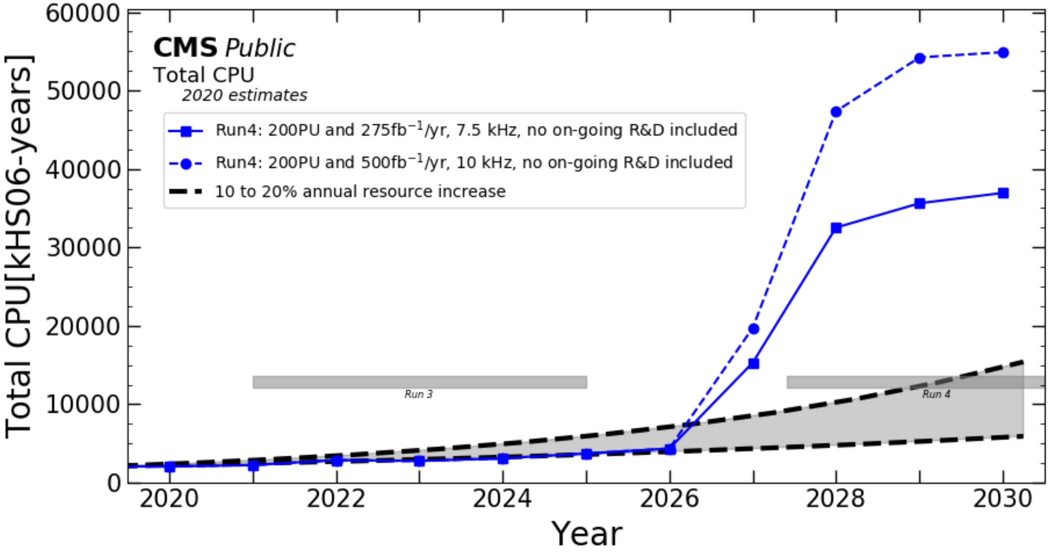
# Machine Learning for Detector Simulation

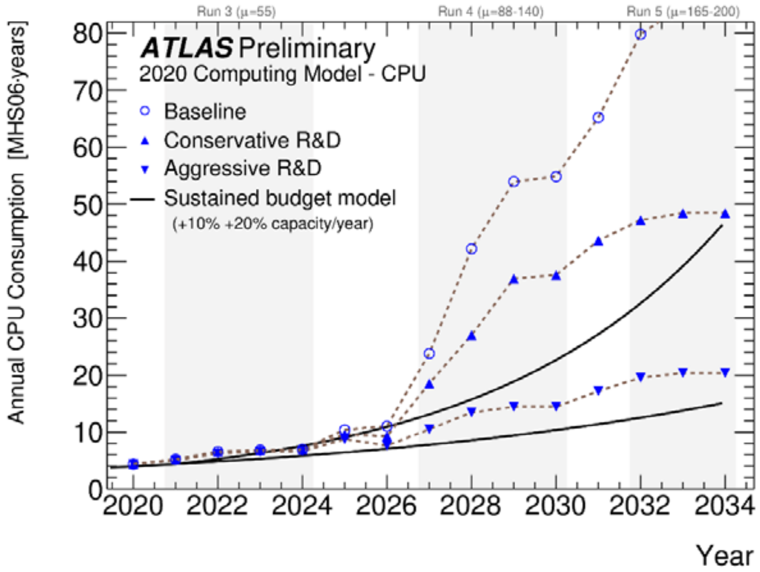
Kevin Pedro (FNAL)
on behalf of ATLAS, CMS, LHCb

November 23, 2020

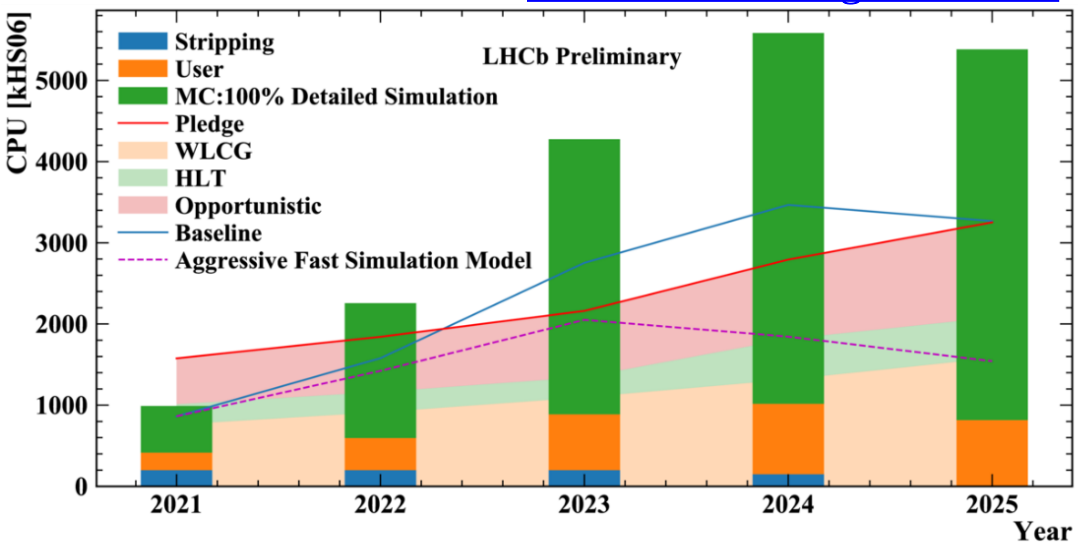
# Computing Challenges

[CMSOfflineComputingResults](#) [AtlasComputingandSoftwarePublicResults](#)




[LHCb CPU Usage Forecast](#)

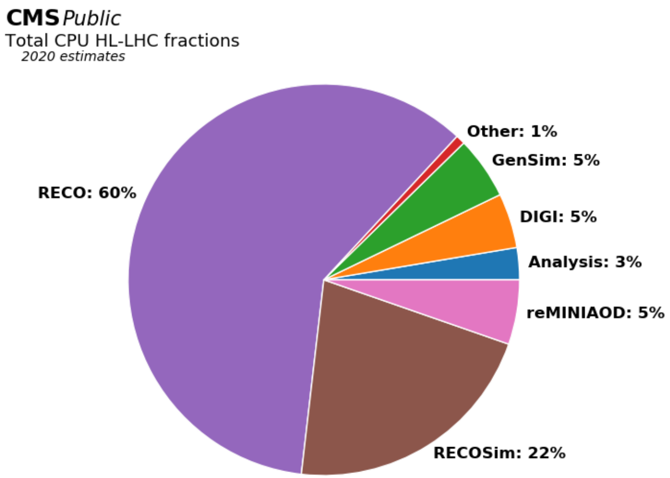


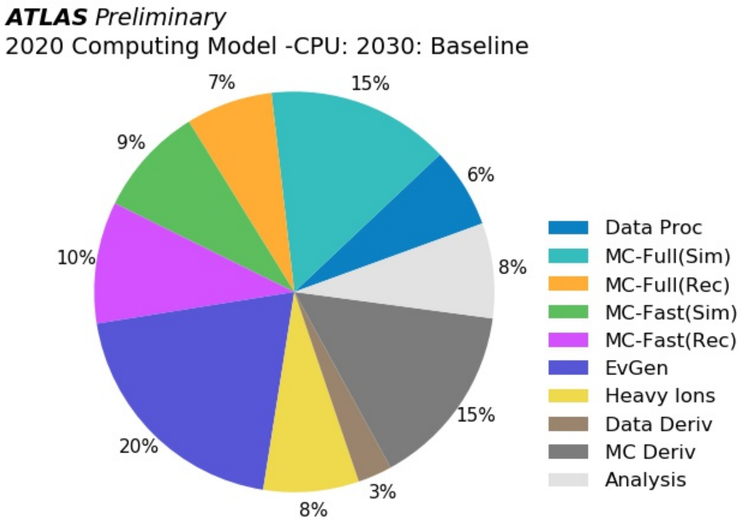
# Motivation

- Beginning of Run 2: full detector simulation (Geant4) took ~40% (plurality) of grid CPU resources for CMS & ATLAS [arXiv:1803.04165]
  - Detector upgrades for HL-LHC: increased complexity [arXiv:2004.02327]
  - Further technical improvements expected to be limited [arXiv:2005.00949]
- Reconstruction CPU usage scales superlinearly with pileup

➢ Simulation needs to deliver more events w/ more complexity
…while using smaller fraction of CPU

  - LHCb detailed simulation exceeds available CPU even for Run 3



# Classical Simulation Engines

- "FullSim": Geant4
  - Common software framework
    - Experiments can provide additional code via user actions
  - Explicit modeling of detector geometry, materials, interactions w/ particles
  - Physics lists include many models of particle interactions (for different energy ranges, etc.)
- "FastSim":
  - Usually experiment-specific framework
  - Implement approximations: analytical shower shapes (e.g. GFLASH), truth-assisted track reconstruction, etc.
- Delphes:
  - Ultra-fast parametric simulation
  - Used for phenomenological studies, future projections, etc.

# Generative Machine Learning

- Machine learning algorithms (e.g. deep neural networks):
  - Typically trained for classification or regression tasks
  - Can also do generation tasks: creating novel output from some input
- Industry has demonstrated impressive, but not foolproof, results, e.g.:
  - Images (StyleGAN2)
  - Text (GPT-3)



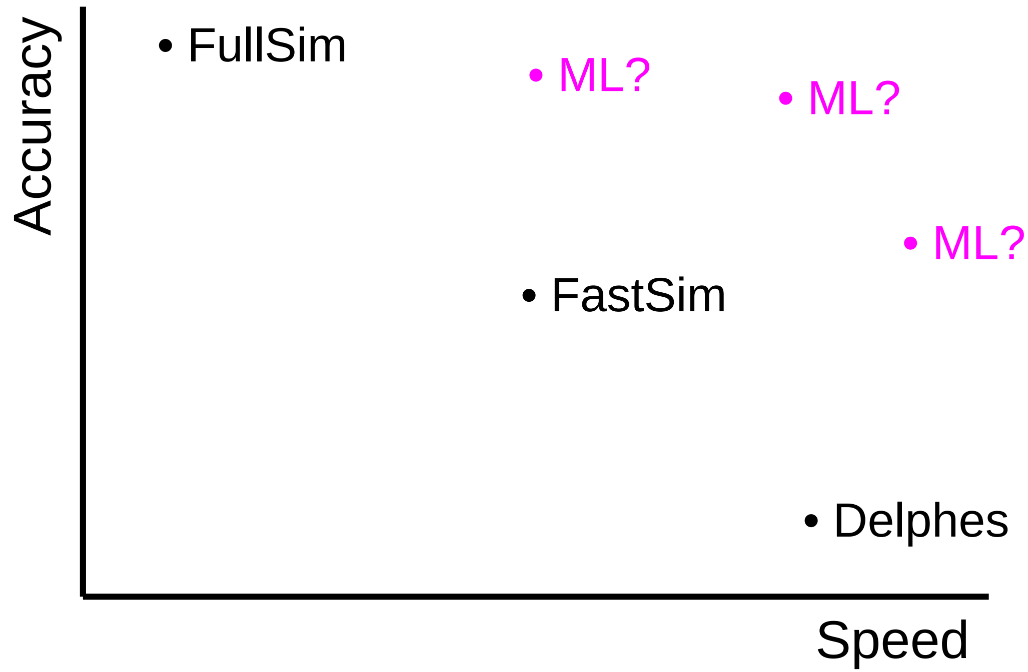
from thispersondoesnotexist.com

# Machine Learning for Simulation

- Pros:
  - Achieve higher accuracy than "simple" fast simulations
  - Produce faster results than Geant4
    - ML inference can be accelerated on coprocessors (GPUs, FPGAs, etc.)
      - avenue to utilize HPCs
  - Generate various quantities
    - Particle showers, 4-vectors, particle ID, high-level features, etc.
- Cons:
  - May need large training datasets and training time
    - StyleGAN2: 25M images, 5-10 days to train on 8 V100 GPUs
    - Cost-benefit analysis should include CPU and GPU usage for training
  - Statistical validity needs careful consideration
    - Extrapolation outside of training dataset may be unreliable

➢ Any claimed speedup is ***only*** meaningful if results are physically accurate

# Speed vs. Accuracy

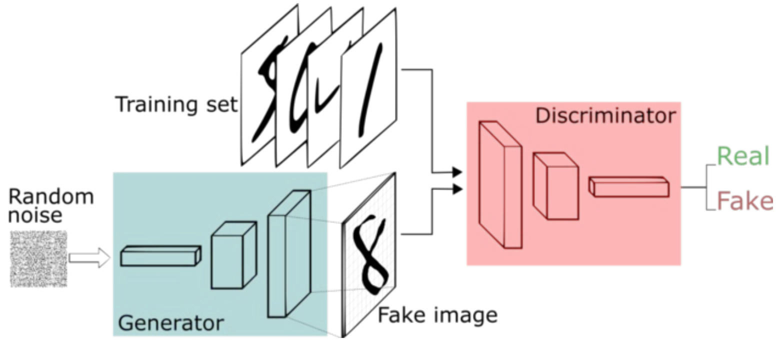



Several different approaches:

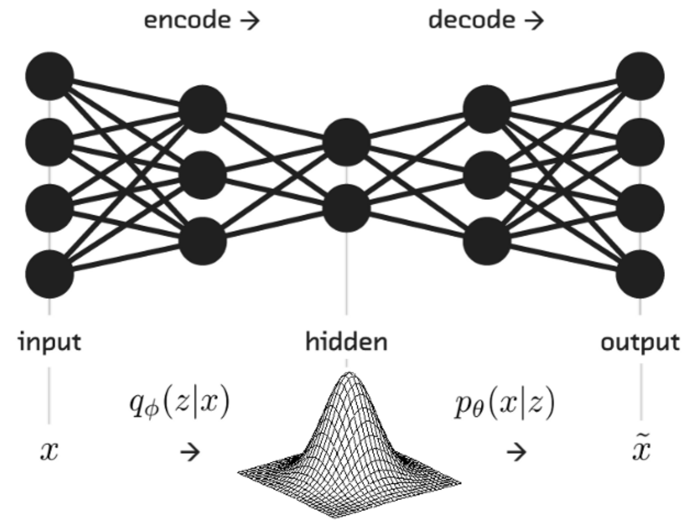
- Replace (part of) FullSim: increase speed, preserve accuracy
- Replace (part of) FastSim: decrease speed (slightly), increase accuracy
- End-to-end: map generated → reconstructed events directly
  (no dedicated simulation step)

# Techniques

## Generative Adversarial Network (GAN)

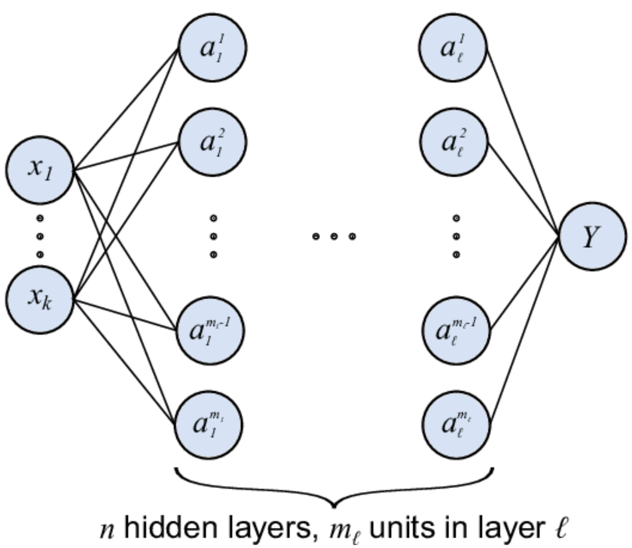



## Variational Autoencoder (VAE)

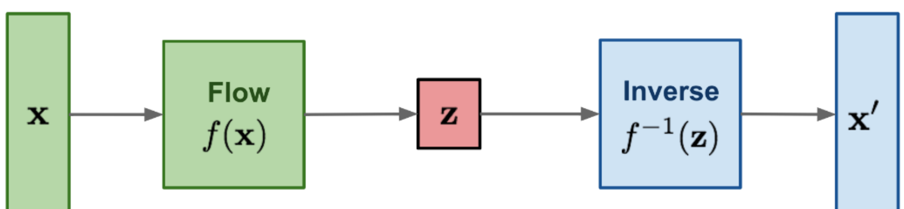



## Fully Connected Network (FCN, regression)

$\boldsymbol{a}_\ell = f(\boldsymbol{W}_\ell \boldsymbol{a}_{\ell-1} + \boldsymbol{b}_\ell)$




## Normalizing Flow (NF)

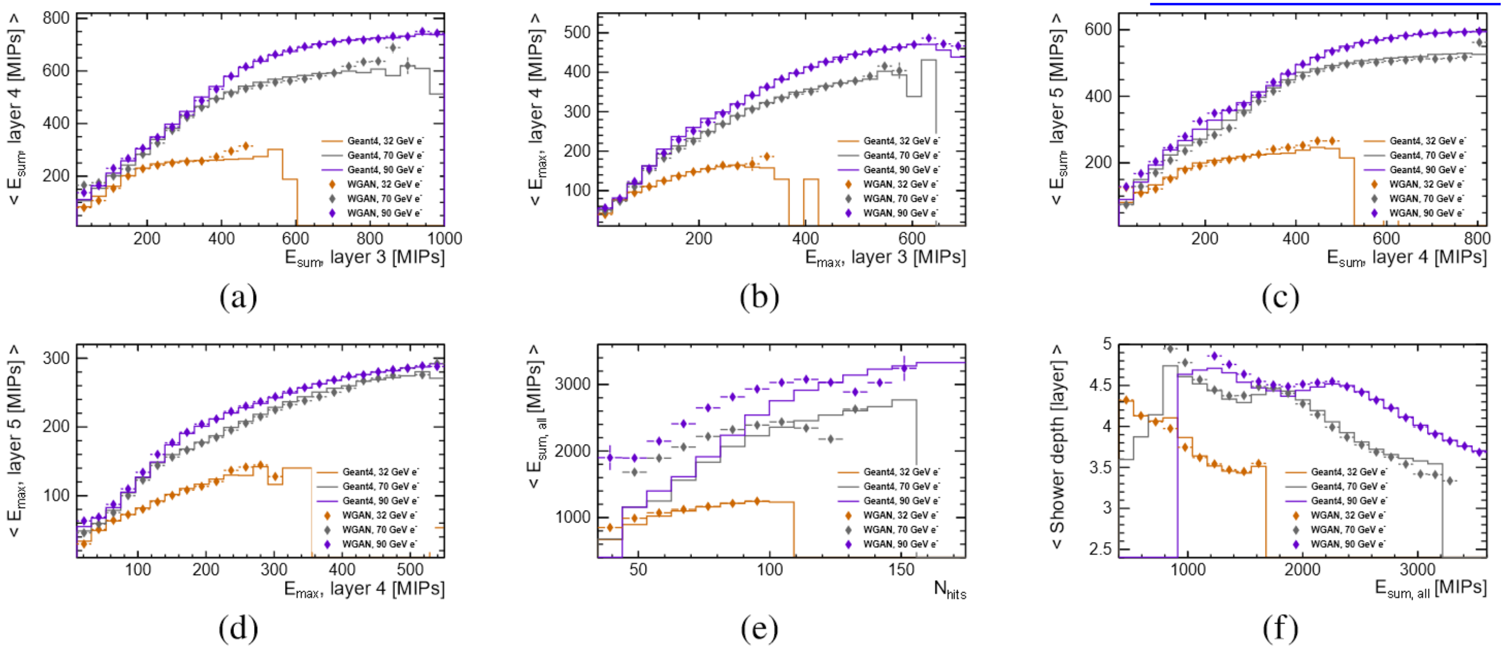



(source)

# Considerations for GANs

- GANs seem like a natural solution, but difficult to train:
  - Iterative process: alternate between training discriminator & generator → not mathematically guaranteed to converge
  - Mode collapse: starts to ignore part of input data/features
  - Vanishing gradient: unable to improve weights in training
- Some improvements are possible:
  - e.g. Wasserstein loss function helps avoid mode & gradient issues
    - Shown to improve results in HEP simulation

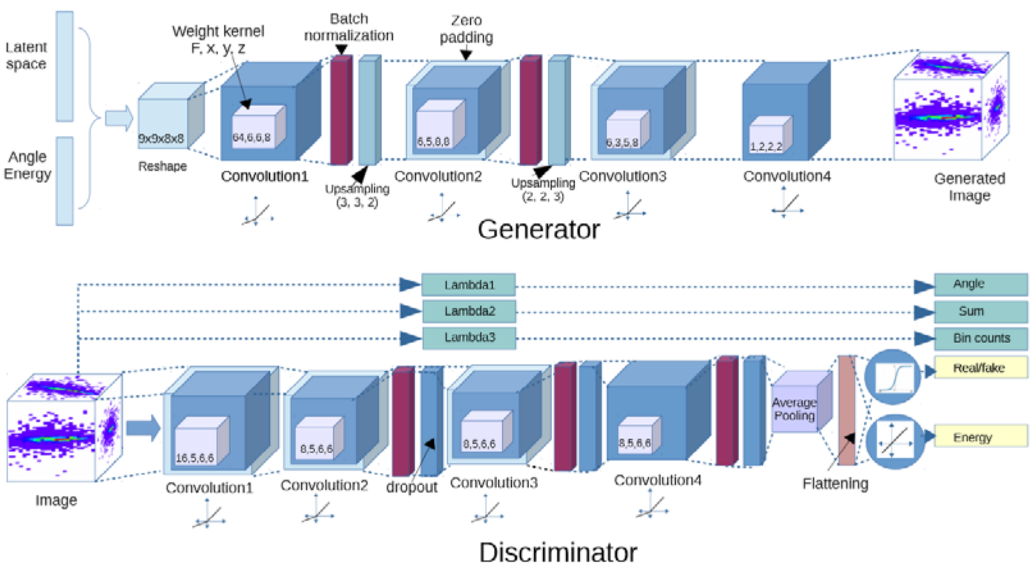
arXiv:1807.01954 Fig. 9

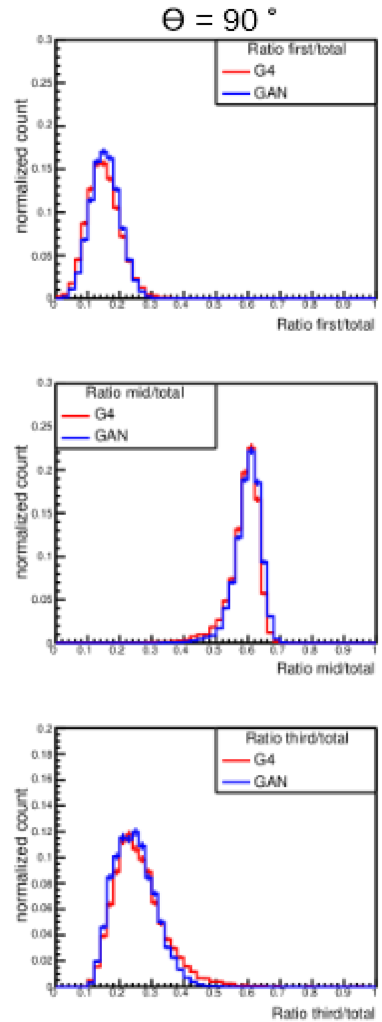


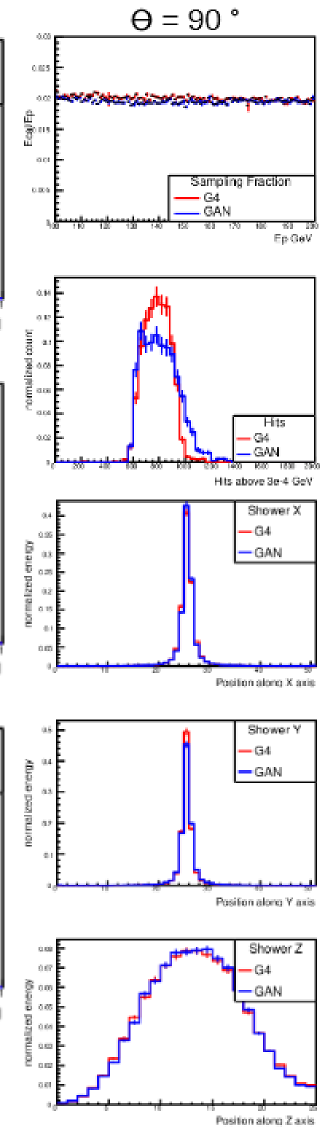
# More GAN Results

- 3D GAN w/ several physics terms included in loss function
- Generation: 4 ms/event on GPU (GTX 1080)
- Geant4: 17 sec/event on CPU (Xeon 8180)

➢ 4250× speedup, with reasonable agreement in many physics quantities



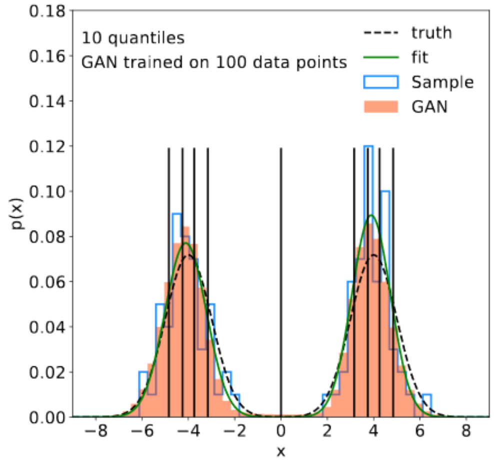



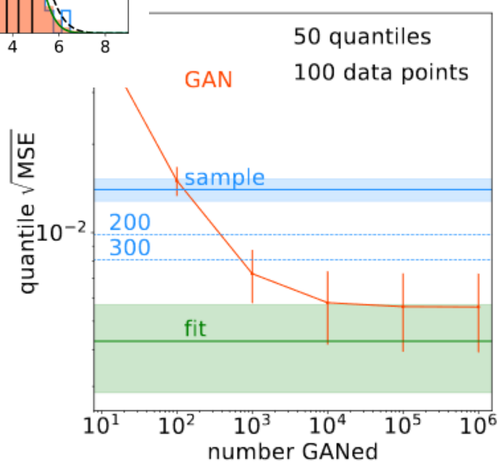

[arXiv:1912.06794](https://arxiv.org/abs/1912.06794)

# Further GAN Developments

- Demonstration that GANs *can* reduce statistical uncertainty beyond training sample by learning to interpolate:

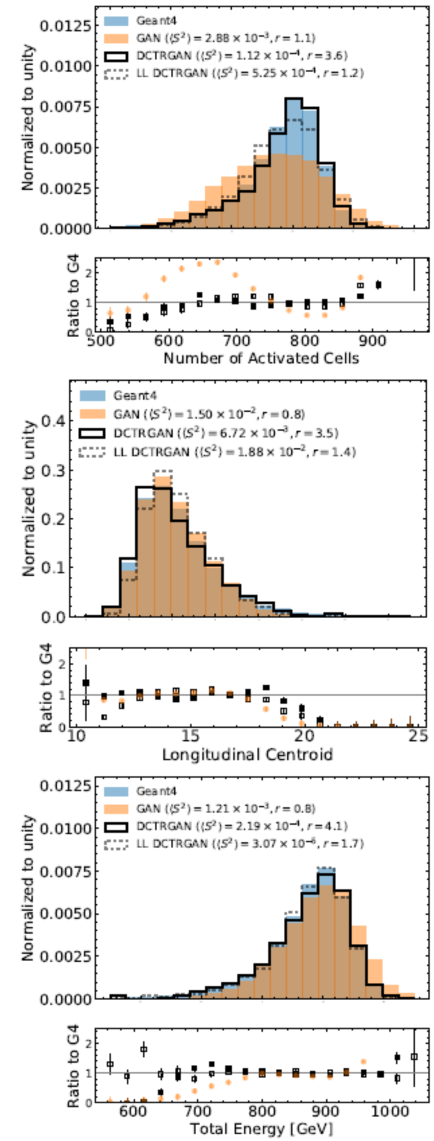



arXiv:2008.06545




- Possible to improve GAN results with an additional classifier: "DCTRGAN"
  - o Trained to reweight events after GAN training finishes

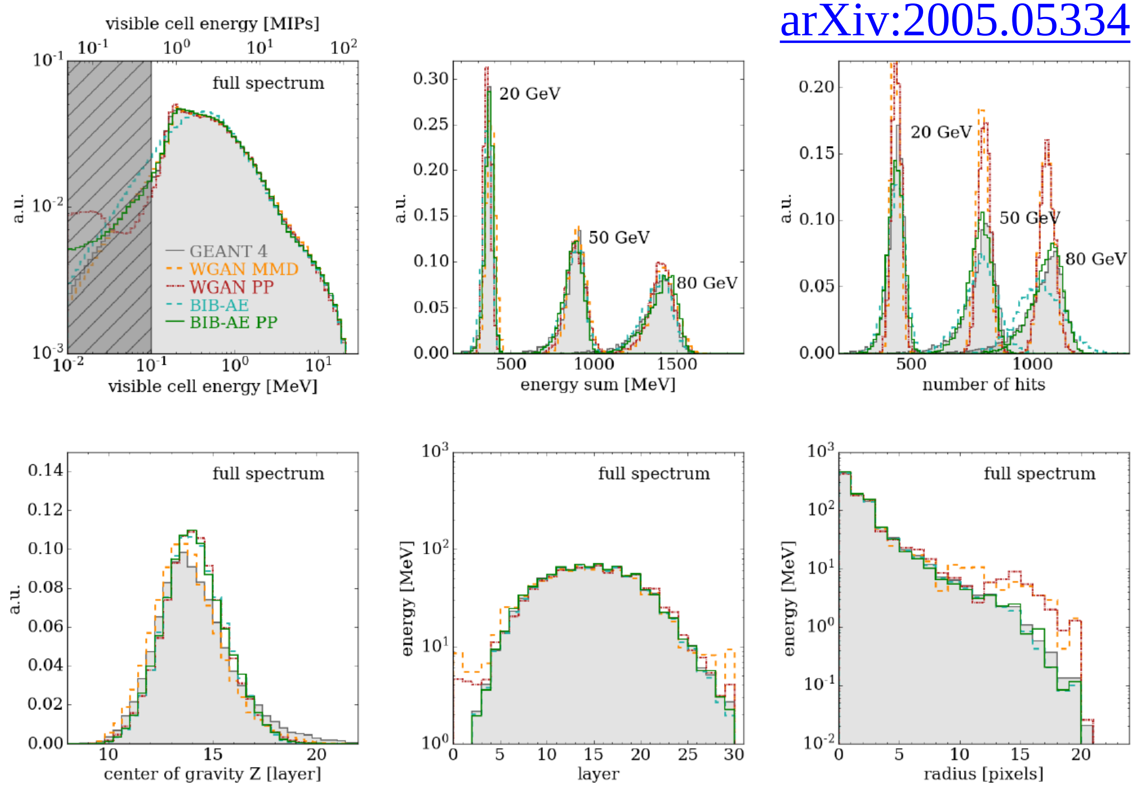
arXiv:2009.03796



# Autoencoders

- Basic: learn compressed representation ("latent space") of inputs, then "reconstruct" output
- Variational: learn *probability distribution* of latent space
  - Better for generative output
  - Still need to make sure important information isn't discarded
- Bounded Information Bottleneck:
  - Generalization/combination of VAE and GAN
  - Aimed at ILC imaging calorimeters
    - Similar to CMS HGCal
  - Improves on standard GANs, but still needs postprocessor network for best results

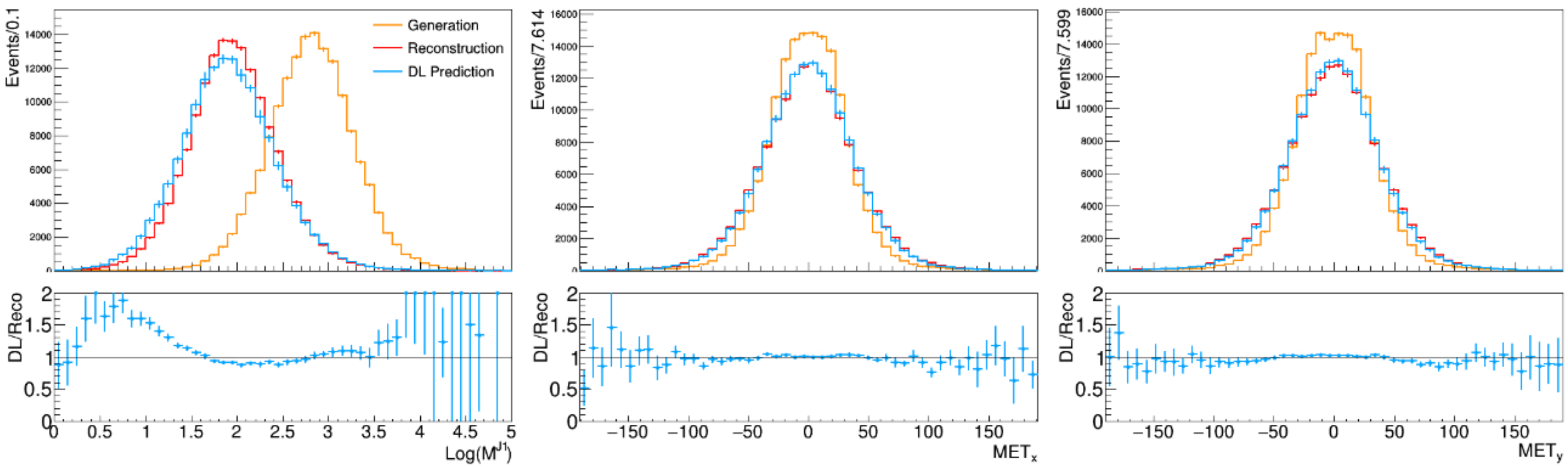
[arXiv:2005.05334](https://arxiv.org/abs/2005.05334)



# Regression

- Directly map inputs to outputs
- Can be used for either simulation or end-to-end
  - Promising results for end-to-end approach: analysis-specific targets (known backgrounds, variables)
    - Mitigates concerns about rapidly changing conditions & algorithms
- Other architectures also being explored: auto-regressive, etc.

[arXiv:2010.01835](https://arxiv.org/abs/2010.01835)



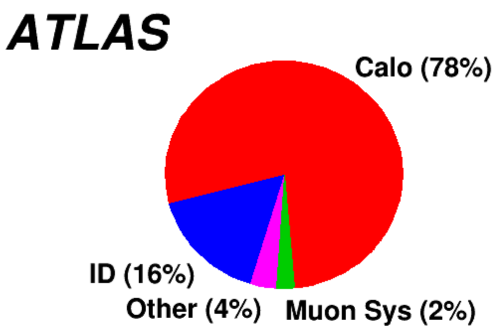
# Experiment Perspective

- ML for simulation provides natural avenue to utilize *heterogeneous computing resources* (GPUs, FPGAs, HPCs, etc.)
  - Inference as a service can facilitate this
- Need to balance *tradeoffs*:
  - Continuing to find significant developments in architectures and mathematical foundations for generative ML
    - Primarily via demonstrations in limited-author papers
    - Crucial work toward ultimately better results
  - Experiments need solutions implemented and tested for Run 4 (at least)
    - Much larger scale than limited-author papers can achieve
    - Technical details to be worked out:<br>Integration w/ Geant4? Standalone implementations? etc.

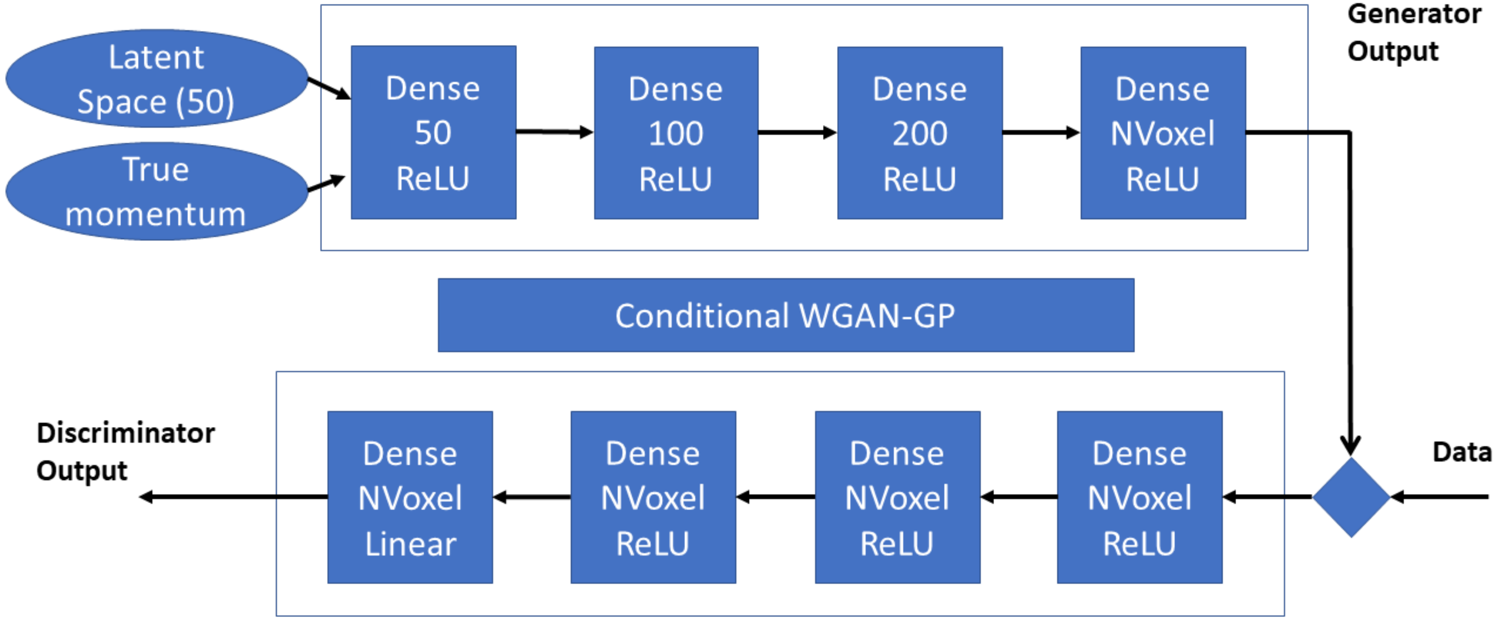
# ATLAS: FastCaloGAN

- Calorimeters use majority of CPU in (full) detector simulation
- Training: detector segmented into 100 η slices; separate electron, photon, pion samples
- Total of 300 GANs created

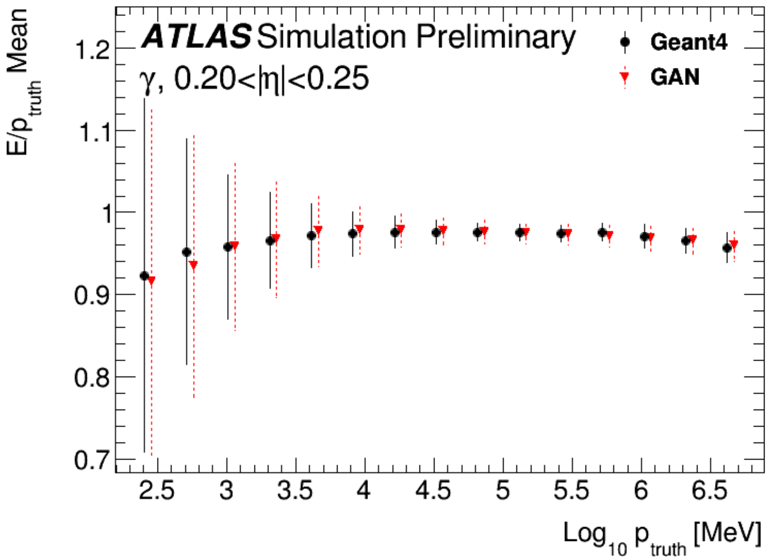
(more info)



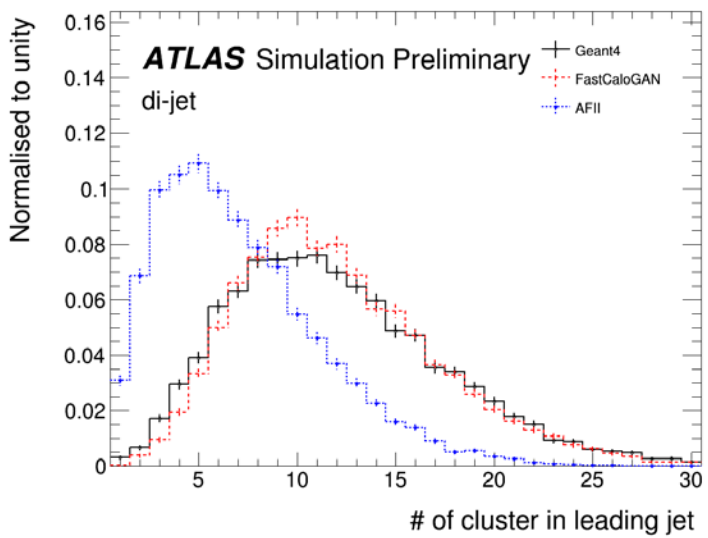

Subdetector CPU fraction for 50 ttbar events
MC16 Candidate Release

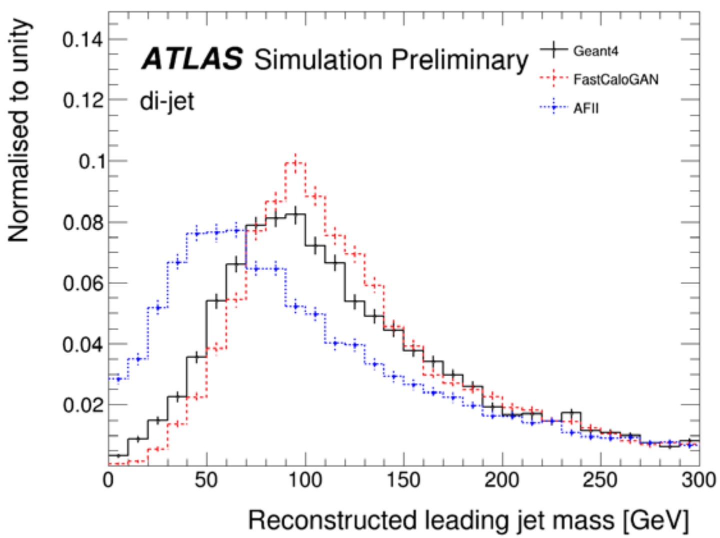


# FastCaloGAN Results



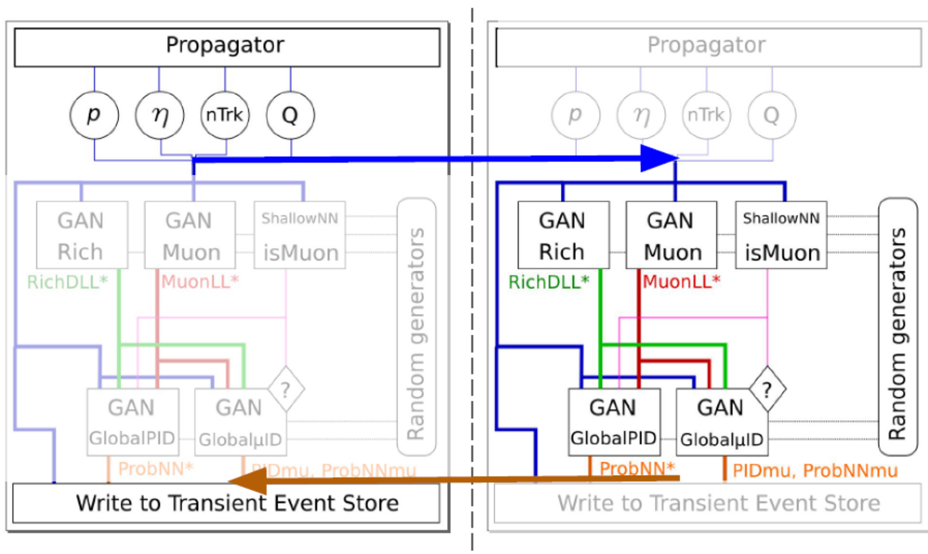

- Significant improvement over previous fast simulation (AFII)
- Good modeling of both electromagnetic and hadronic objects, including boosted regime

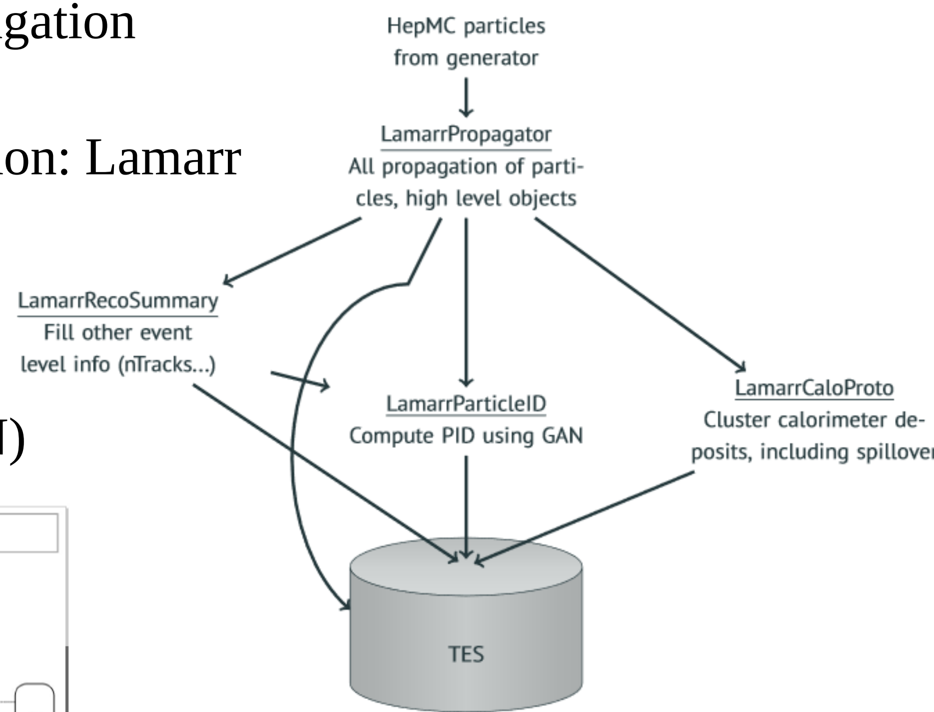


# LHCb: Particle ID in Lamarr

- Full simulation uses 95–99% of CPU time
  - Dominated by optical photon propagation & calorimeter showering
- Developing custom ultra-fast simulation: Lamarr
  - Faster than similar Delphes setup!
- Stacked GANs for PID
- Also investigating GANs for calorimeter response (and VAE+GAN)

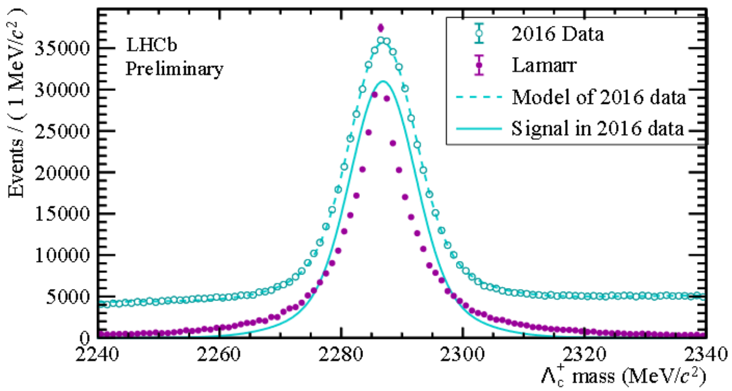


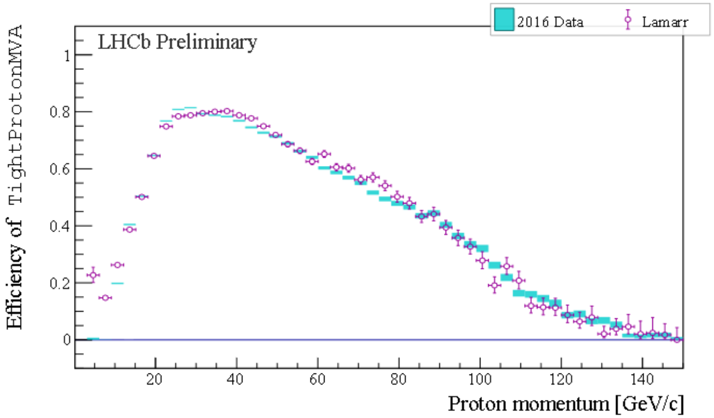




[CHEP2019 (1)]
[CHEP2019 (2)]
[ICHEP2020]

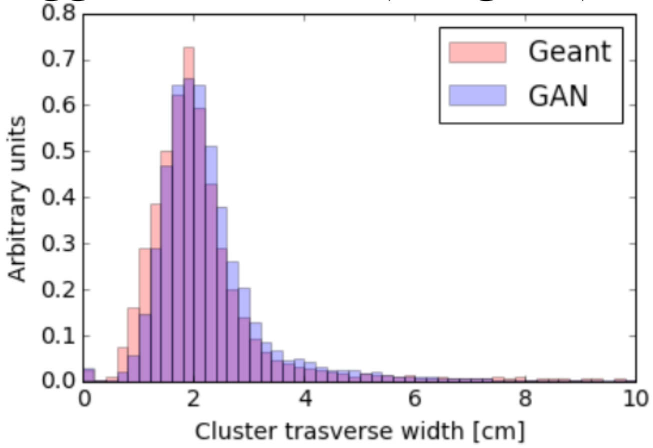
# LHCb GAN Results

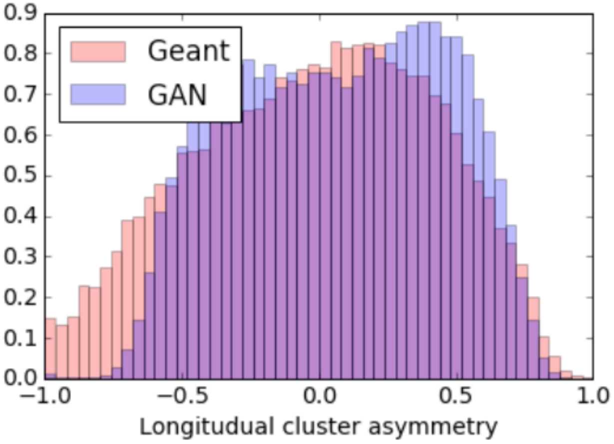
- Promising initial results for PID
  - Further optimizations ongoing




- Calorimeter GAN reproduces some distributions well
  - Struggles w/ others (marginal)



# CMS Simulation

- CMS FullSim is 4–6× faster than baseline Geant4
  - Numerous technical optimizations & physics-preserving approximations
  - Sustained effort to commission and adopt new Geant4 versions
- CMS FastSim application: 60–100× faster than FullSim
  - Includes sim- and reco-level optimizations (tracking)
  - Currently used for generation of large supersymmetric model scans, some studies of systematic uncertainties

➢ Well-positioned for Run 3, but further acceleration crucial for Phase 2

- Exploring latest architectures and use cases described here:
  BIB-AE, DCTRGAN, end-to-end analysis-specific regression, and more
  - Goal: develop common tools for comparison of different approaches
    - Datasets, physics validation quantities, etc.

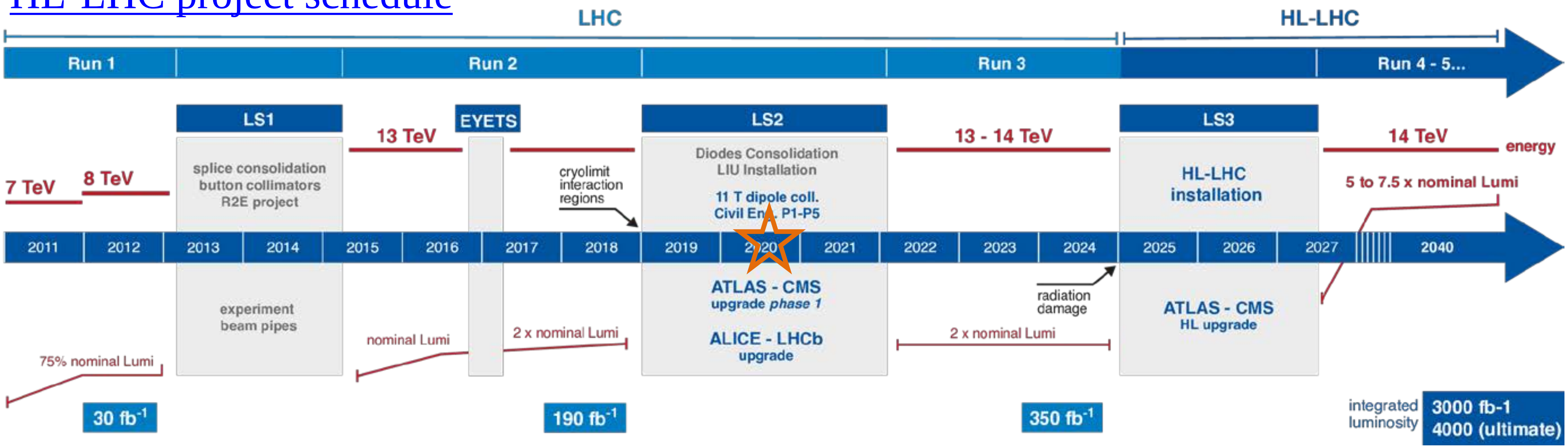
# Conclusion

- ML provides numerous possibilities for fast, accurate detector simulation
  - Can augment existing full or fast simulation
  - End-to-end approaches an interesting alternative
  - Generative (GAN, VAE) or regression algorithms can be employed
- Significant research interest in improving physical validity of results
  - Many new architectures and approaches under development
- Experiments starting to deploy GANs for fast simulation applications:
  - FastCaloGAN in ATLAS, PID GAN for LHCb
- Going forward, important transition from simplified examples to production-ready implementations
  - Experiments need to be prepared for HL-LHC computing challenges
- Bonuses: utilization of coprocessors and development of common resources
  - Also of interest to other fields that use MC simulation: neutrinos, astrophysics, etc.

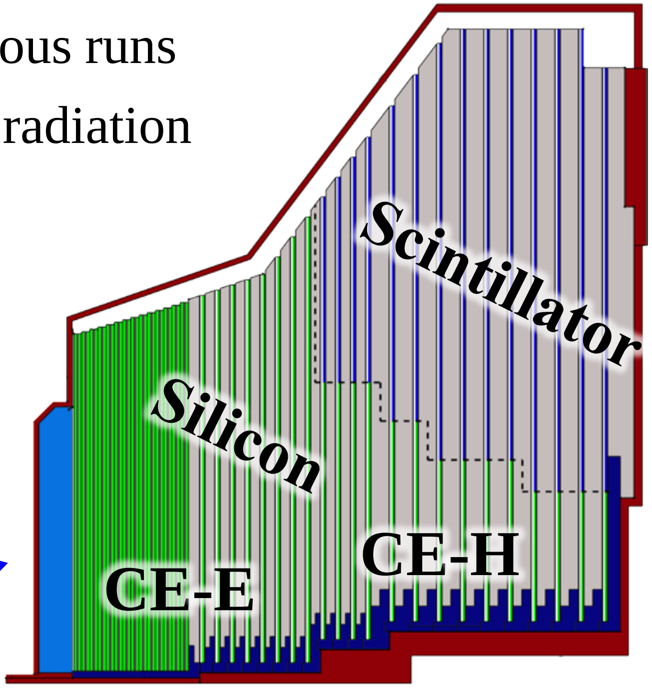
# Backup

# Upgrades

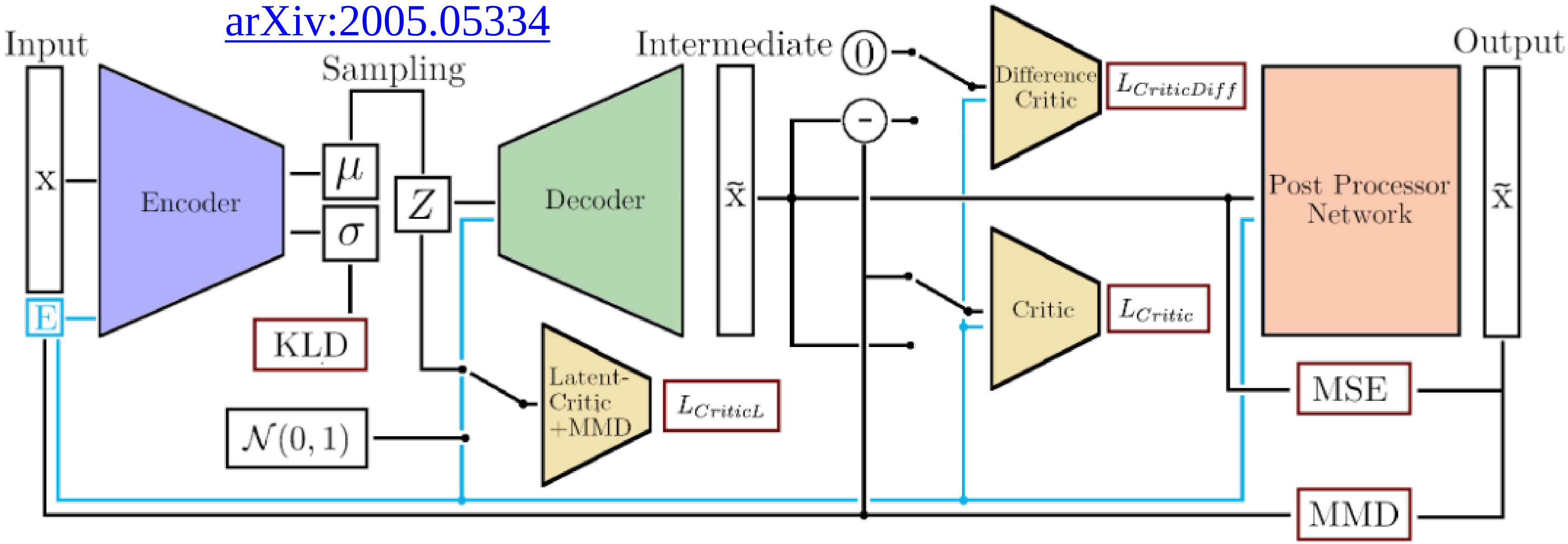
[HL-LHC project schedule](#)




- Run 4+ expected to deliver ~10× data from previous runs
  - Higher luminosity: higher occupancies, higher radiation → need new detectors!
- CMS detector upgrades include:
  - Pixel (inner tracker): 66M → 1947M channels
  - Outer tracker: 9.6M → 215M channels
  - High Granularity Calorimeter (HGCal): 85K → 6M channels



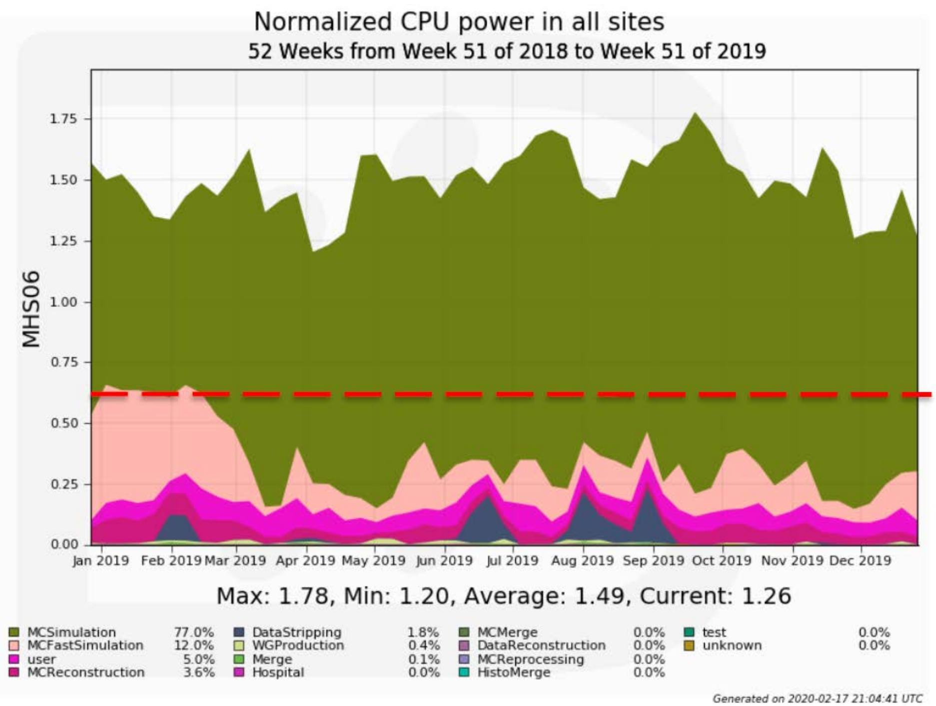
# BIB-AE Architecture

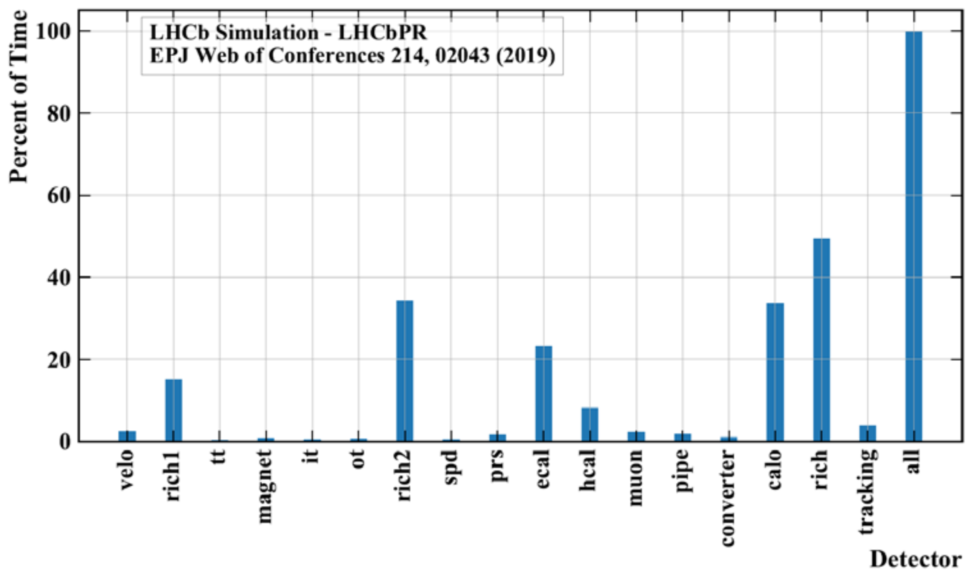



$$
\begin{aligned}
L_{\text{BIB-AE}} = & -\beta_{C_L} \cdot \mathbb{E}[C_L(E(x))] \\
& -\beta_{C} \cdot \mathbb{E}[C(D(E(x)))] \\
& -\beta_{C_D} \cdot \mathbb{E}[C_D(D(E(x)) - x)] \\
& +\beta_{\text{KLD}} \cdot \text{KLD}(E(x)) \\
& +\beta_{\text{MMD}} \cdot \text{MMD}(E(x), \mathcal{N}(0,1)).
\end{aligned}
$$

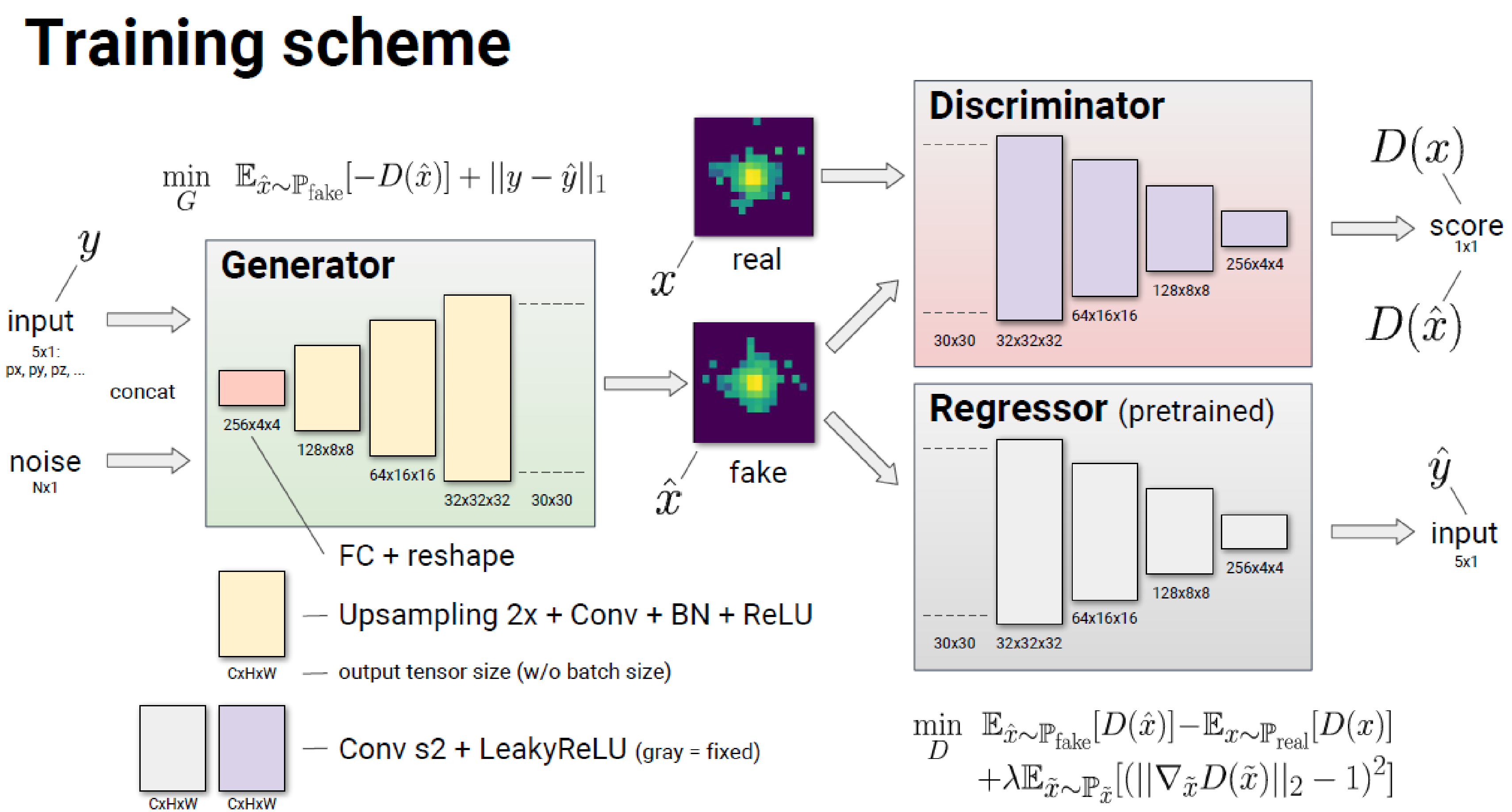
# LHCb FullSim CPU Usage

- From [M. Rama, ICHEP2020]
- Also *Eur. Phys. J. Web Conf.* 214 (2019) 02043

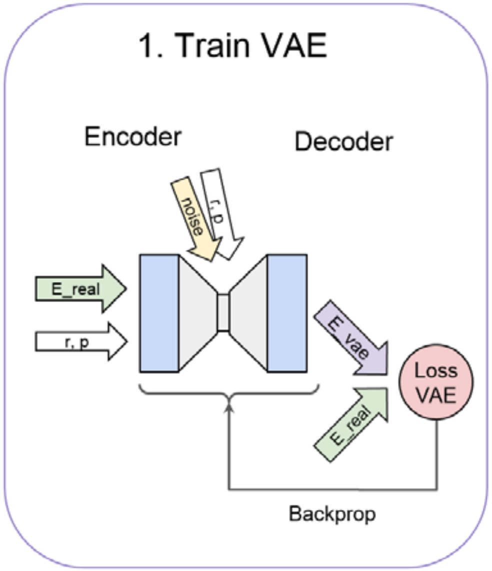


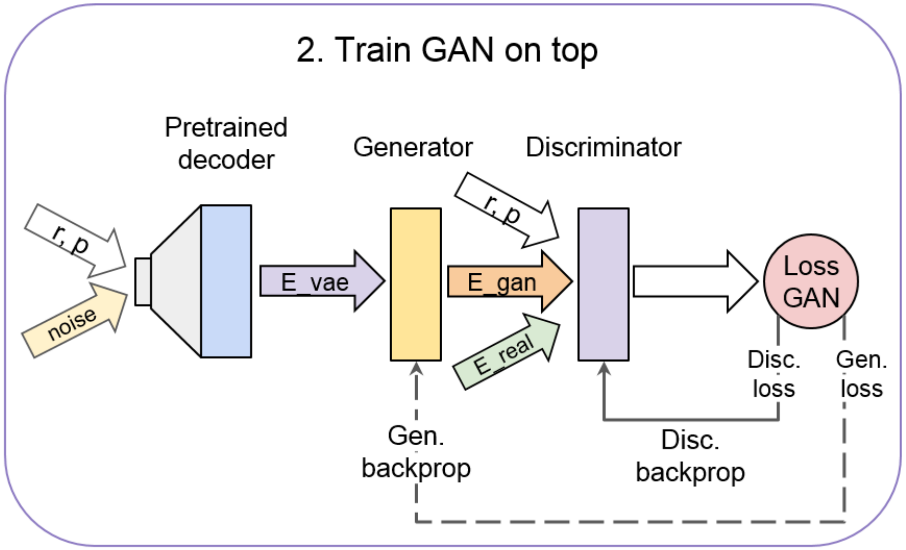
# LHCb Calorimeter GAN



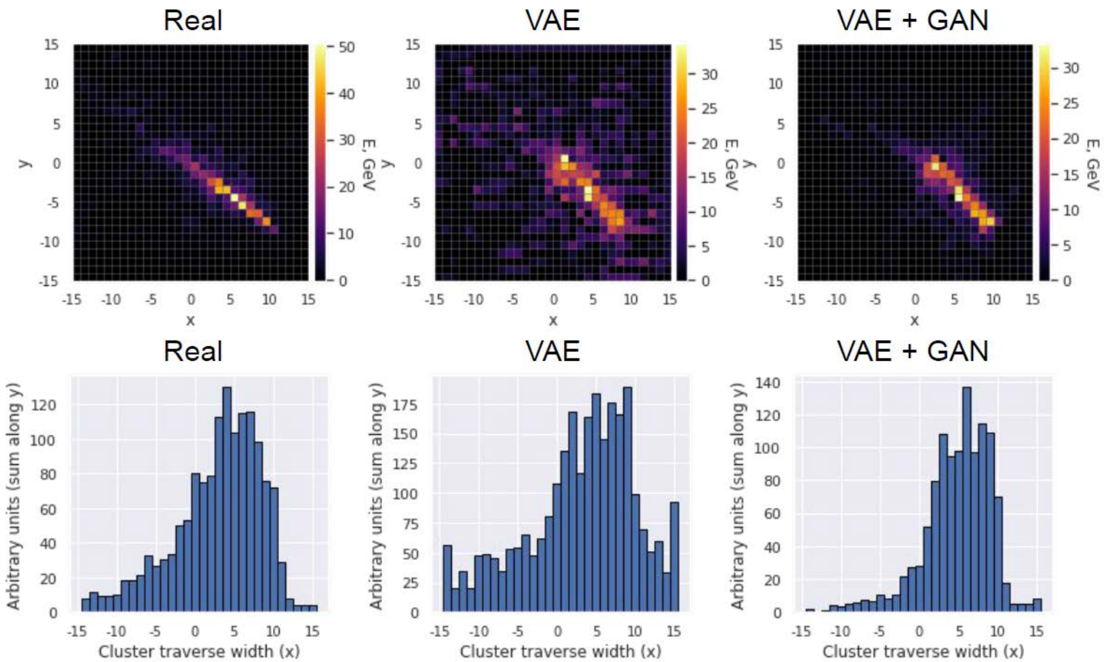
# LHCb VAE+GAN



1. Train VAE
Encoder
Decoder
noise
r, p
E_real
r, p
E_vae
Loss VAE
E_real
Backprop

2. Train GAN on top
Pretrained decoder
Generator
Discriminator
r, p
noise
E_vae
r, p
E_gan
E_real
Loss GAN
Disc. loss
Gen. loss
Gen. backprop
Disc. backprop

Real
VAE
VAE + GAN
y
x
E, GeV
Real
VAE
VAE + GAN
Arbitrary units (sum along y)
Cluster traverse width (x)