Particle-GAN for Full Event Simulation at the LHC

Thong Nguyen[1]

in collaboration with

Jesus Arjona Martinez[2], Maurizio Pierini[3],
Maria Spiropulu[1], and Jean-Roch Vlimant[1]

[1]Caltech [2]Cambridge [3]CERN

19th International Workshop on Advanced Computing and Analysis Techniques in Physics Research
12th March 2019. Saas-Fee, Switzerland

Caltech

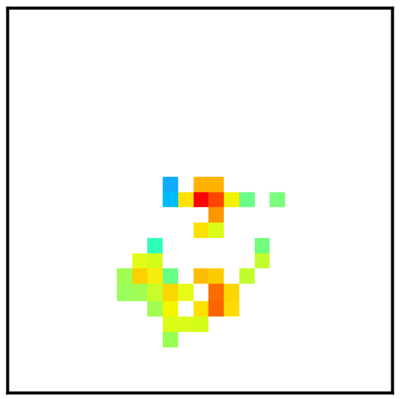
Generative Adversarial Networks (GANs)

- Zero-sum non-cooperative game between a forger (generator) and a detective (discrimator).
- Solution: Nash's equilibrium to the objective function:

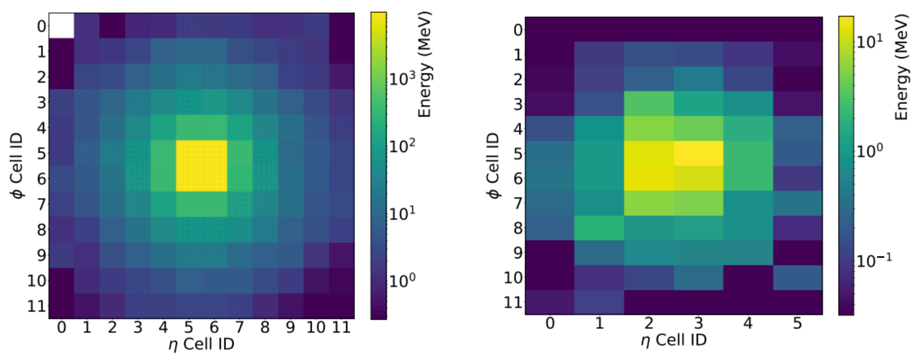
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

GANs at the LHC

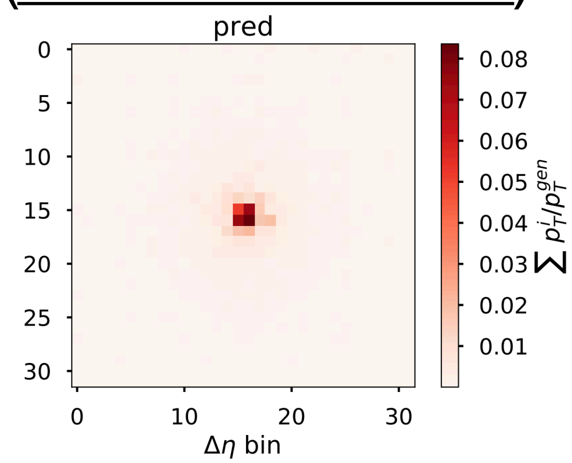
LAGAN
(arXiv:1701.05927)



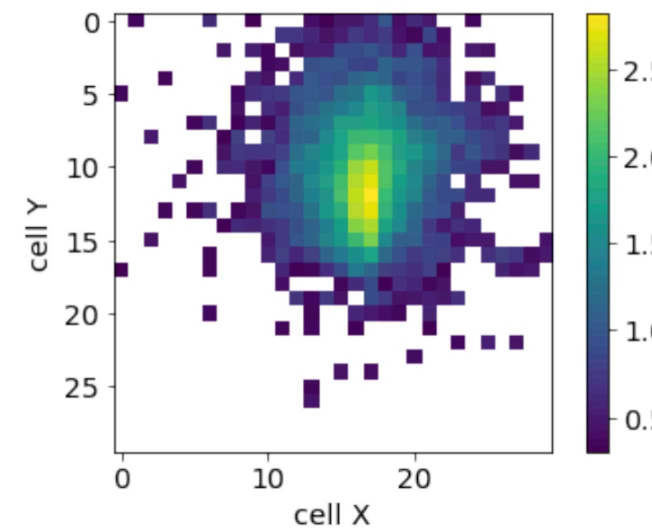
CaloGAN (arXiv:1712.10321)



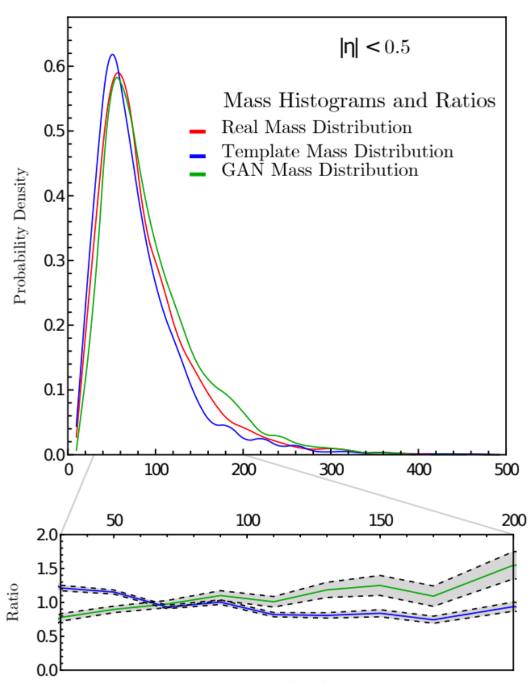
GAN for hadronic jets
(arXiv:1805.00850)



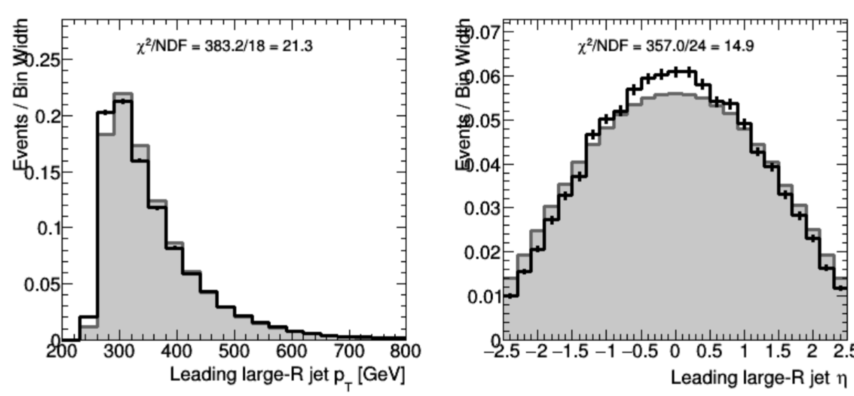
GAN for fast calo sim
(arXiv:1812.01319)



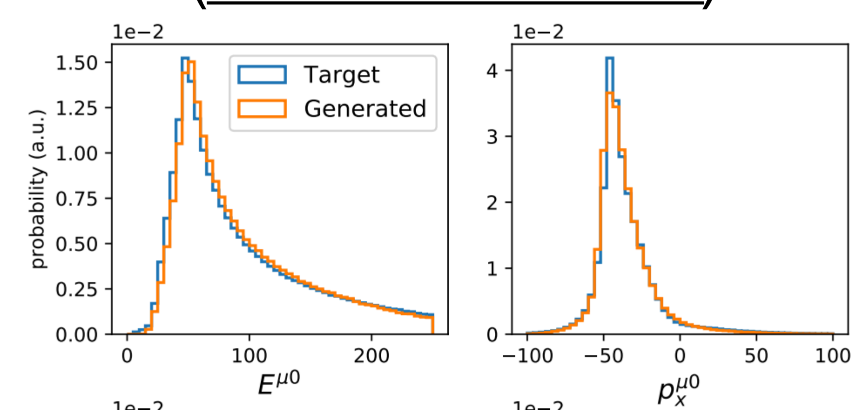
GAN for QCD factorization
(arXiv:1903.02556)



DijetGAN (arXiv:1903.02433)



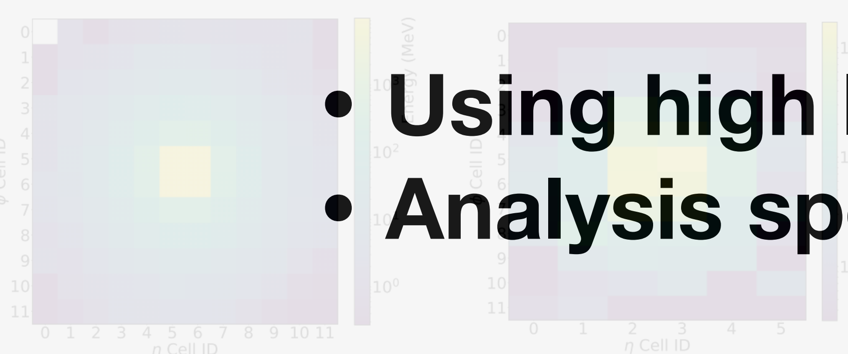
Analysis-specific GAN
(arXiv:1901.05282)



GANs at the LHC

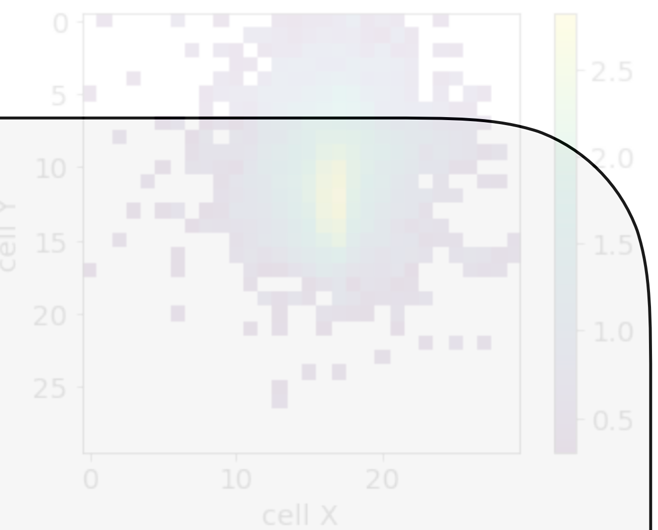
LAGAN (arXiv:1701.05927)

CaloGAN (arXiv:1712.10321)



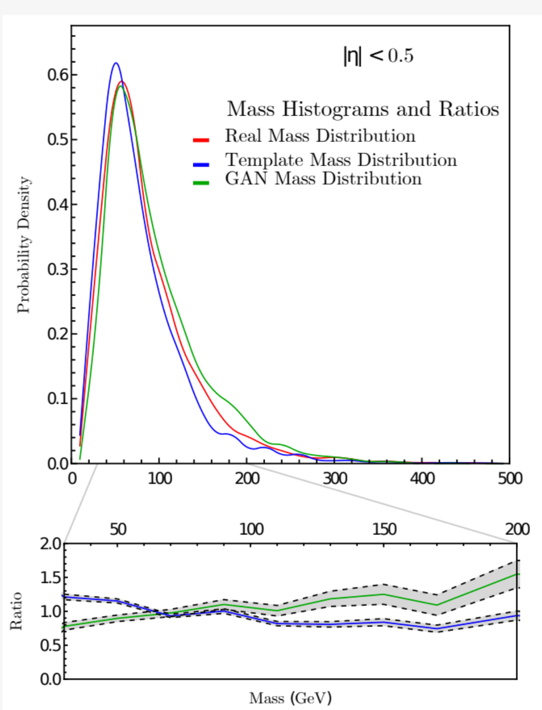
GAN for hadronic jets (arXiv:1805.00850)

GAN for fast calo sim (arXiv:1812.01319)

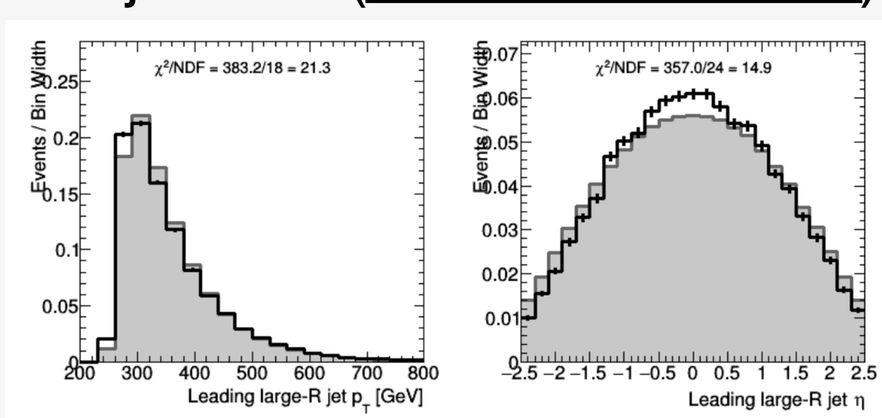


- **Using high level features.**
- **Analysis specific.**

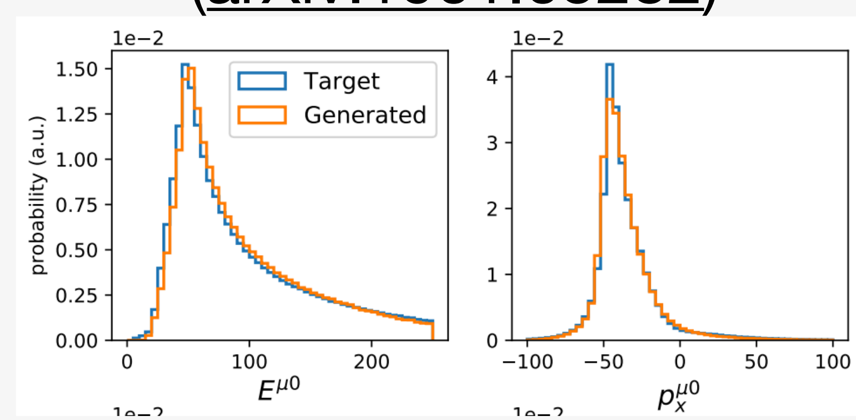
GAN for QCD factorization (arXiv:1903.02556)



DijetGAN (arXiv:1903.02433)

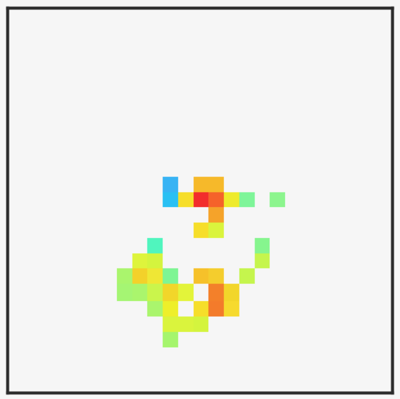


Analysis-specific GAN (arXiv:1901.05282)

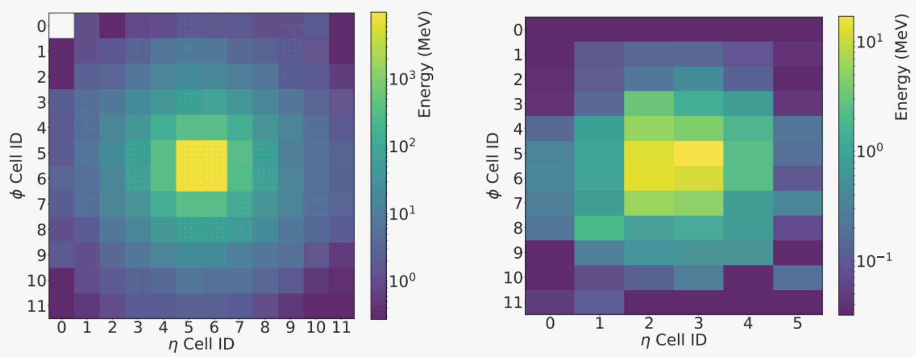


GANs at the LHC

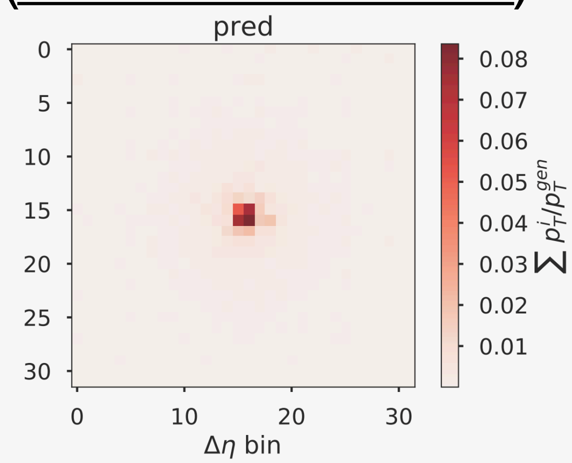
LAGAN (arXiv:1701.05927)



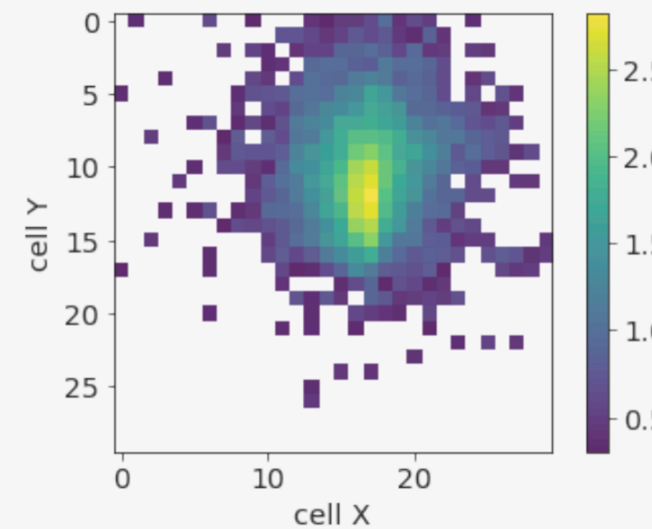
CaloGAN (arXiv:1712.10321)



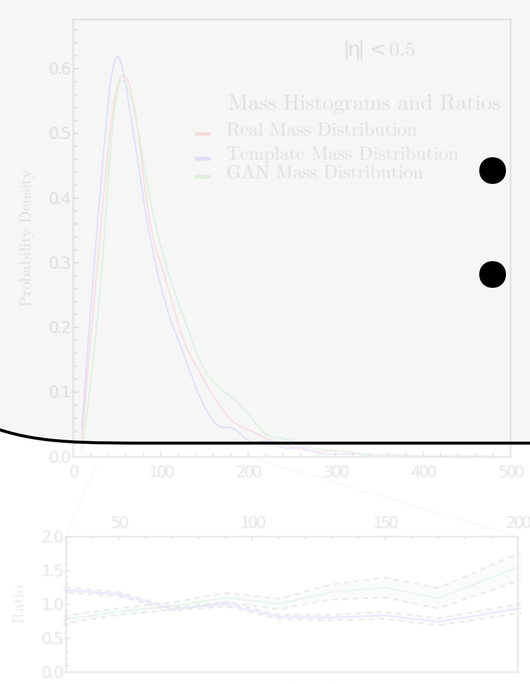
GAN for hadronic jets (arXiv:1805.00850)



GAN for fast calo sim (arXiv:1812.01319)

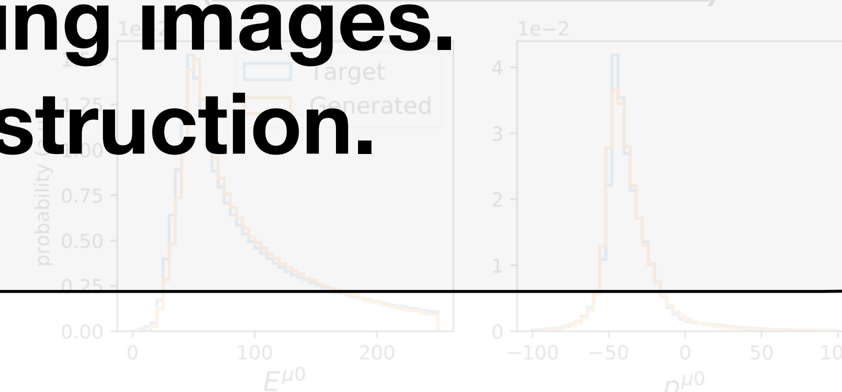


GAN for QCD factorization (arXiv:1903.02556)



DijetGAN (arXiv:1903.02433)

Analysis-specific GAN (arXiv:1901.05282)

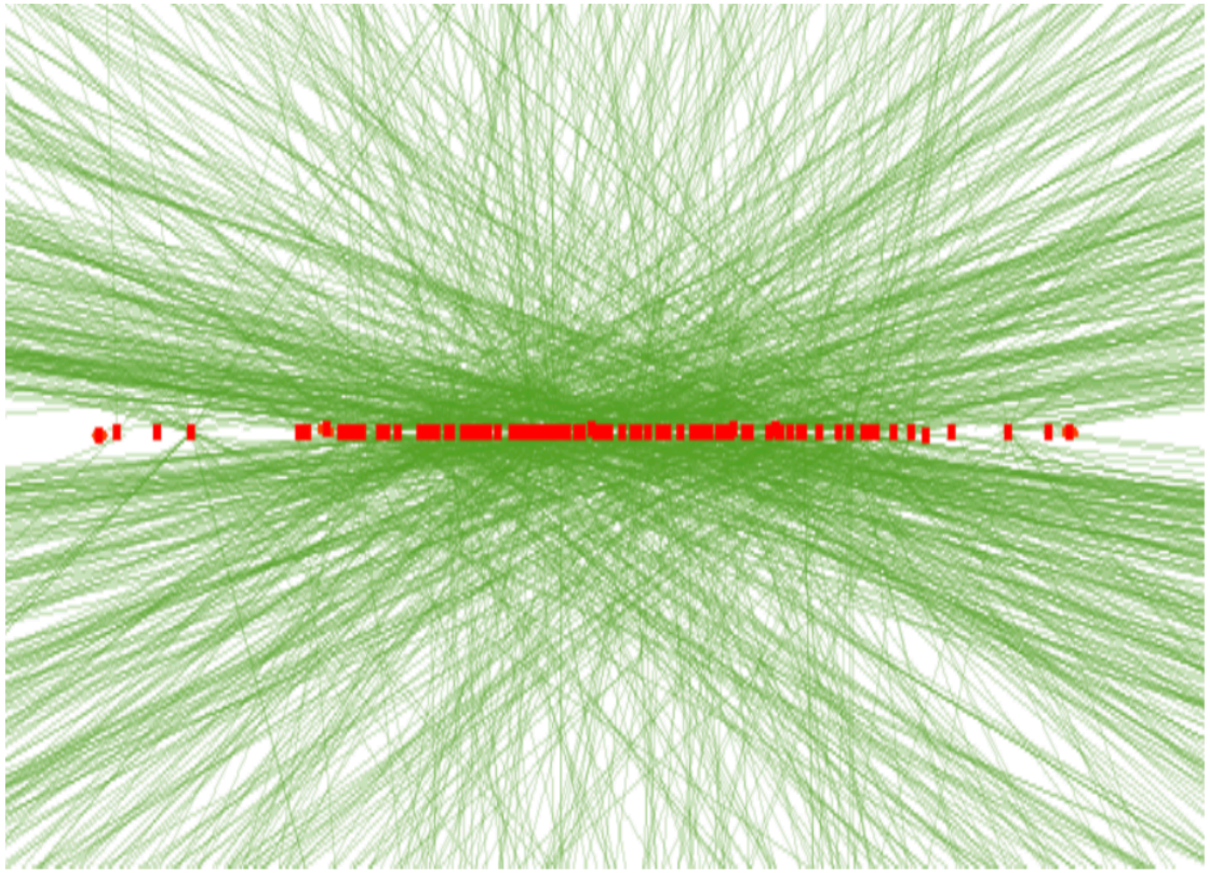


- **Particle-based, generating images.**
- **Can't be used for reconstruction.**

Particle-GAN

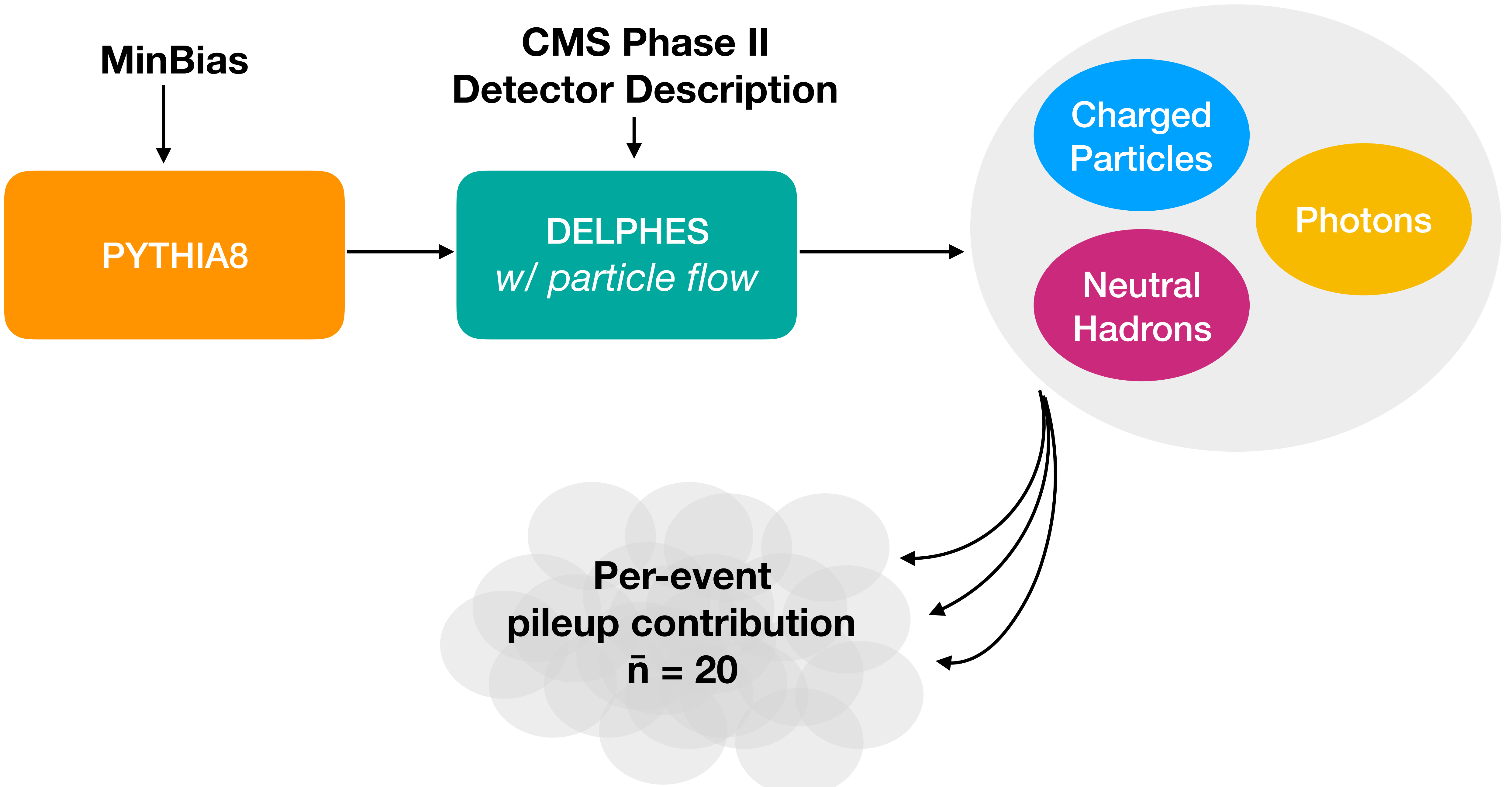
- Generating list of particles mimicking particle-flow candidates to be used directly by reconstruction algorithms.
  - Speed up fullsim.
  - Improve fastsim.
- Analysis-independent.
- Starting with MinBias event simulation, each PF candidate represented by (p_T, η, ϕ).
  - Possible use case: generate per-event pileup.

Pile-up at the LHC



- Pileup: Low energy interactions simultaneous with interesting hard interaction in one event.
- Realistic simulation:
  - Classical PU: **expensive** computation.
  - Premix PU library: **heavy cost** on network/IO, **limitation** on flexibility of simulation conditions.
  - GAN: **~zero cost**.

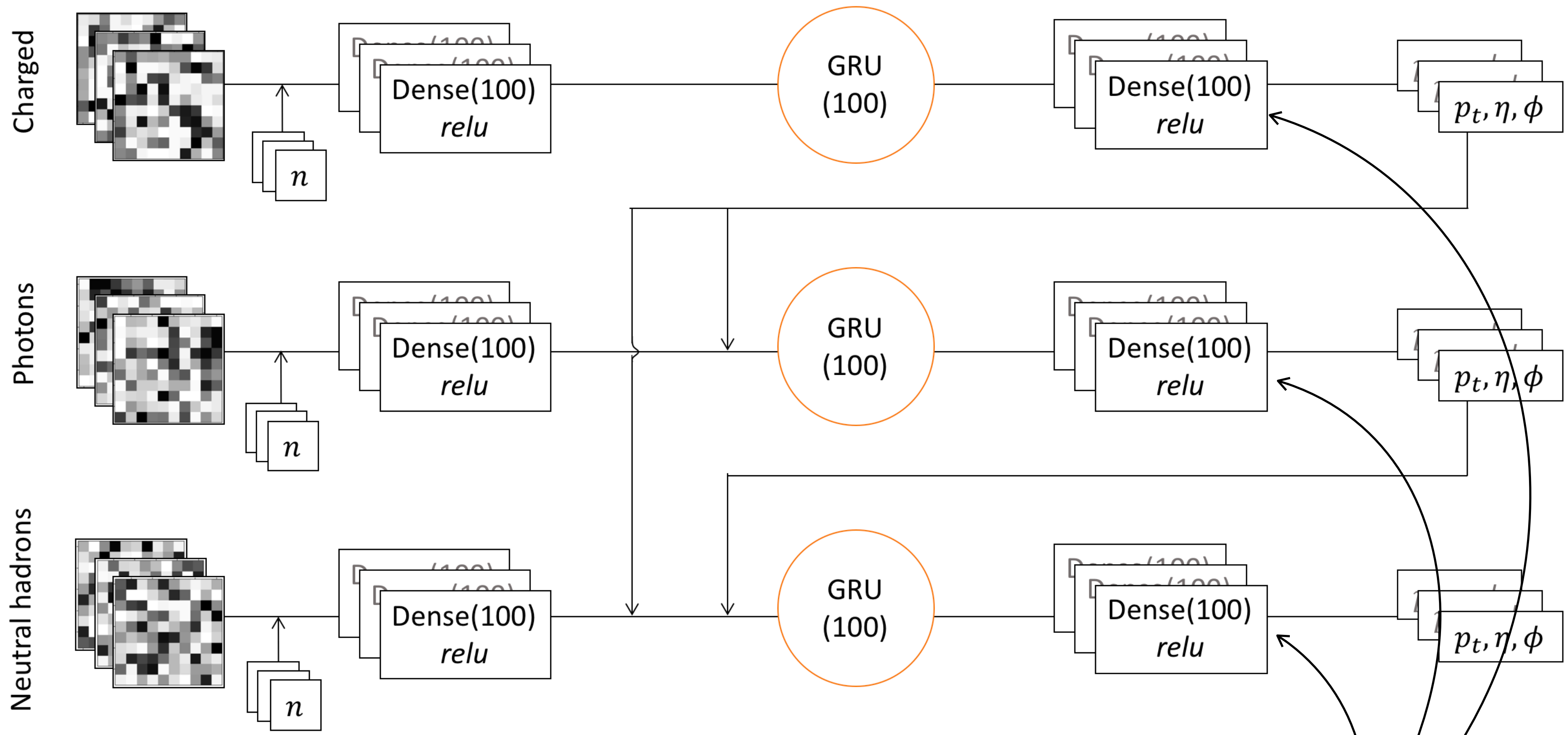
Dataset



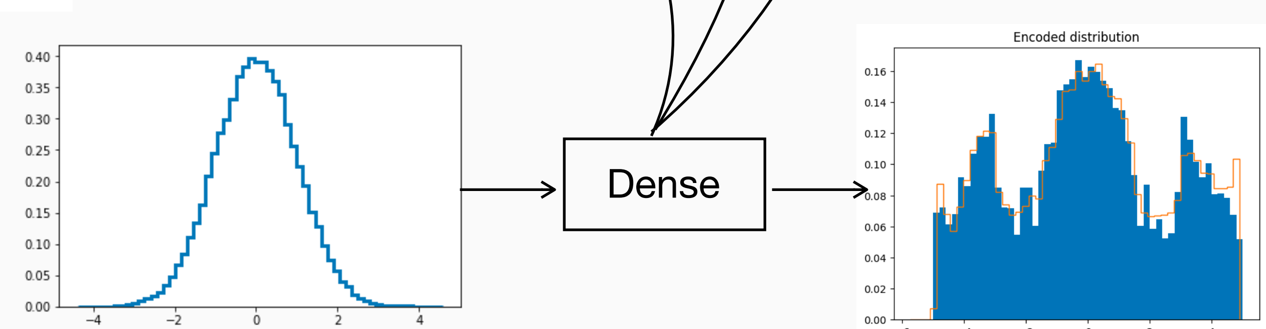
MinBias
CMS Phase II
Detector Description
PYTHIA8
DELPHES
w/ particle flow
Charged Particles
Photons
Neutral Hadrons
Per-event
pileup contribution
n̄ = 20

Baseline P-GAN

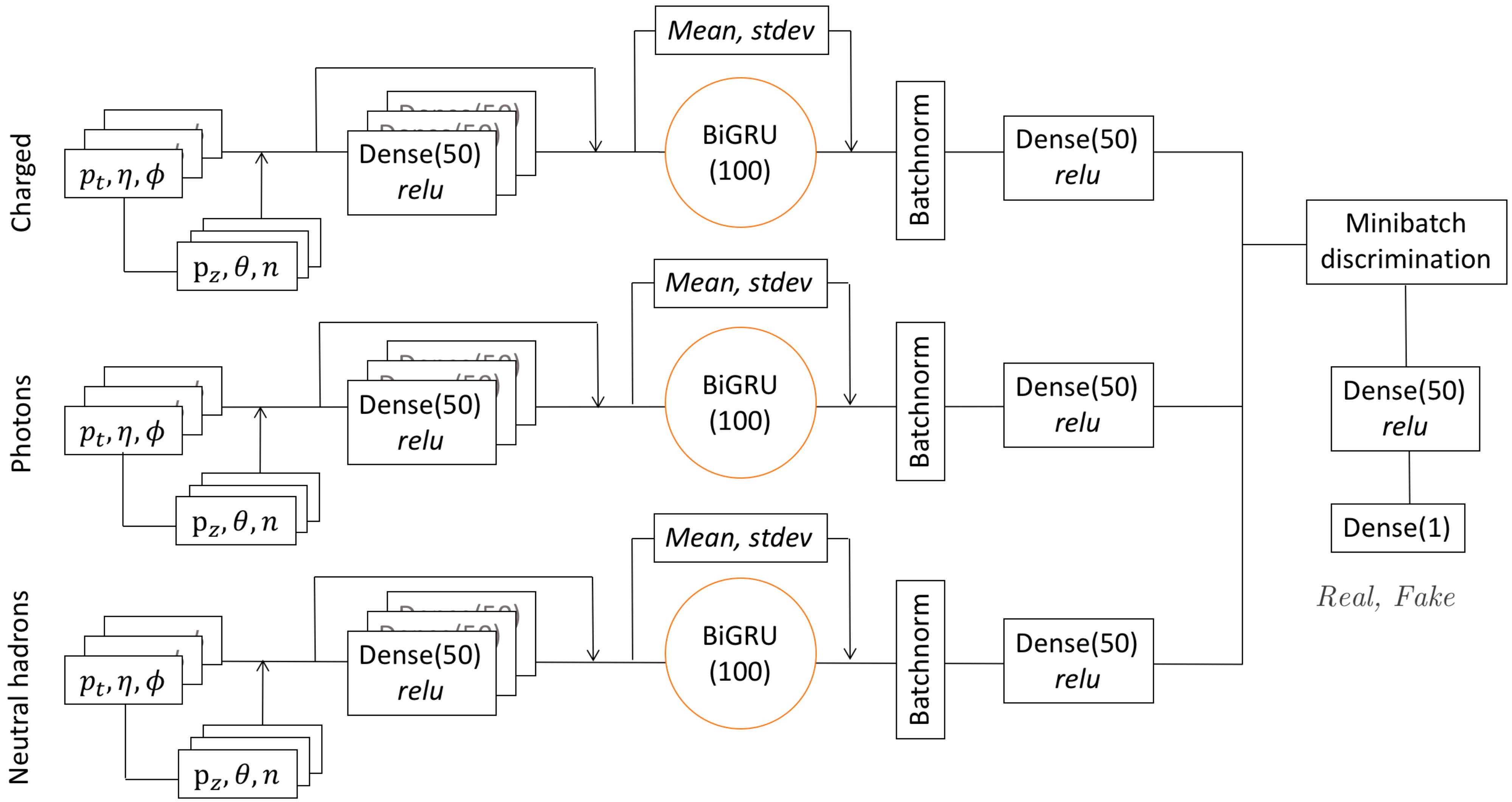
Generator Architecture



η generation layers are pretrained in separated networks then inserted into the GAN generator.



Discriminator Architecture



- A physics layer maximizes the information given to the discriminator.
- Minibatch discrimination and feature matching to avoid mode collapse.

Evaluation Metrics

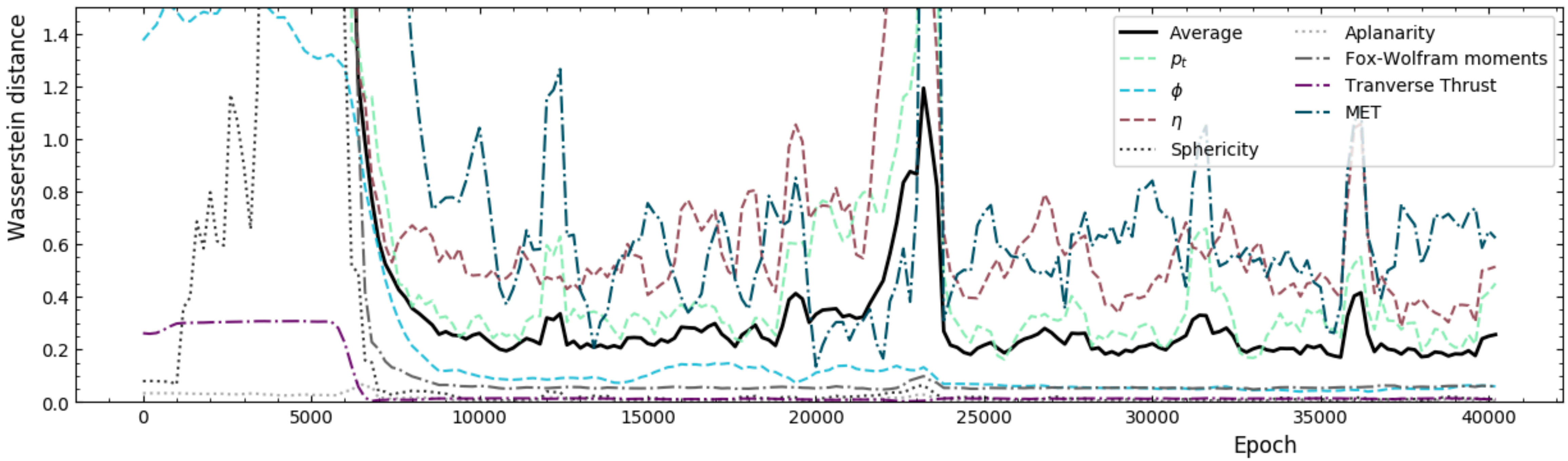
- Least-square GAN:

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\big[(D(\boldsymbol{x})-1)^2\big] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\big[(D(G(\boldsymbol{z})))^2\big]$$
$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\big[(D(G(\boldsymbol{z}))-1)^2\big],$$

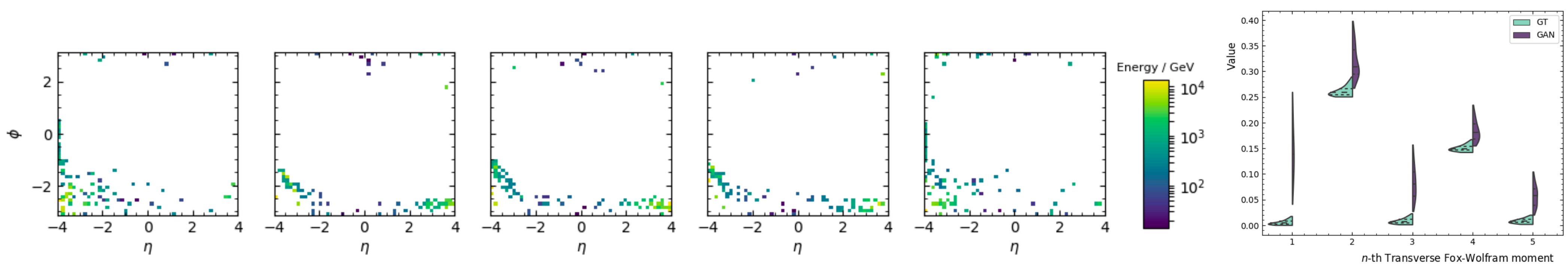
- Performance metrics:
  - Fox-Wolfram moments
  - Sphericity
  - Aplanarity
  - Global transverse thrust
  - Missing transverse energy
- Wassserstein distances between distributions of real and fake events over above metrics are used to evaluate the model.

Training Process

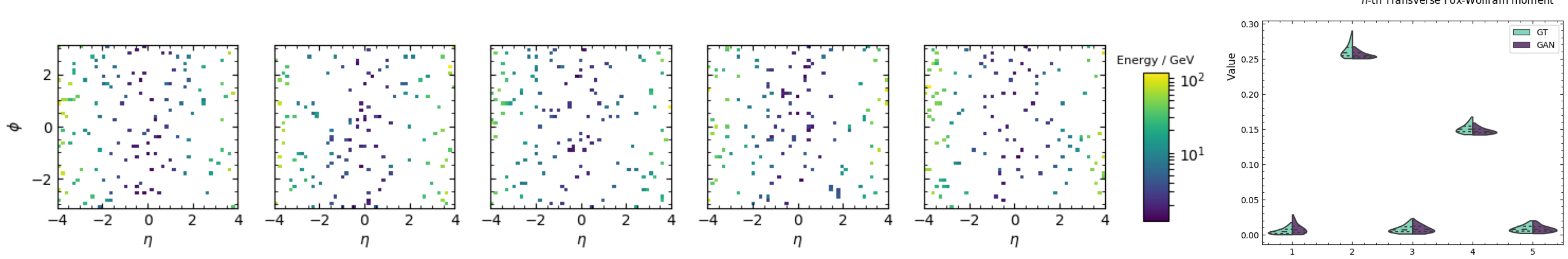
Evaluation metrics vs epochs. Model with lowest average value (black line) is chosen.



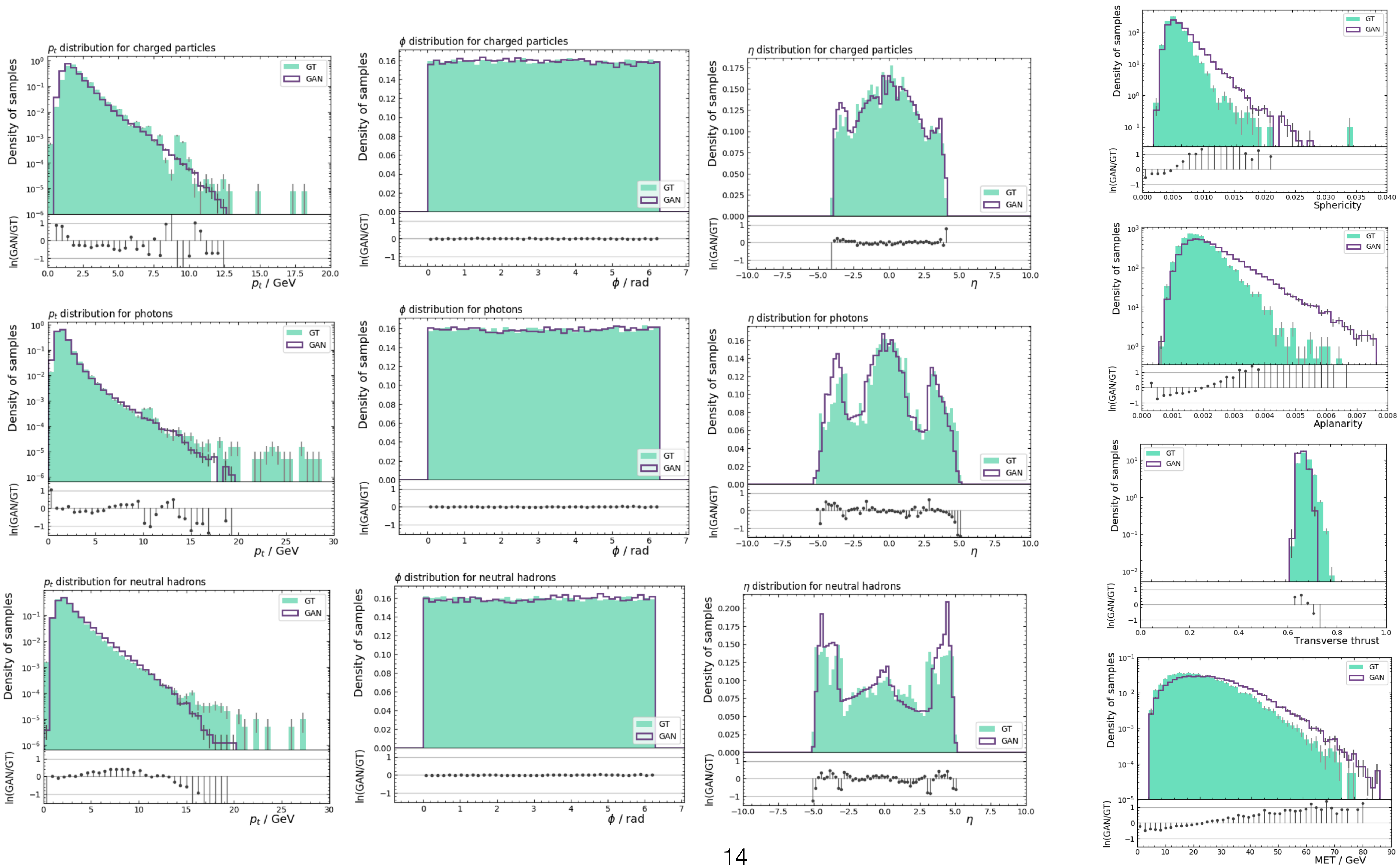
Early epoch



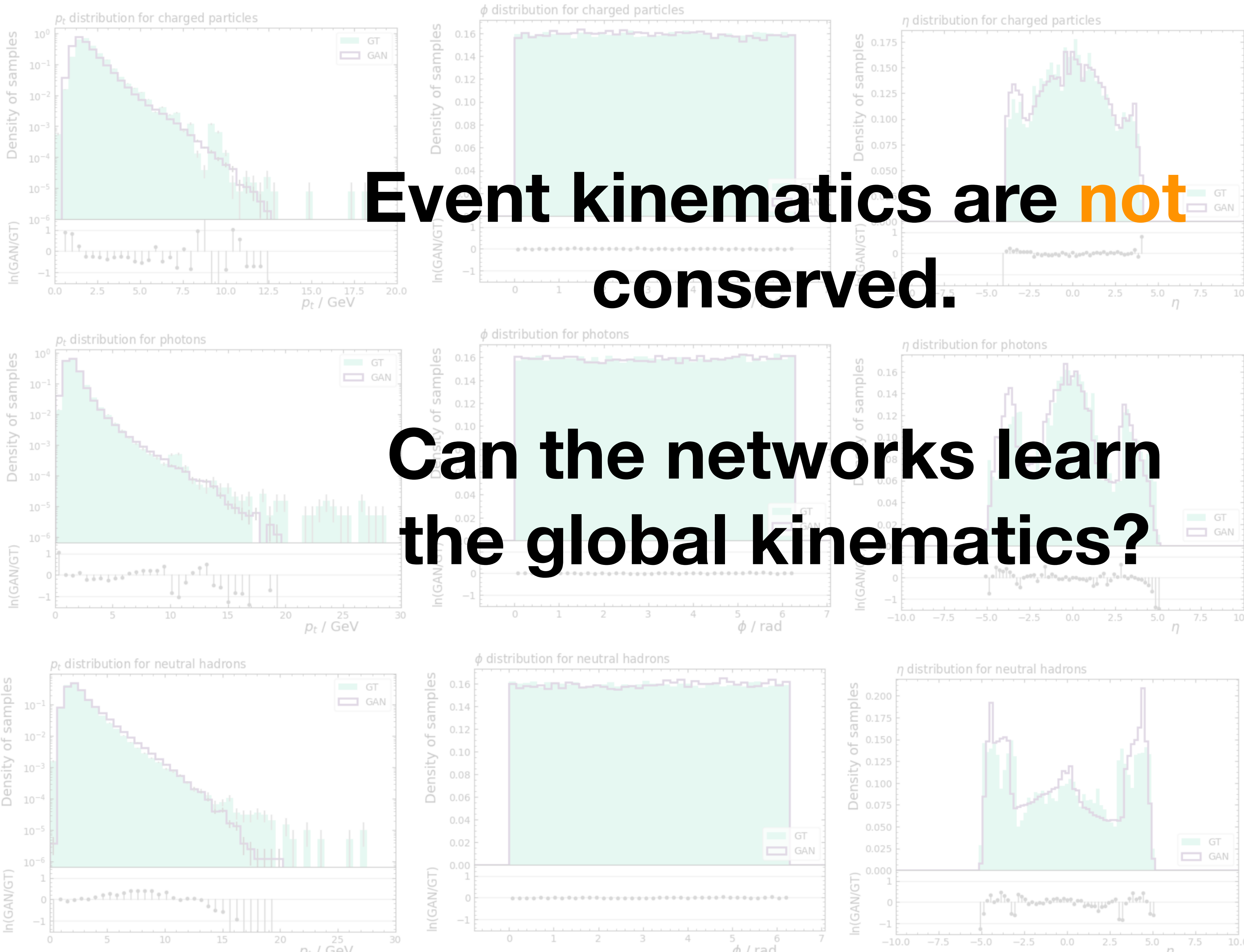
Optimal model



Performances

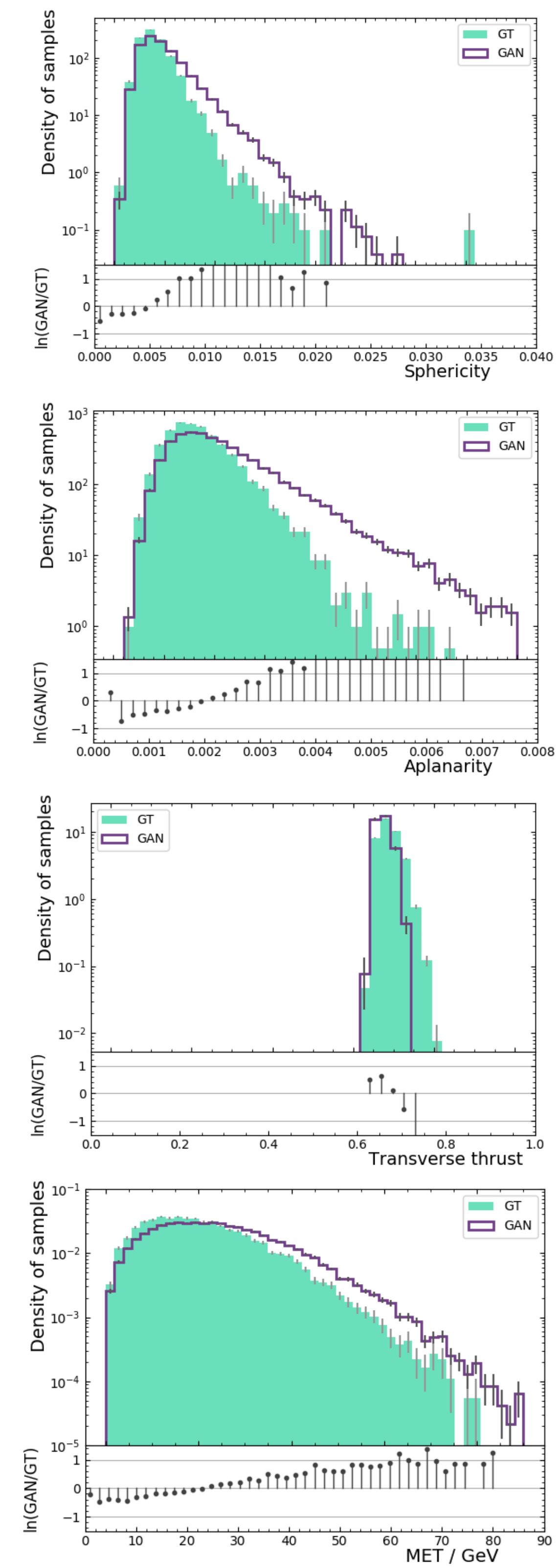


Performances



Event kinematics are not conserved.

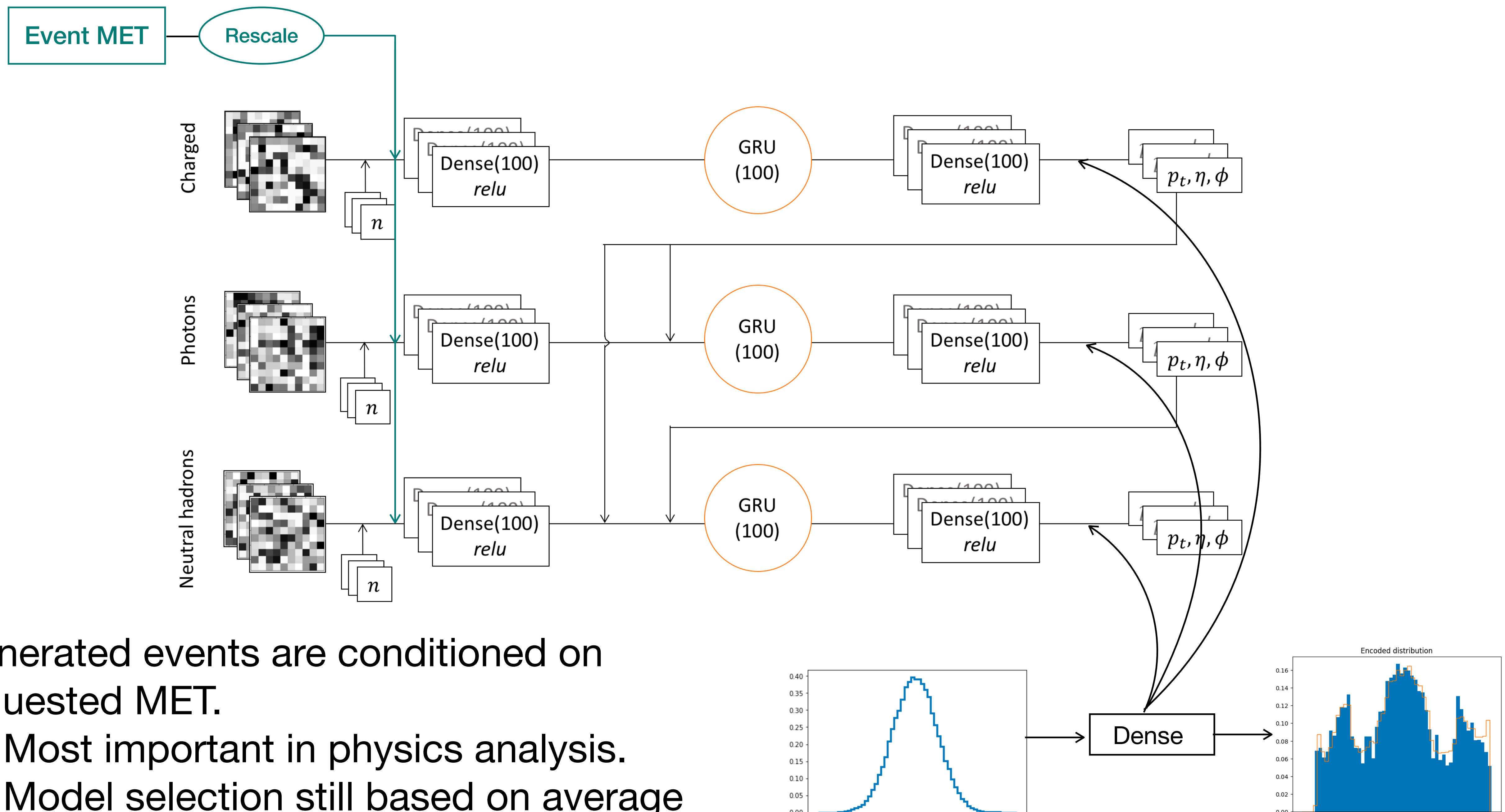
Can the networks learn the global kinematics?



Conditional P-GAN

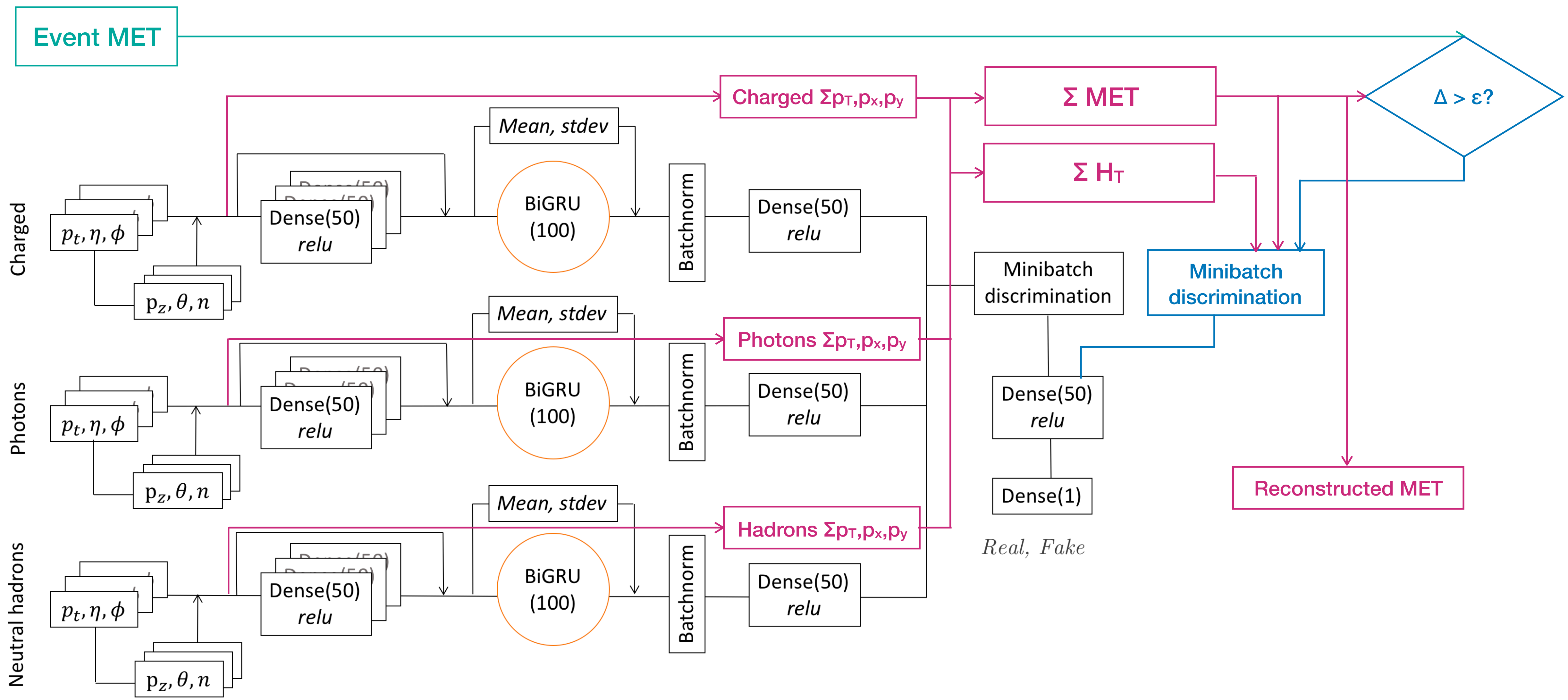
Learning to conserve event kinematics

Generator Architecture



- Generated events are conditioned on requested MET.
  - Most important in physics analysis.
  - Model selection still based on average metrics rather than MET alone.
- Training data are transformed s.t. $\phi_{MET} = 0$.

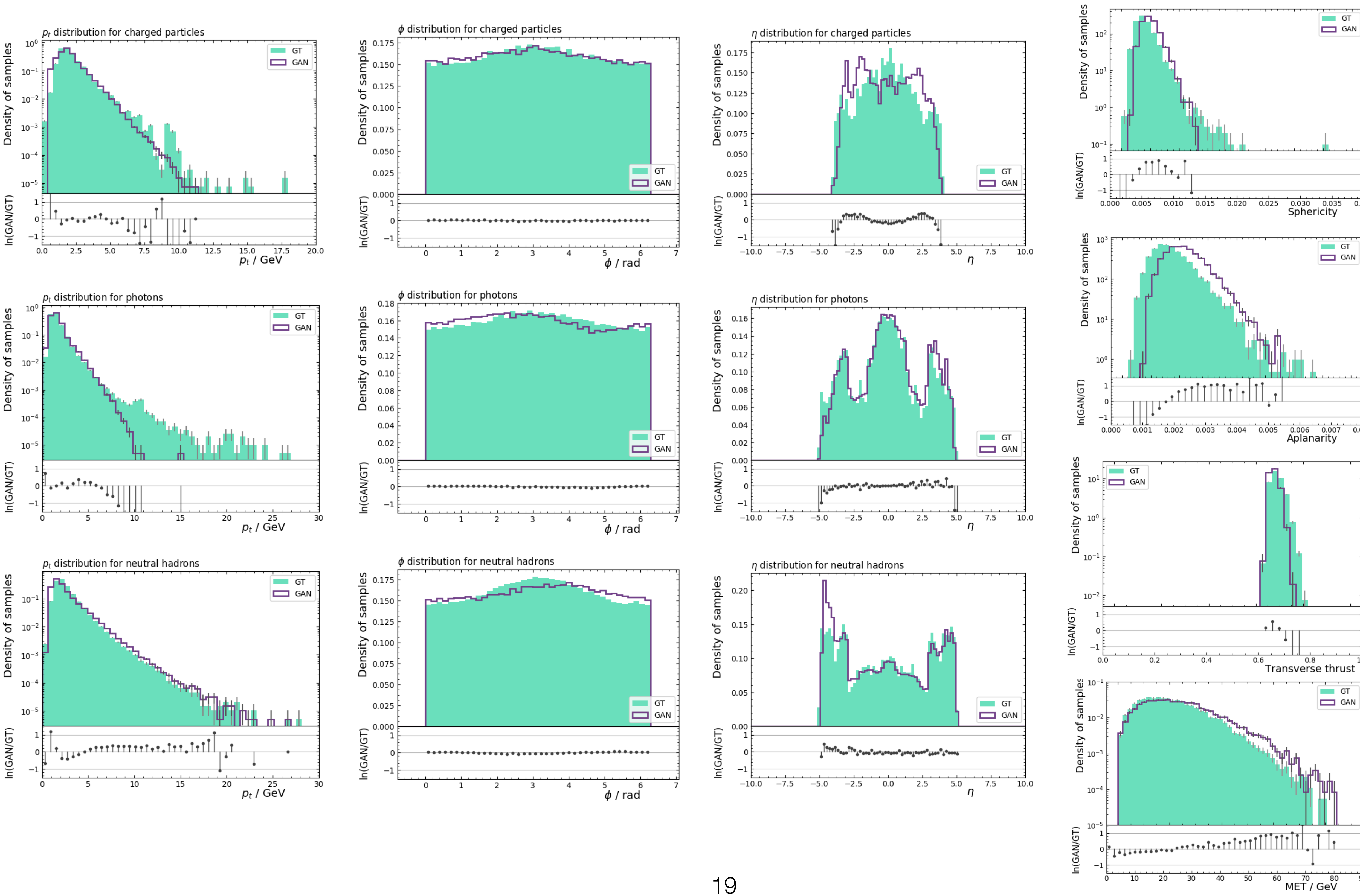
Discriminator Architecture



Additional regression term to the loss functions

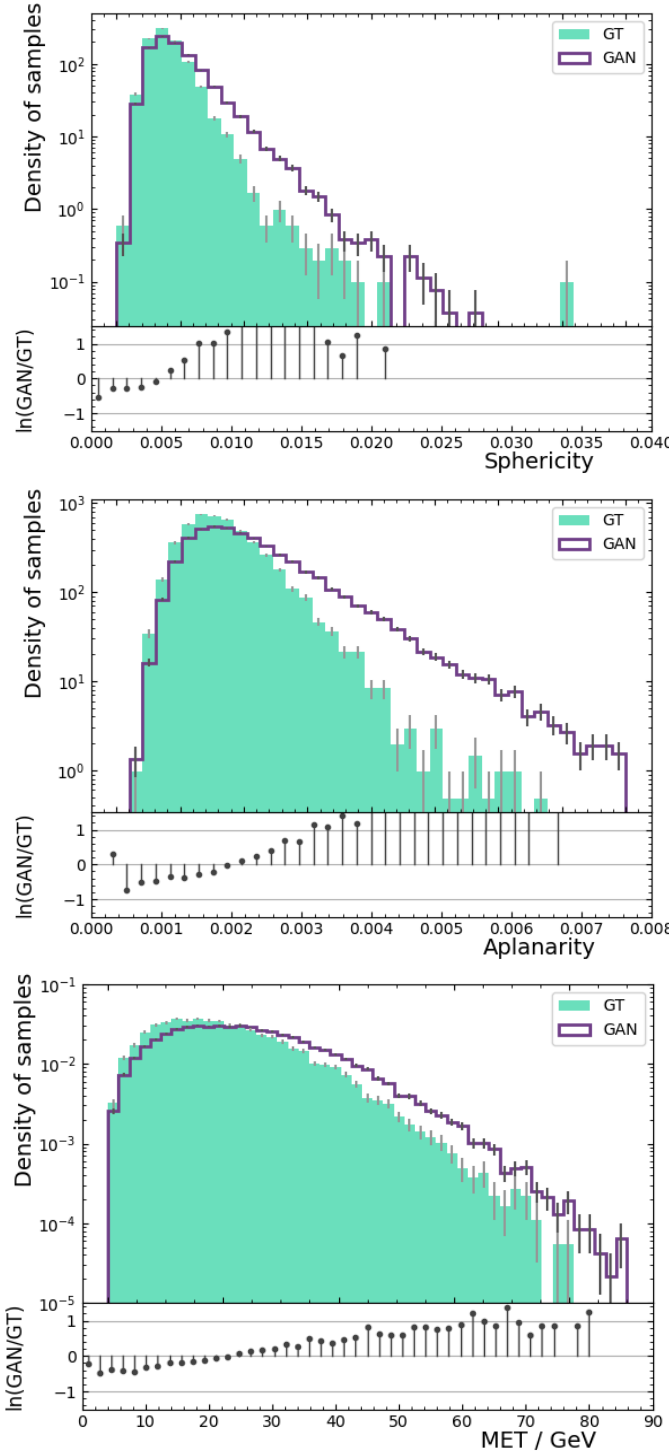
$$\mathscr{L}_{MET} = E_{z\sim p(z),\, E_T\sim data}[\Delta(E_T, \hat{E}_T(G(z|E_T)))] + E_{x\sim data}[\Delta(E_T(x), \hat{E}_T(x))]$$

Performances

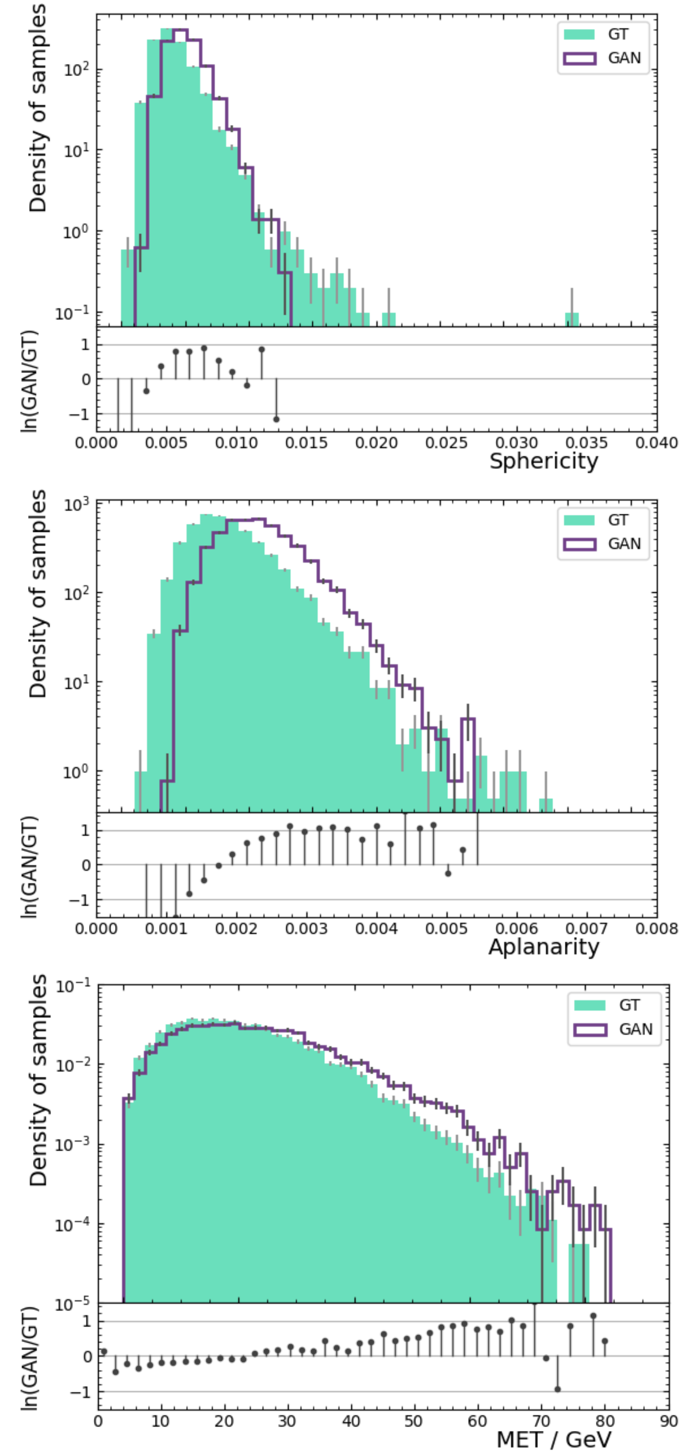


Performance Comparison

Baseline P-GAN



Conditional P-GAN



- **Improvements on event kinematic conservation.**

Summary

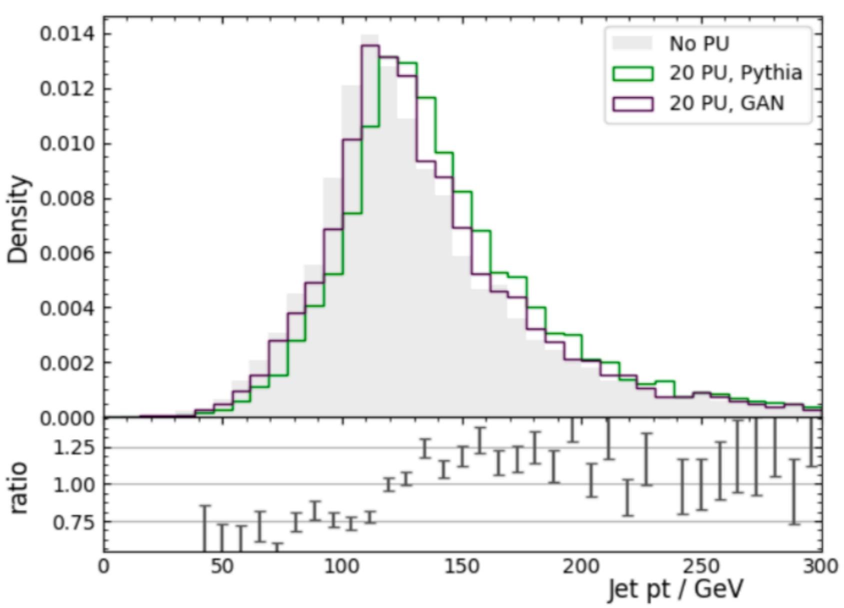
- We investigate a potential use case of generative adversarial networks for particle-based full event simulation at the LHC.
- **Work in progress**: There are still challenges for achieving precision in global kinematic reconstruction and some particle-feature distributions.
  - By exploiting smarter network architectures, we **reduce** significant discrepancy in certain kinematic phase spaces.
  - More work needs to be done to achieve a full-event fastsim solution with deep neural networks.

BACKUP

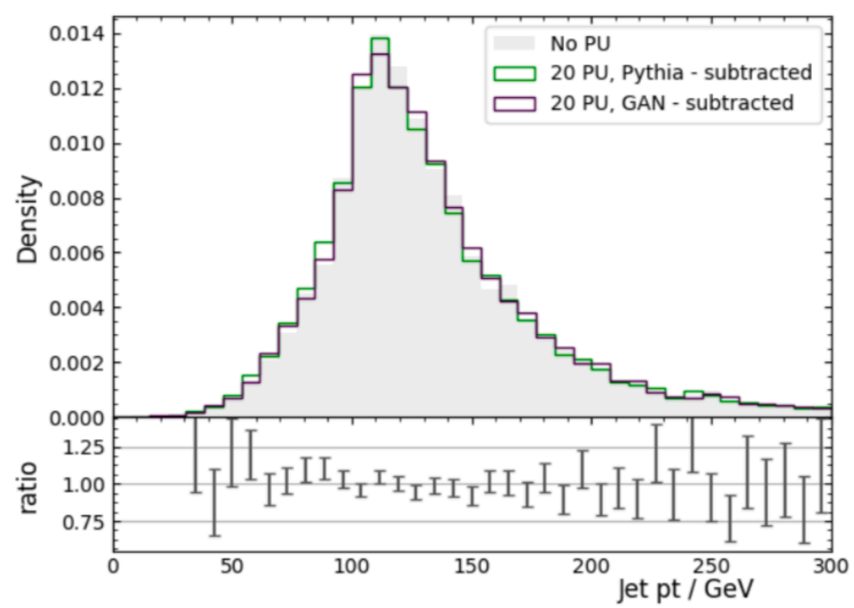
Pileup Subtraction

- Pileup removal: subtract soft radiation from QCD.
  - More relevant to physics analysis at the LHC.
- Apply SoftKiller algorithm to $Z \to \nu\nu$ sample with and without pileup emulation at $\bar{n}_{PU} = 20$.

Before SoftKiller



After SoftKiller



Mean leading jet p_T

	$\langle p_T \rangle$ / GeV
No PU	**136.8**
Pileup - GT	146.6
Pileup - RGAN	141.1
Pileup - GT - subtracted	**135.0**
Pileup - RGAN - subtracted	**135.7**

- After pileup subtraction, average leading jet p_T generated with Pythia and RGAN agree within 0.7 GeV.

Computational Benchmark

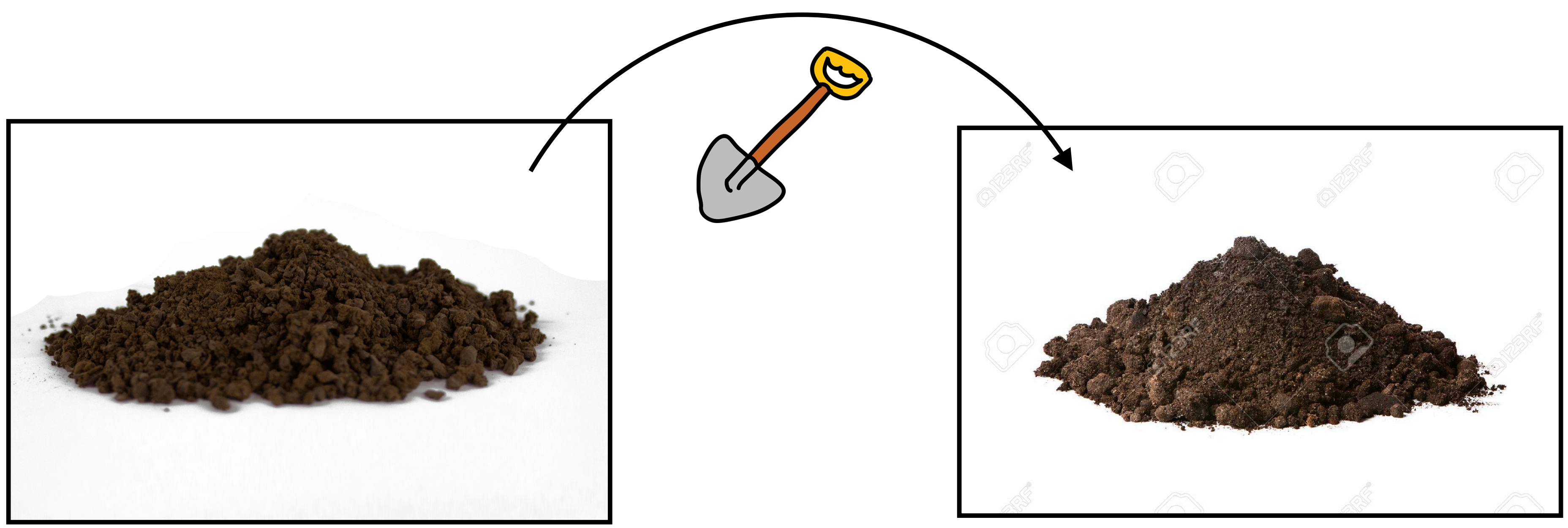
- RGAN inference time: 3.19 ms per event (single thread Intel® Xeon® CPU E5-2650 v4 @ 2.2 GHz.
- Average simulation time: O(100s) per event[1].
- Improvement by a factor of 10^5.

[1] https://arxiv.org/pdf/1903.02433.pdf

GAN Evaluation

- Loss functions can't be used to benchmark GAN's performance.

- Wasserstein/Earth mover's distance: the minimum work (stuff x distance) to rearrange one pile of dirt into another.



- Use Wasserstein distances of different metrics between real and generated event distributions to evaluate the best model.