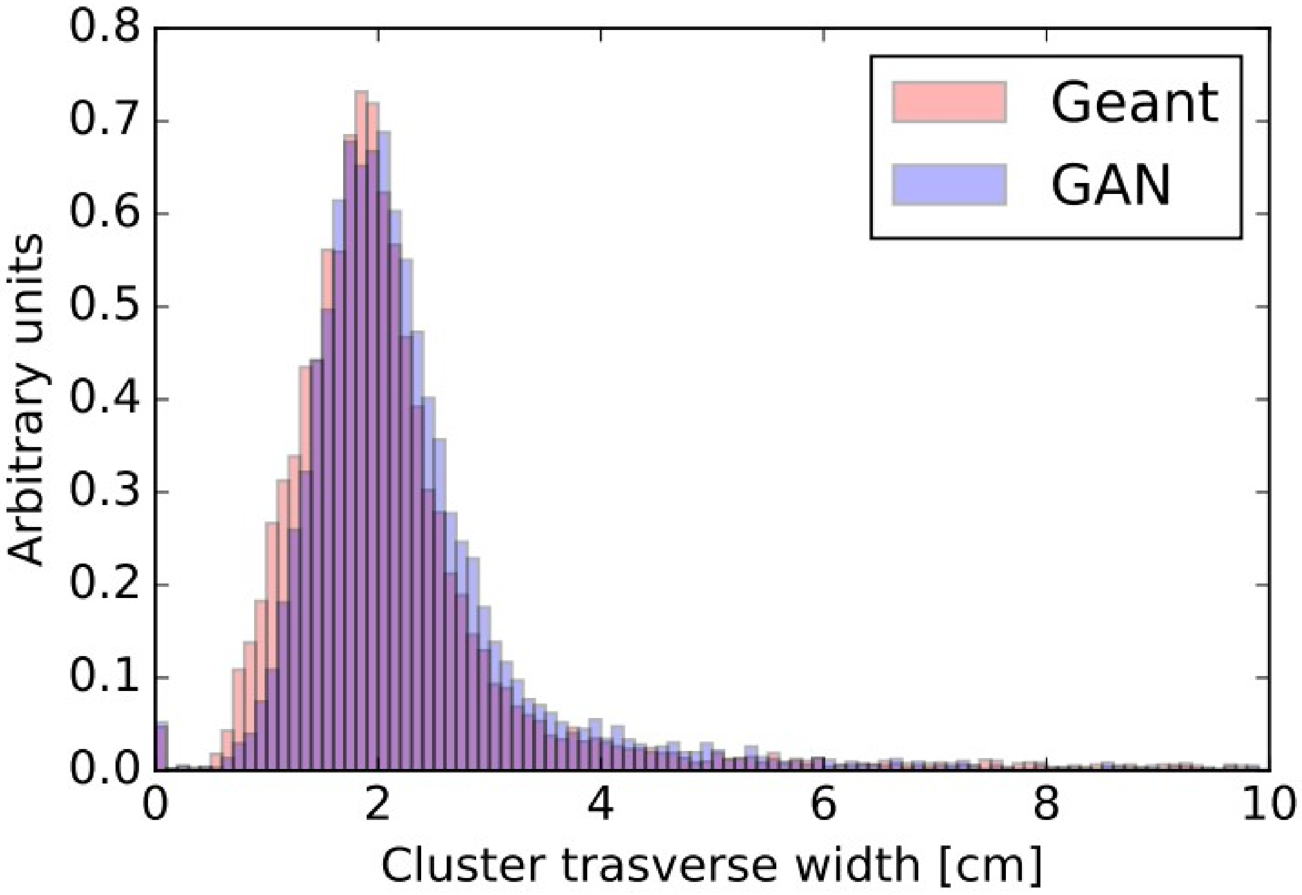
(a) $E_0 = 63.7$ GeV

(b) $E_0 = 6.5$ GeV

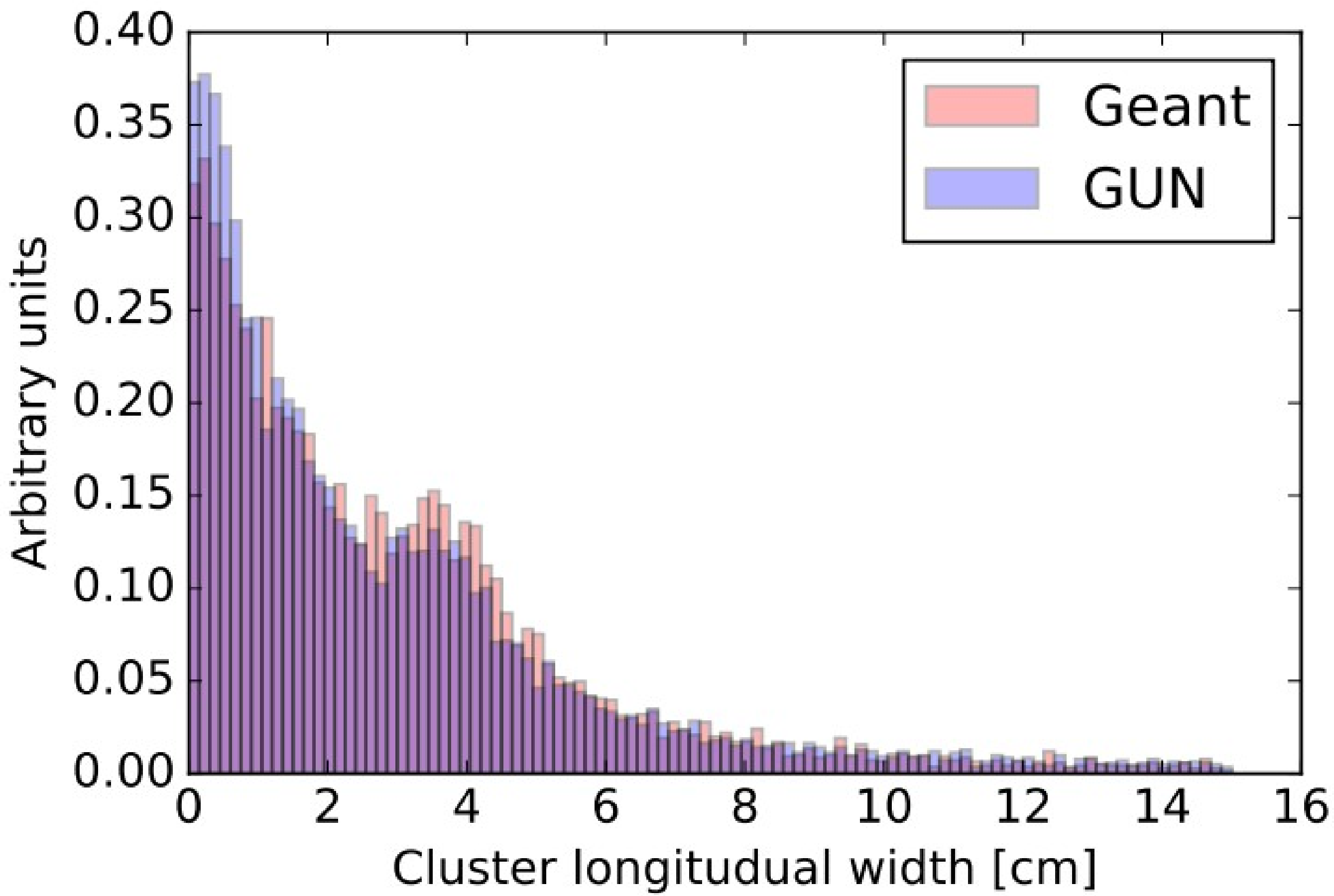
(c) $E_0 = 15.6$ GeV

(d) $E_0 = 15.9$ GeV

Порівняння

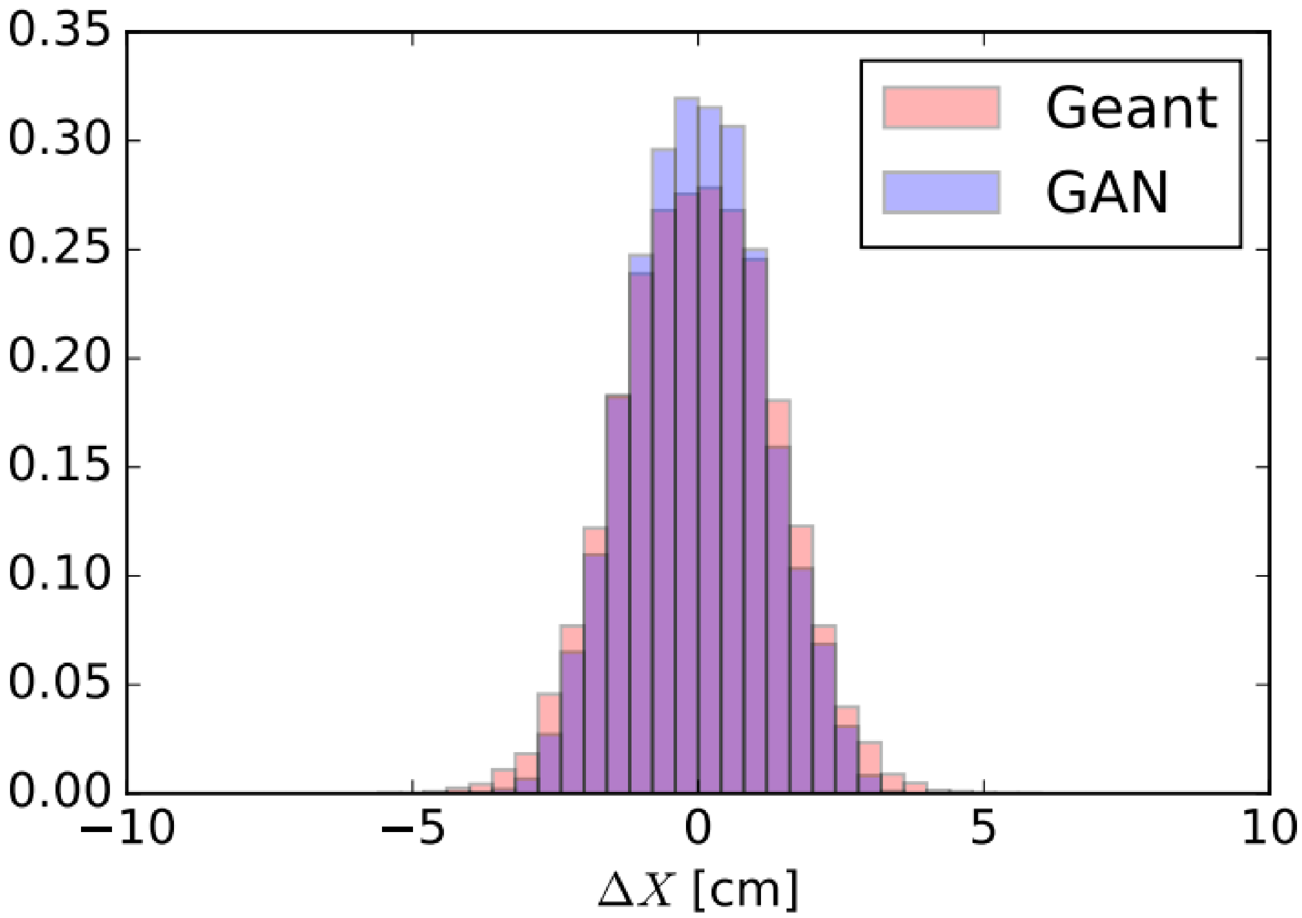

Geant
GAN
Arbitrary units
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
2
4
6
8
10
Cluster trasverse width [cm]

Порівняння

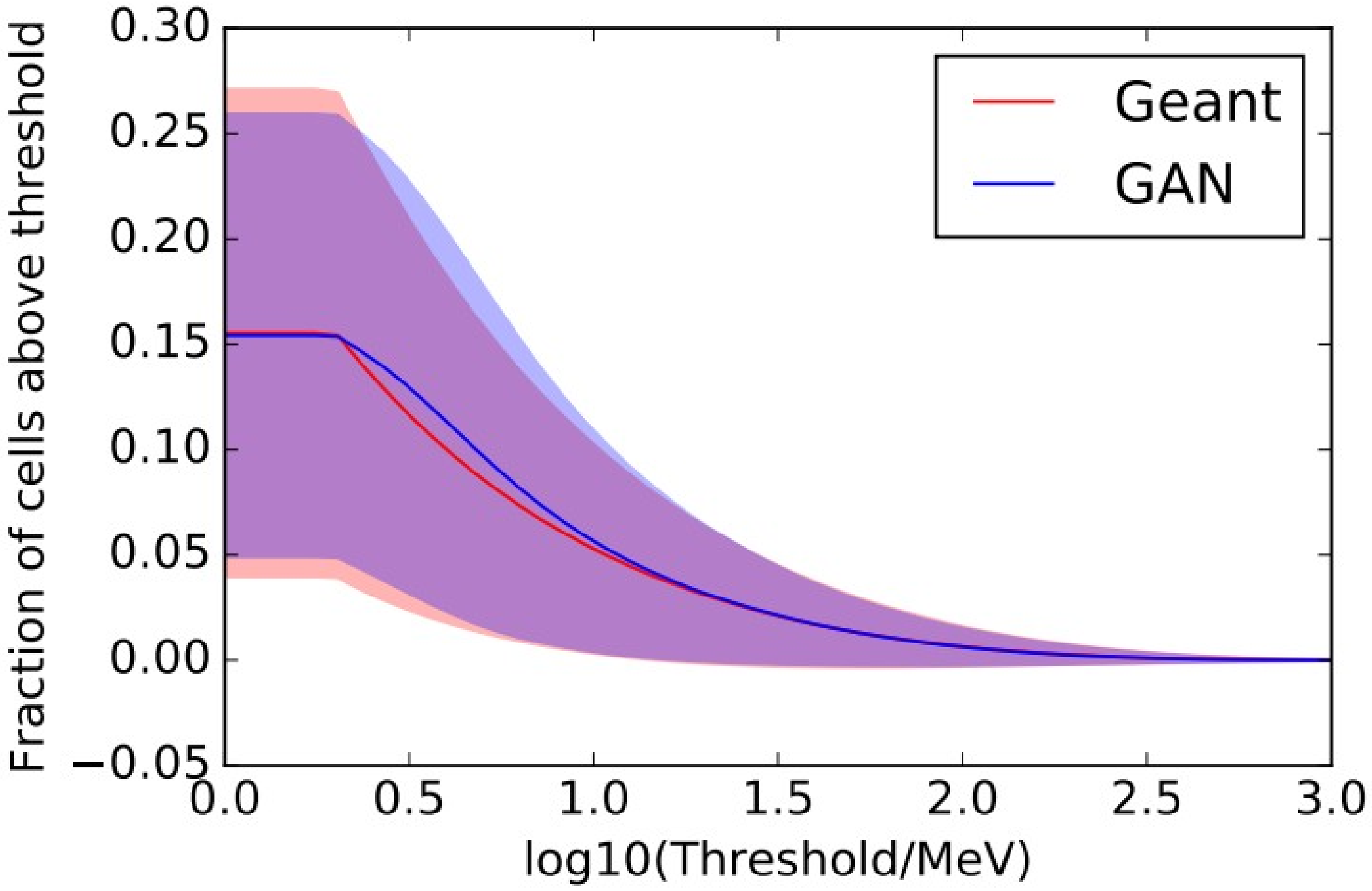


Geant
GUN
Arbitrary units
Cluster longitudual width [cm]
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
14
16

Порівняння

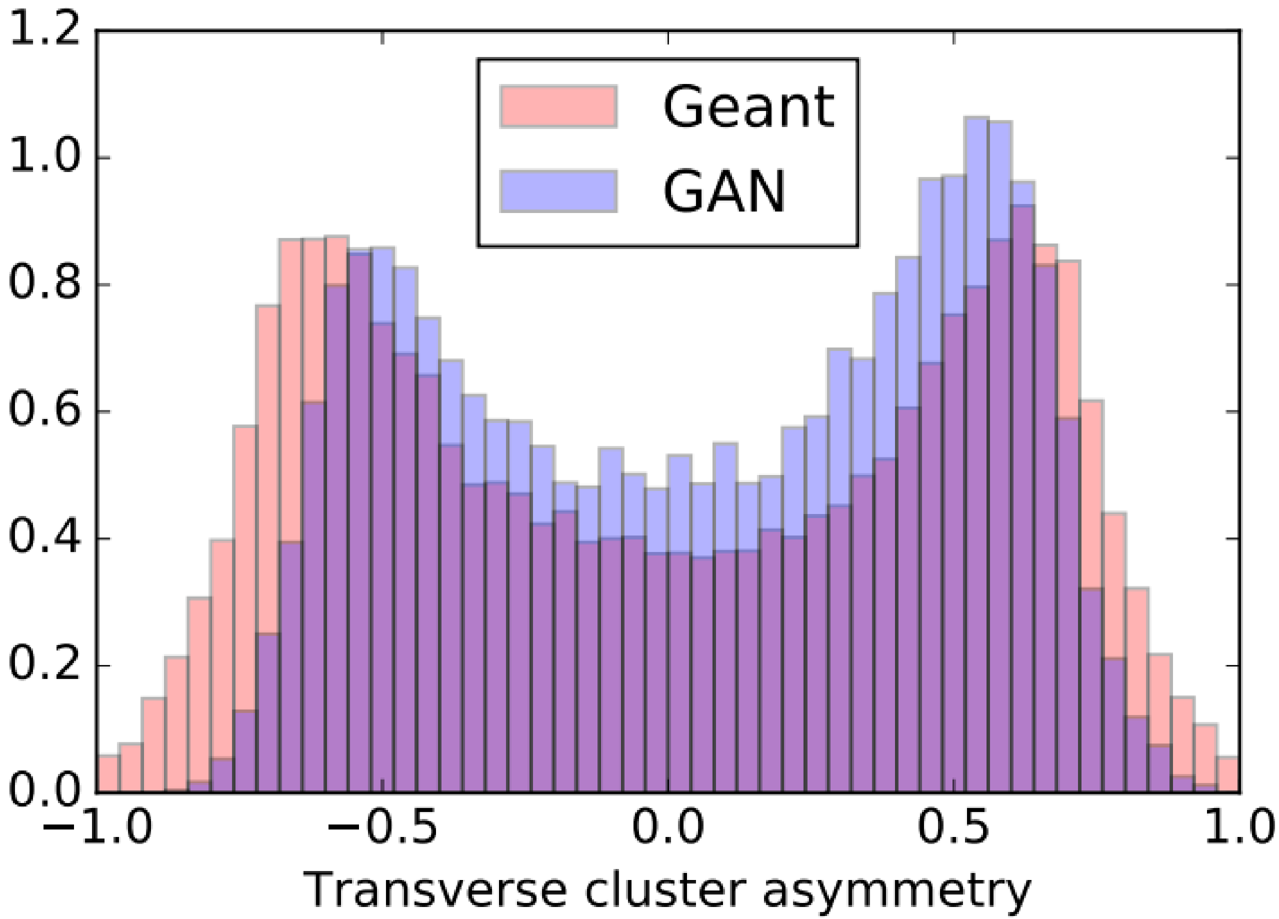


Порівняння

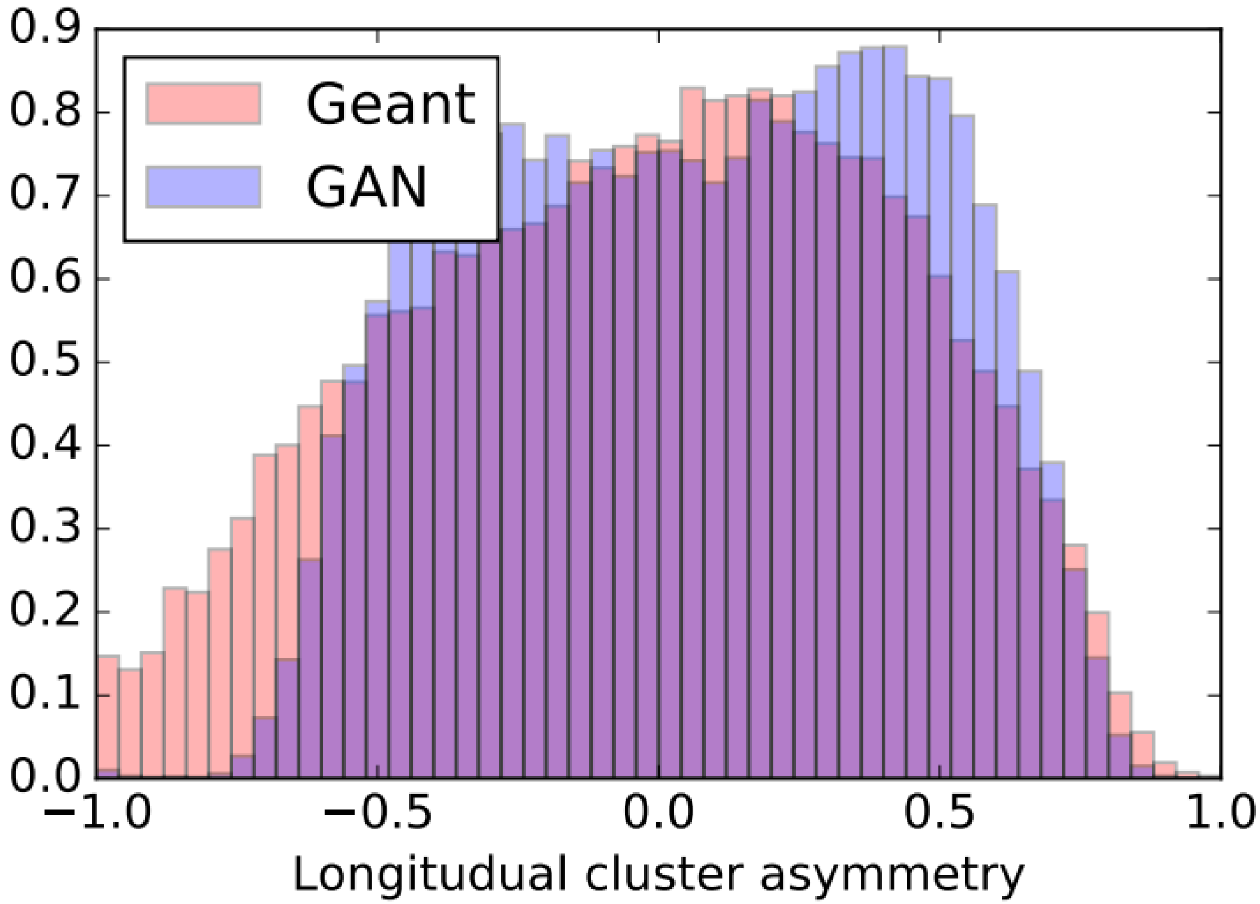


Fraction of cells above threshold
log10(Threshold/MeV)
Geant
GAN
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Порівняння


Geant
GAN
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
Transverse cluster asymmetry

Порівняння



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−1.0
−0.5
0.0
0.5
1.0
Geant
GAN
Longitudual cluster asymmetry