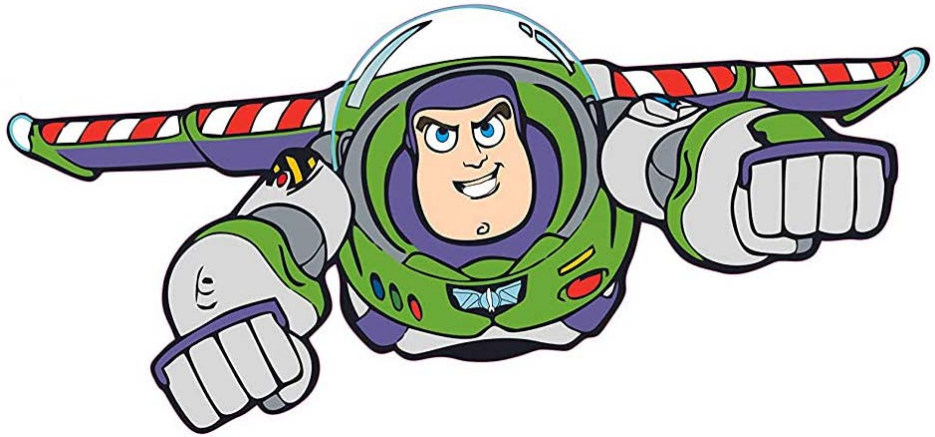
FERMILAB-SLIDES-23-059-CMS-CSAID-PPD

AI and Beyond: New Techniques for Simulation and Design in HEP

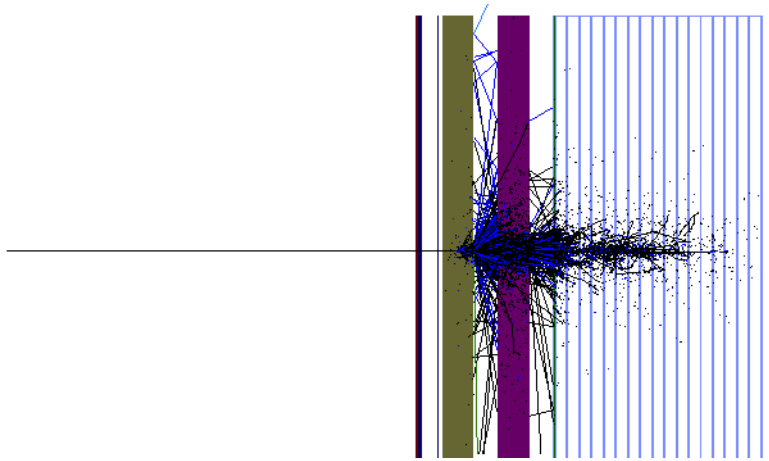
Kevin Pedro

(FNAL)

May 8, 2023



Simulation Landscape

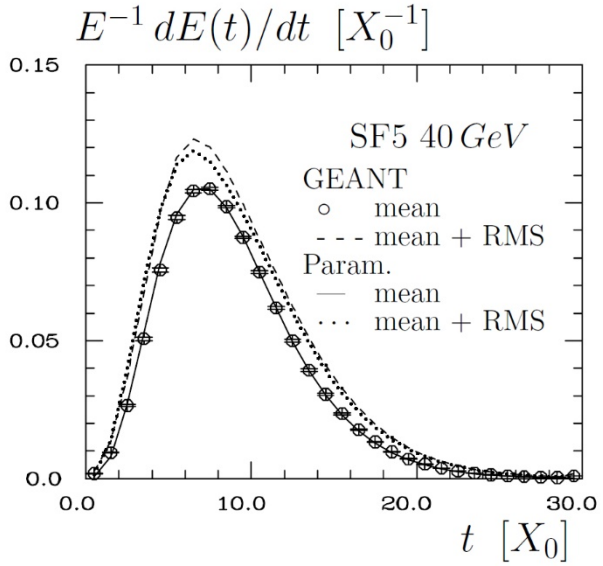


"FullSim"

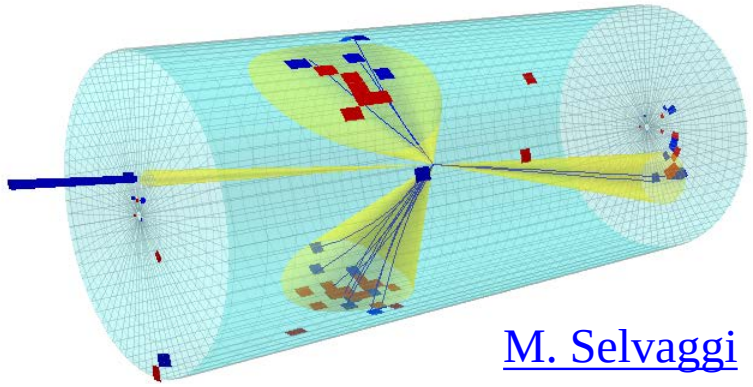
- Common software framework (i.e. Geant4)
  - Experiments can provide additional code via user actions
- Explicit modeling of detector geometry, materials, interactions w/ particles

"FastSim"

- Usually experiment-specific framework
- Implement approximations: analytical shower shapes (e.g. GFLASH), truth-assisted track reconstruction, etc.



arXiv:hep-ex/0001020



M. Selvaggi

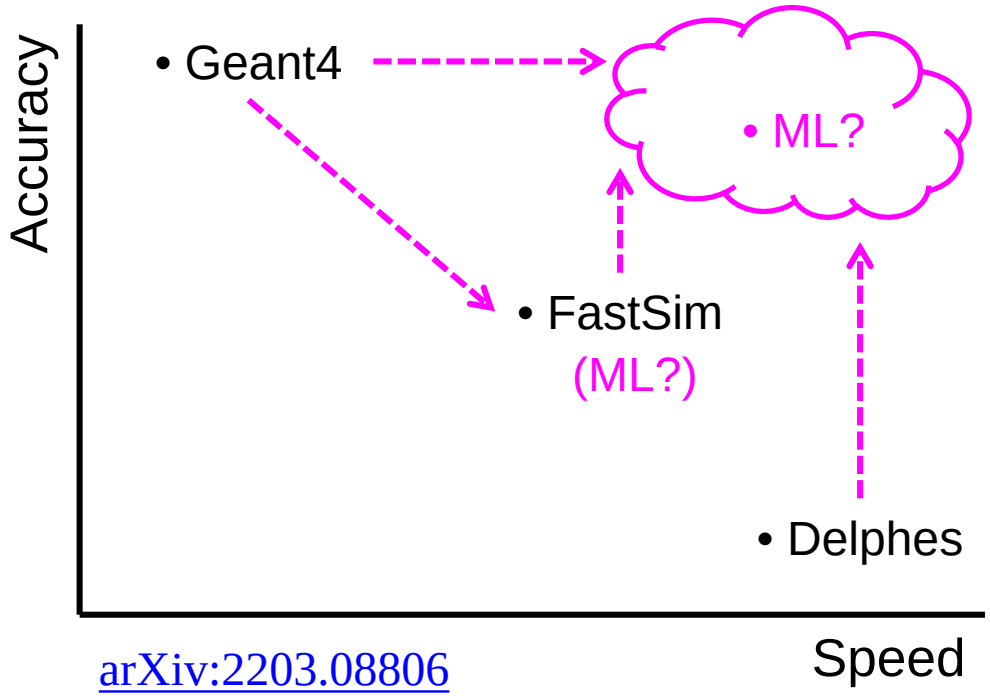
Delphes

- Ultra-fast parametric simulation
- Used for phenomenological studies, future projections, etc.

Simulation is crucial in HEP!

AI† Landscape

- Options to use ML† for sim:
  1. Replace or augment (part or all of) Geant4
  2. Replace or augment (part or all of) FastSim
- Goals:
  1. Increase speed while preserving accuracy
  2. Preserve speed while increasing accuracy
- ML can also create faster, but less accurate simulation
  - à la existing classical FastSim
    - then augment w/ more ML to improve accuracy
- Another option: replace entire chain (“end-to-end”)
  - Exciting prospect, potentially complements other cases



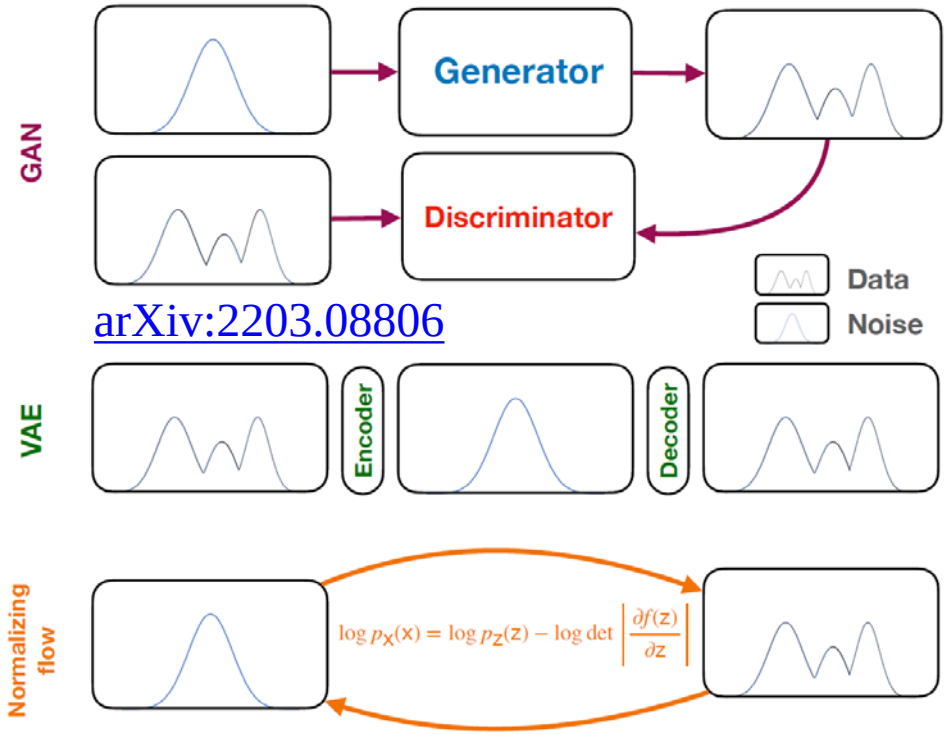
arXiv:2203.08806

ML can also be used for event generation, hadronization, etc.!

† “AI” or “ML” here: almost always deep neural networks (DNNs)

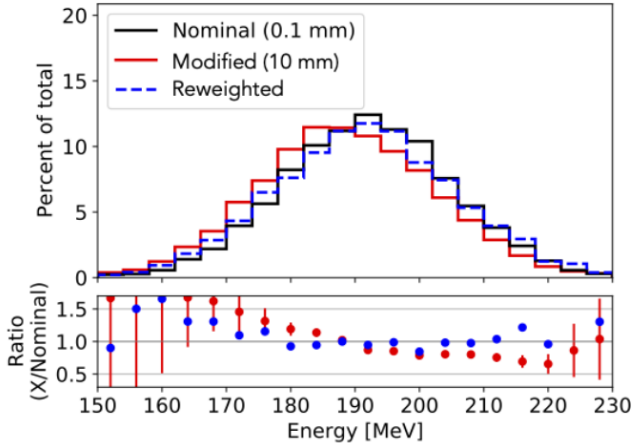
Taxonomy

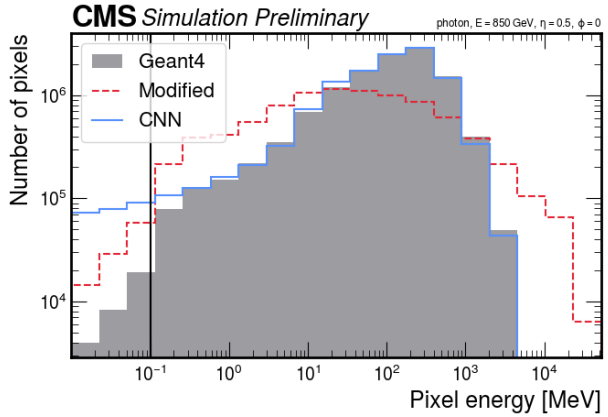
- Generative models (“replace”):
  - Usually *stochastic*
  - Generative Adversarial Networks (GANs)
  - Variational Autoencoders (VAEs)
  - Normalizing Flows (NFs)



arXiv:2203.08806

- Refinement techniques (“augment”):
  - Usually *deterministic*
  - Classification-based (reweighting)
  - Regression-based (correcting)





Metrics

- Speed only matters if needed accuracy is achieved
  - Wrong answers can be obtained infinitely fast
- Looking at 1D histograms: not good enough!
  - Can miss high-dimensional correlations
- Best category: **integral probability metrics**

$$D_{\mathcal{F}}(p_{\text{real}}, p_{\text{gen}}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{\mathbf{x} \sim p_{\text{real}}} f(\mathbf{x}) - \mathbb{E}_{\mathbf{y} \sim p_{\text{gen}}} f(\mathbf{y})|$$

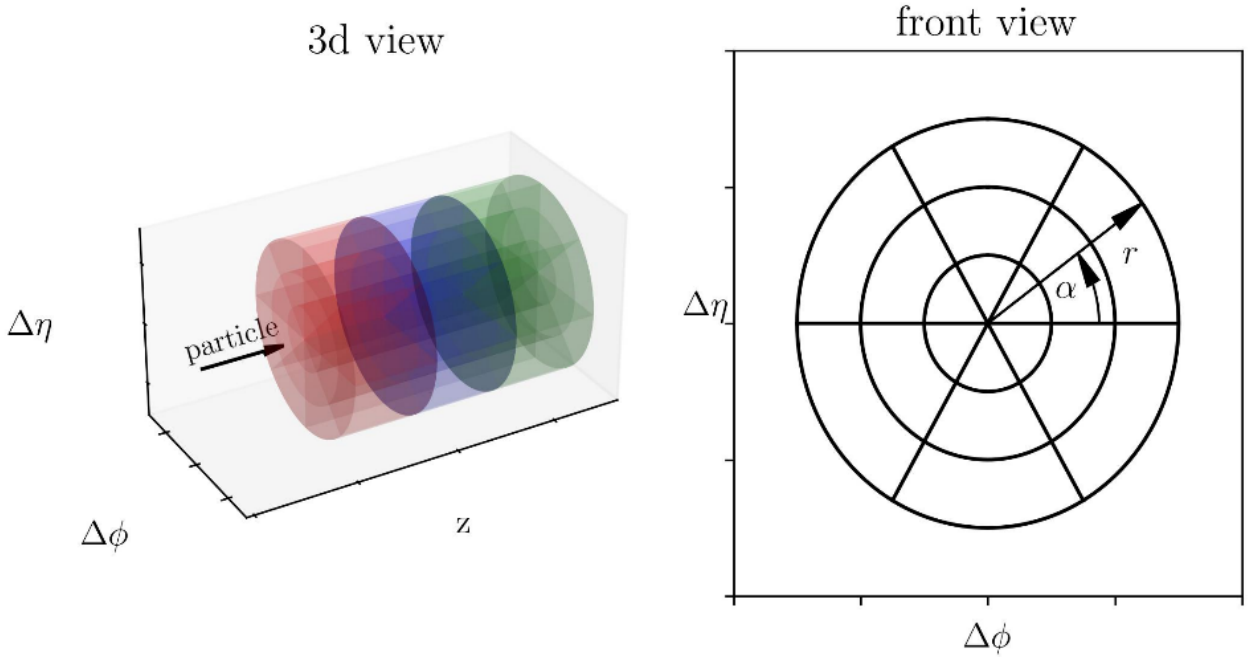
  - *Wasserstein distance* W_1: $\mathcal{F}$ is set of all K-Lipschitz functions
    - Only works well in 1D, biased in high-D
  - *Maximum mean discrepancy* (MMD): $\mathcal{F}$ is unit ball in reproducing kernel Hilbert space
    - Depends on choice of kernel
  - *Fréchet distance*: W_2 distance between Gaussian fits to (high-D) feature space
    - Features can be hand-engineered or obtained from NN activations
- Another interesting category: *classifier scores*
  - Train NN to distinguish real vs. generated
  - AUC score ranges from 0.5 to 1.0
- *Fréchet Particle Distance* most clearly distinguishes between two similar approaches (message passing GAN and generative adversarial particle transformer)

	FPD $\times 10^3$	KPD $\times 10^3$	W_1^M $\times 10^3$
Truth	0.08 ± 0.03	-0.006 ± 0.005	0.28 ± 0.05
MPGAN	$\mathbf{0.30 \pm 0.06}$	$\mathbf{-0.001 \pm 0.004}$	$\mathbf{0.54 \pm 0.06}$
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08

arXiv:2211.10295

Common Datasets

#549



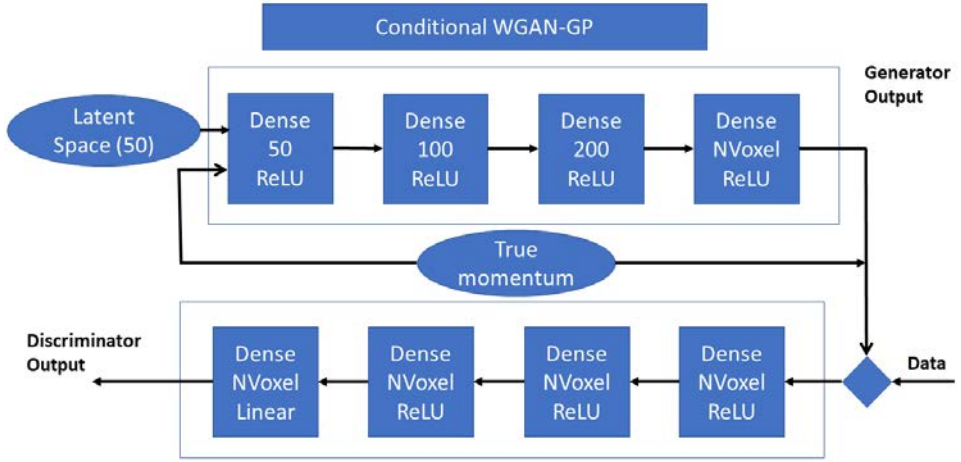
- CaloChallenge: first competition for generative ML for detector simulation
- Three public datasets provided:
  1. Low granularity, irregular geometry (based on ATLAS calorimeter), photon & pion showers
  2. Medium granularity, silicon-tungsten sampling calorimeter, electron showers
  3. High granularity, otherwise same as #2

- Common datasets are crucial to compare different generative methods
  - Using all the metrics just discussed (and more)
- Already many new methods developed for the challenge!
  - Some will be shown at CHEP this week

Generative Models at Colliders: ATLAS

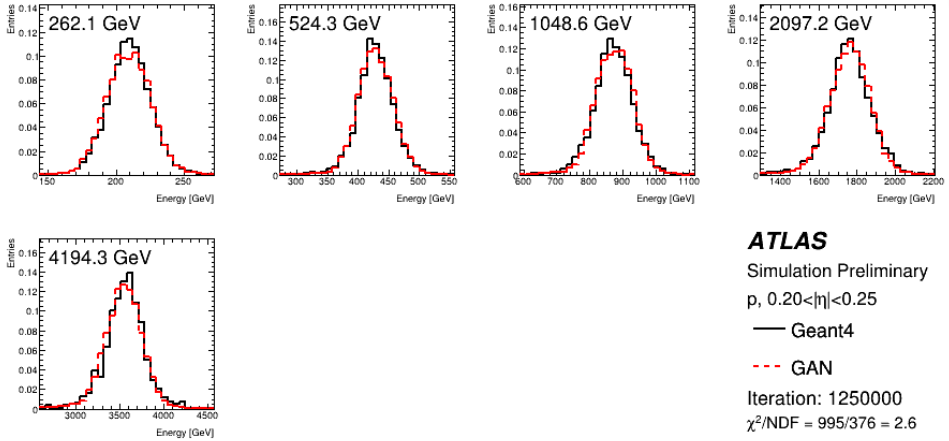
#420

#424

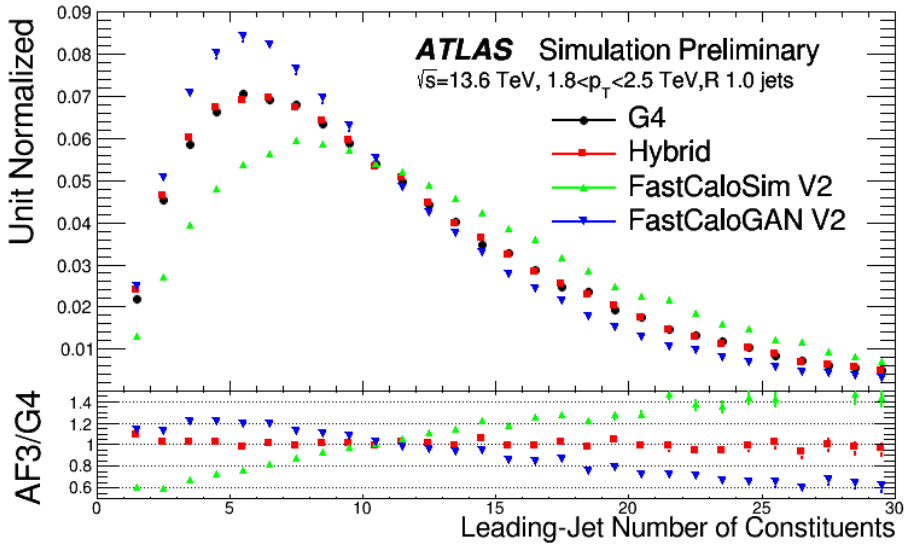


- FastCaloGAN architecture: W_1 used as loss
  - Stabilizes GAN training & performance
- Separate GANs trained for 100 η slices and for each particle type: γ, e, $\pi^{\pm}$, p
  - Hyperparameters optimized for each particle
- Irregular geometry voxelized for training
- Incorporated in AtlFast3 along with FullSim and FastSim modules (depending on particle type, etc.)

- Good agreement for protons (new!)

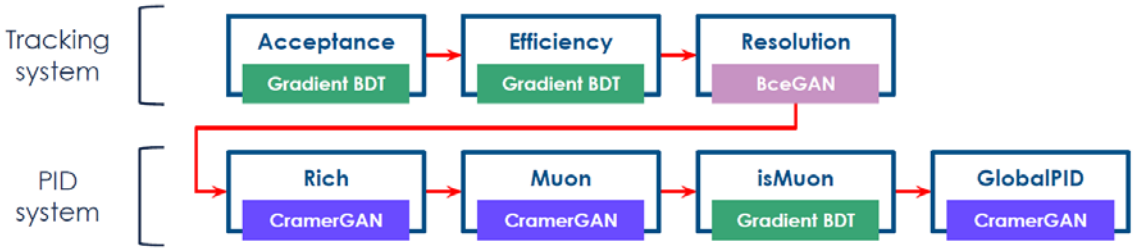


- Hybrid approach improves modeling of high-level quantities



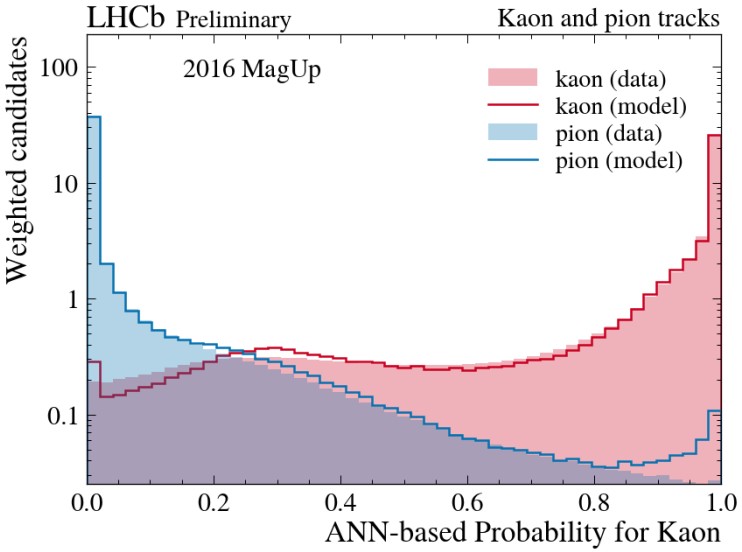
Generative Models at Colliders: LHCb

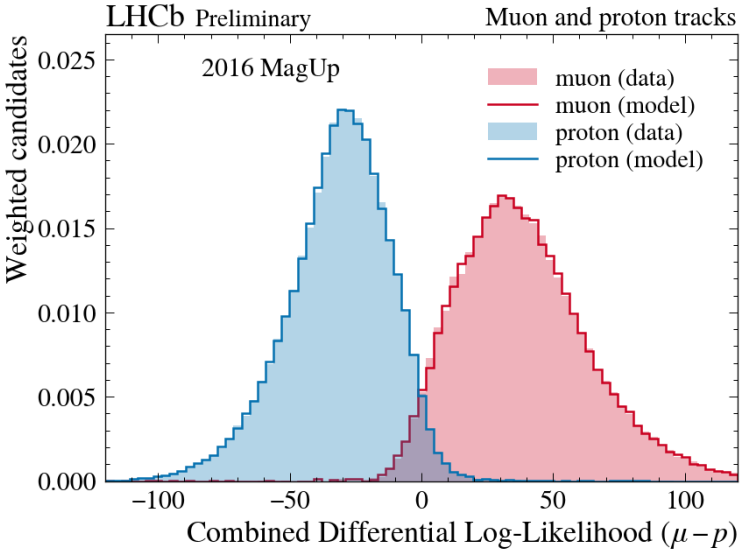
#629

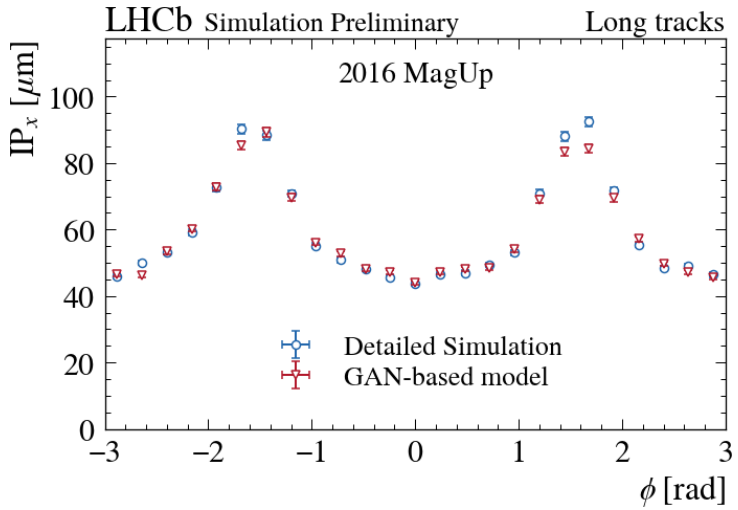


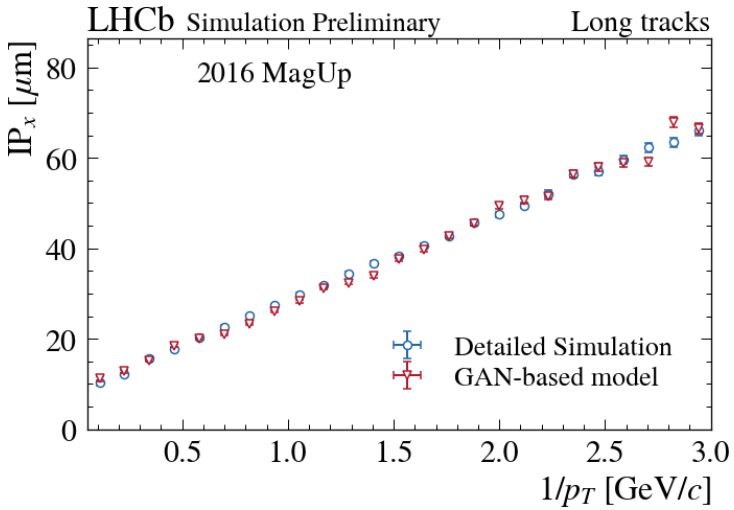
- "Stacked GAN" approach to parameterize different detector aspects
  - Cramér distance related to W_1
- Tracking resolution: well reproduced in p_T & φ

- Global PID variables also well reproduced:
  - Top: $K^{\pm}$ vs. $\pi^{\pm}$
  - Bottom: μ vs. p







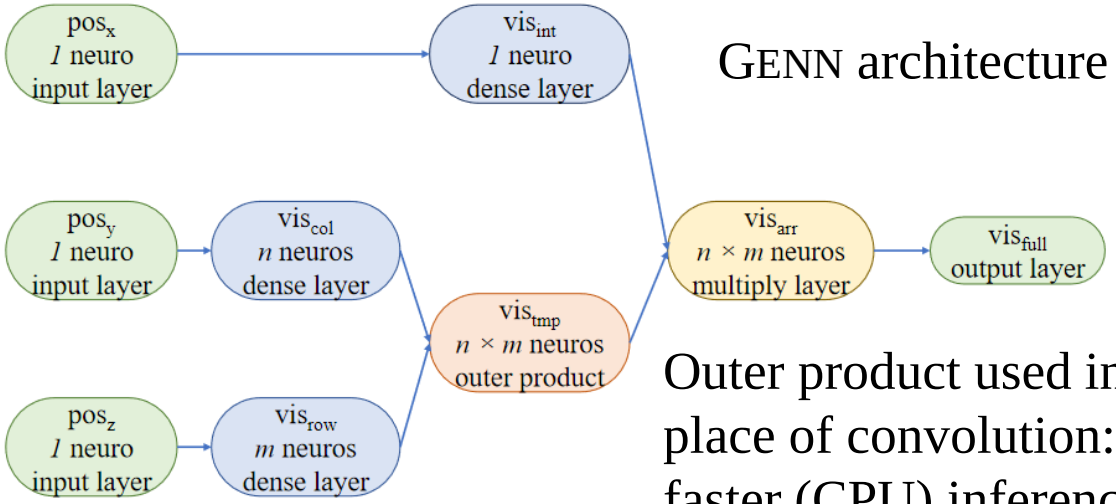


LHCB-FIGURE-2022-004

Generative Models for Neutrino Experiments

- Outside of collider physics: major computing hurdle is optical photon propagation

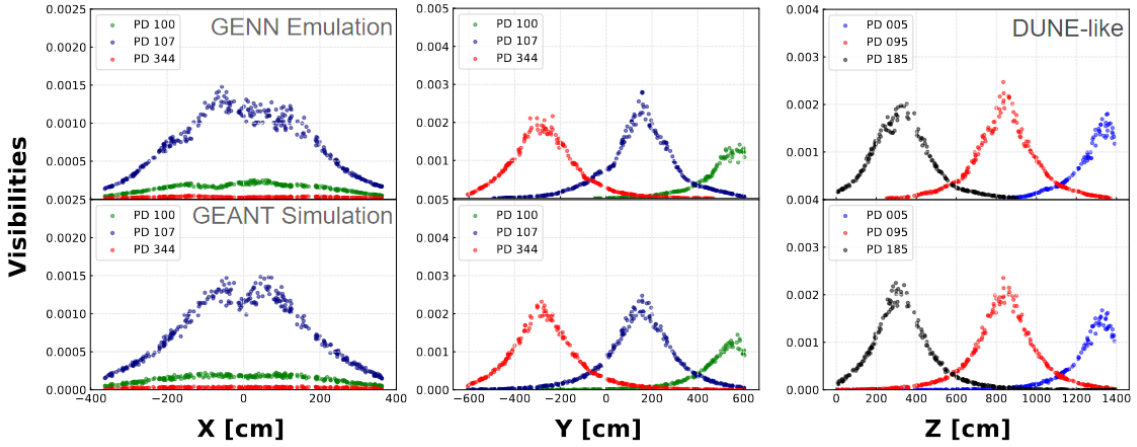
arXiv:2211.01505



GENN architecture

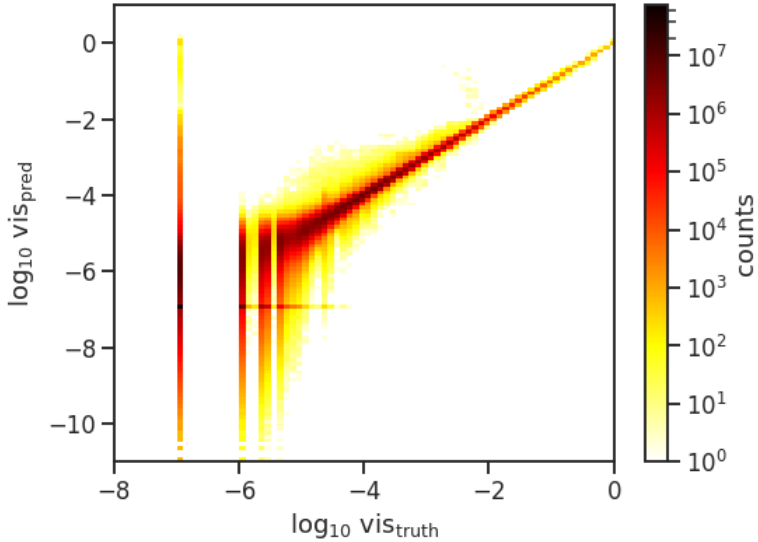
Outer product used in place of convolution: faster (CPU) inference

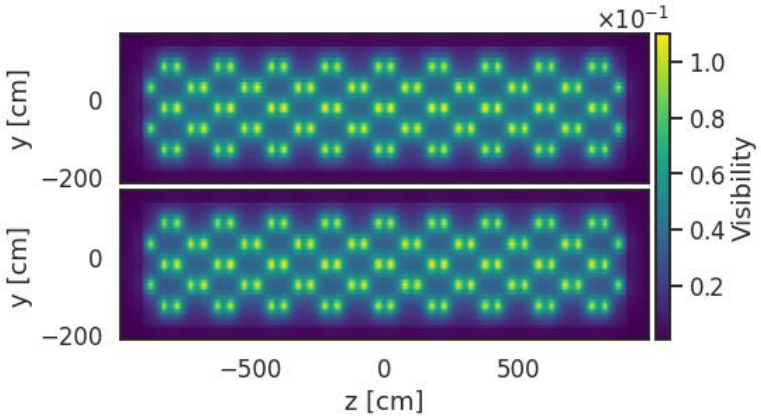
arXiv:2109.07277



- SIREN: SInusoidal REpresentation Networks
  - Sine function used for activation
    - Can reproduce high-frequency features
    - Continuous & differentiable → represents gradients as well as values

#463



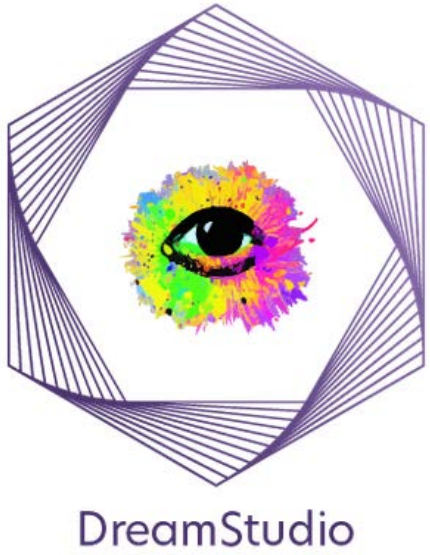


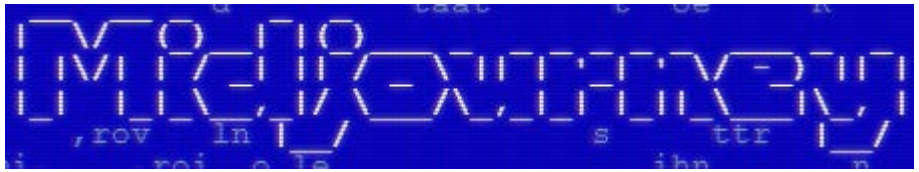
(ICARUS detector;
top: photon library, bottom: SIREN)

Generative AI in Industry

Diffusion

Autoregression





- No GANs, VAEs, NFs in sight!
  - Although sometimes used as components of larger models
- GANs especially disfavored: lack of convergence, mode collapse, vanishing gradients…
  - Issues can be mitigated, but newer architectures benefit from avoiding them altogether

Use Industry AI for HEP?

- DreamStudio results for prompt:

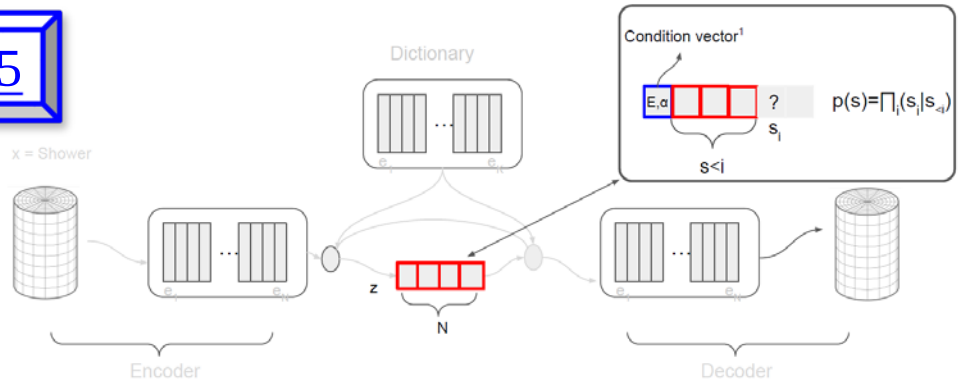
> A GEANT4 simulation of a pion shower with energy 100 GeV in the Compact Muon Solenoid High Granularity Calorimeter at the CERN Large Hadron Collider, a particle physics experiment

- Cool-looking, but not probably not quite accurate enough out of the box…

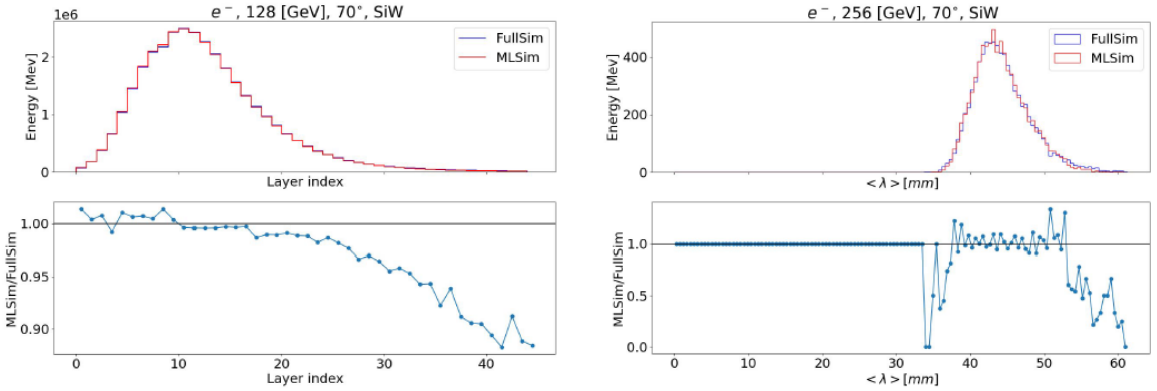
New Generative Approaches for HEP

- Transformer architecture with vector quantized VAE to learn latent space and autoregressive prior to sample from latent space

#405



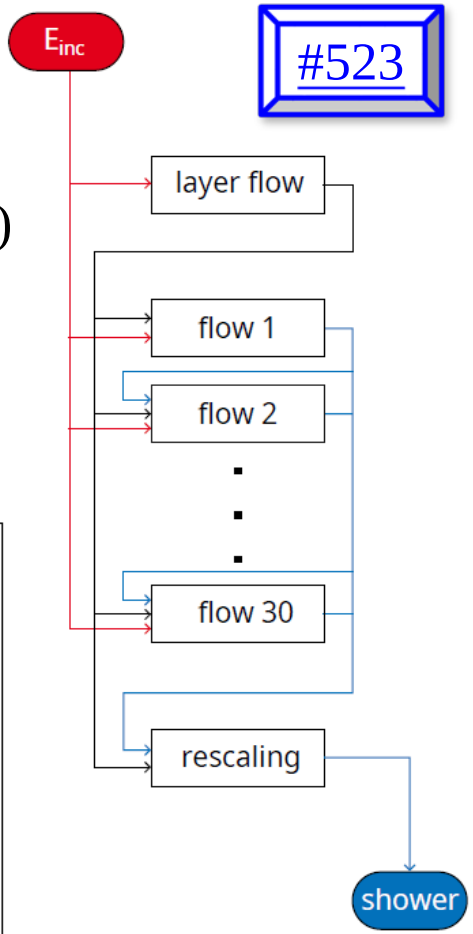
- Good modeling of moments, work ongoing to improve sampling in tails

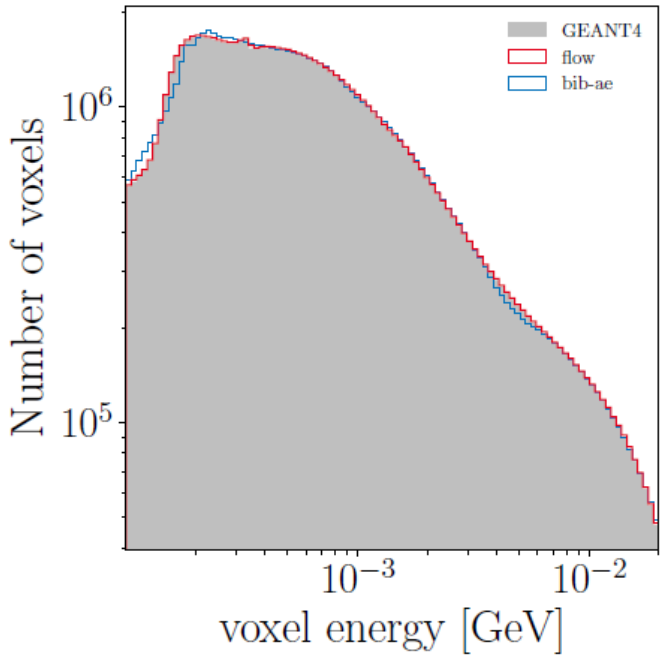


Longitudinal profile *Longitudinal first moment*

- Extension of L2LFlow (arXiv:2302.11594, series of stacked, conditioned NFs) to full ILD ECAL geometry (30×30×30)
  - Outperforms BIB-AE† (GAN-VAE hybrid) in distributions, metrics: W_1, classifier AUC

#523

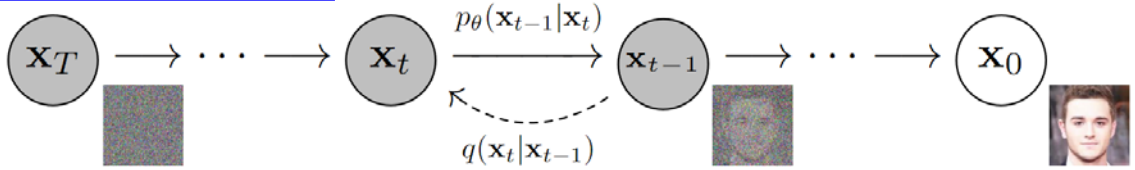




†Bounded Information Bottleneck Autoencoder

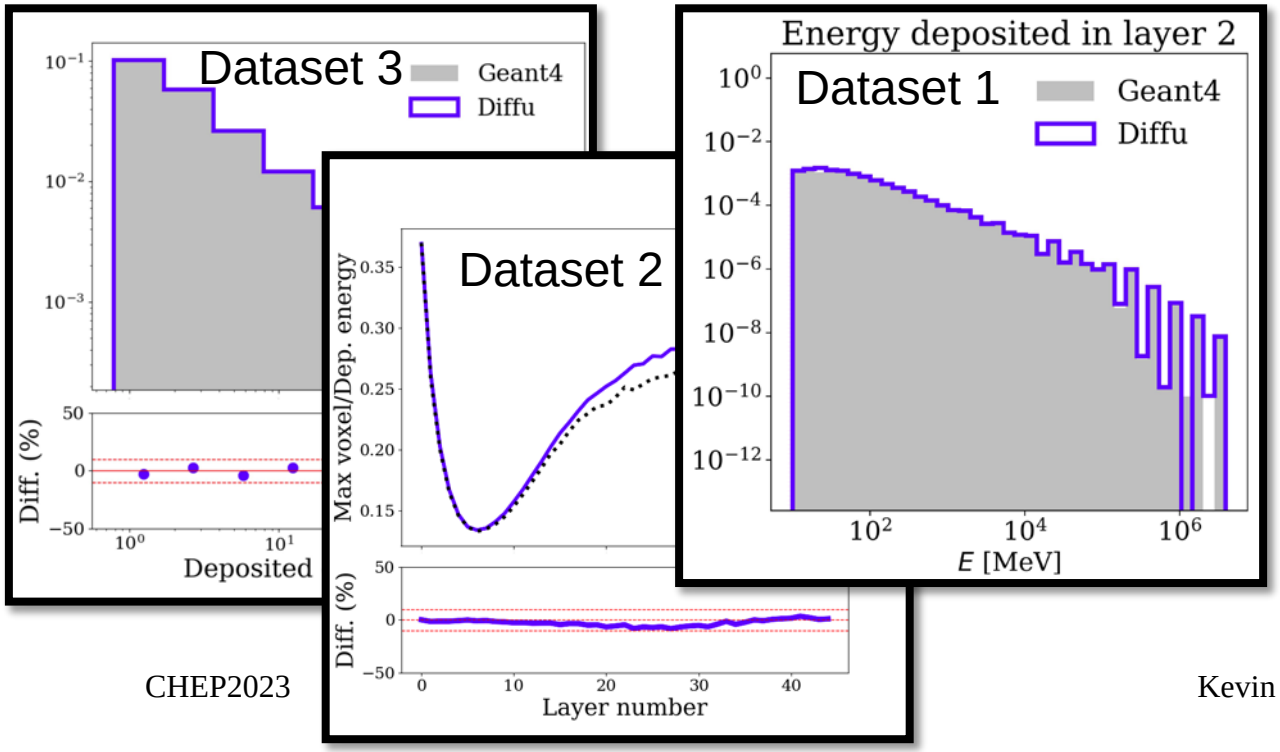
Diffusion for HEP

arXiv:2006.11239



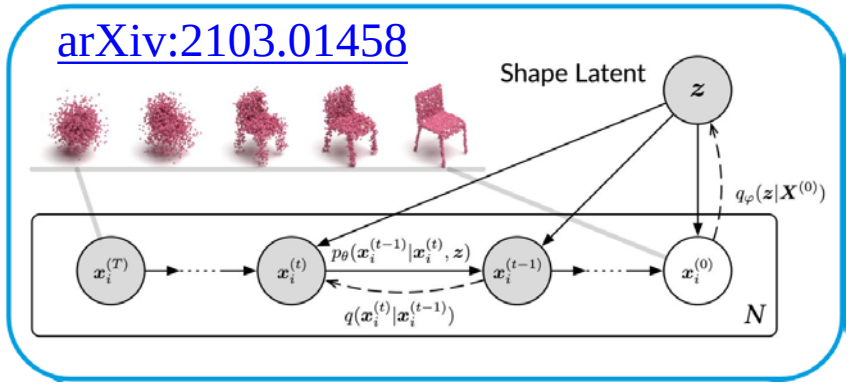
#54

- CaloDiffusion: denoising diffusion on CaloChallenge datasets
  - Use cylindrical convolutions, learned embedding for irregular geometries
- Classifier AUC 0.6–0.7
  - FPD 0.035–0.275, KPD 0.0001–0.006

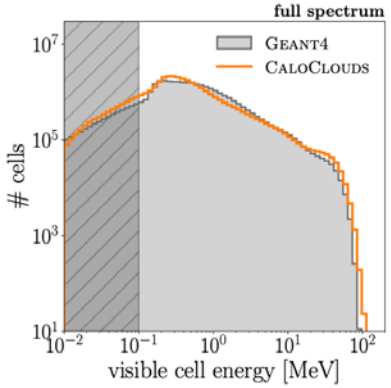


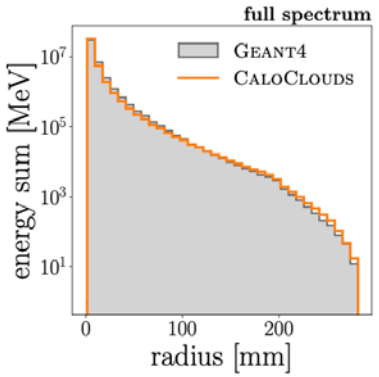
arXiv:2103.01458

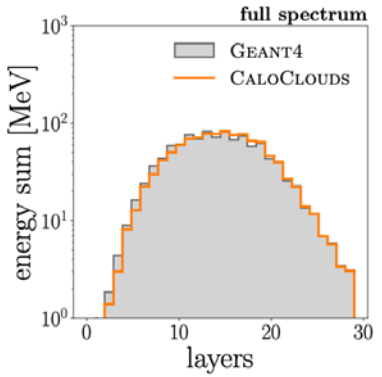
#538



- CaloCloud: point cloud diffusion for ILD detector
  - Project Geant4 energy steps into highly granular grid, smear to dequantize
  - PointWise Net w/ EPiC encoder and multiple NFs for inference
- Reasonable agreement in photon showers





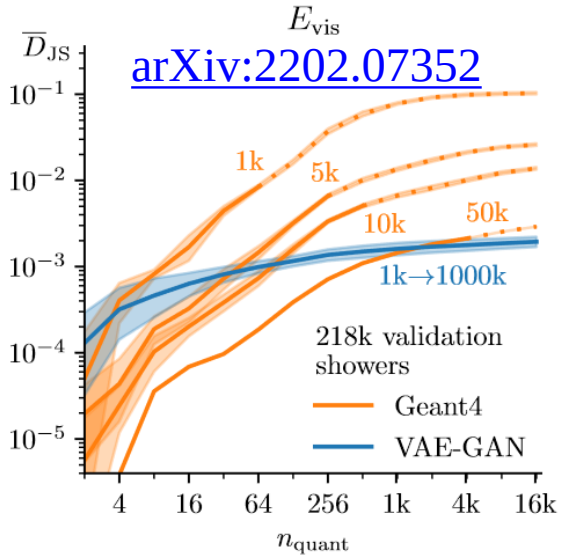


Costs & Scaling

Model	# params	# inputs	Time	Cost
StyleGAN3	~20M	~25M	~100 V100 GPU-days	$2.4K
DALL·E 2	~3.5B	~500M	~23 V100 GPU-years	>$300K
DreamStudio	890M	2.3B	~17 A100 GPU-years	$600K

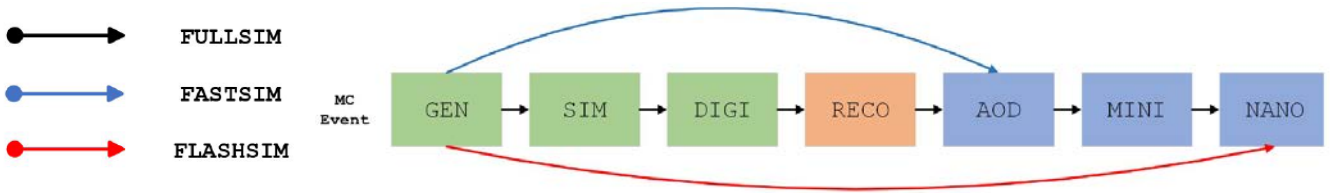
- Comparing "last generation" state of the art to "new generation" (diffusion-based)
  - Caveat: numbers obtained from the Internet
- This table just considers time to train "final" version of algorithm
  - "Exploration" stage can be orders of magnitude higher
    - Estimated 92 GPU-years for StyleGAN3, 115 GPU-years for DALL·E 2

- Case study: CMS Run 2 MC campaign
  - ~100 billion events
  - Scale up to HL-LHC (×30): 3 *trillion* events
- Need reliable generative ML *far beyond* scale of training dataset to be time- and cost-effective
  - Ability of a VAE-GAN architecture to "amplify" statistical power of input photon showers has been demonstrated
  - For real deployment, need to quantify exactly how much input data needed for any given campaign



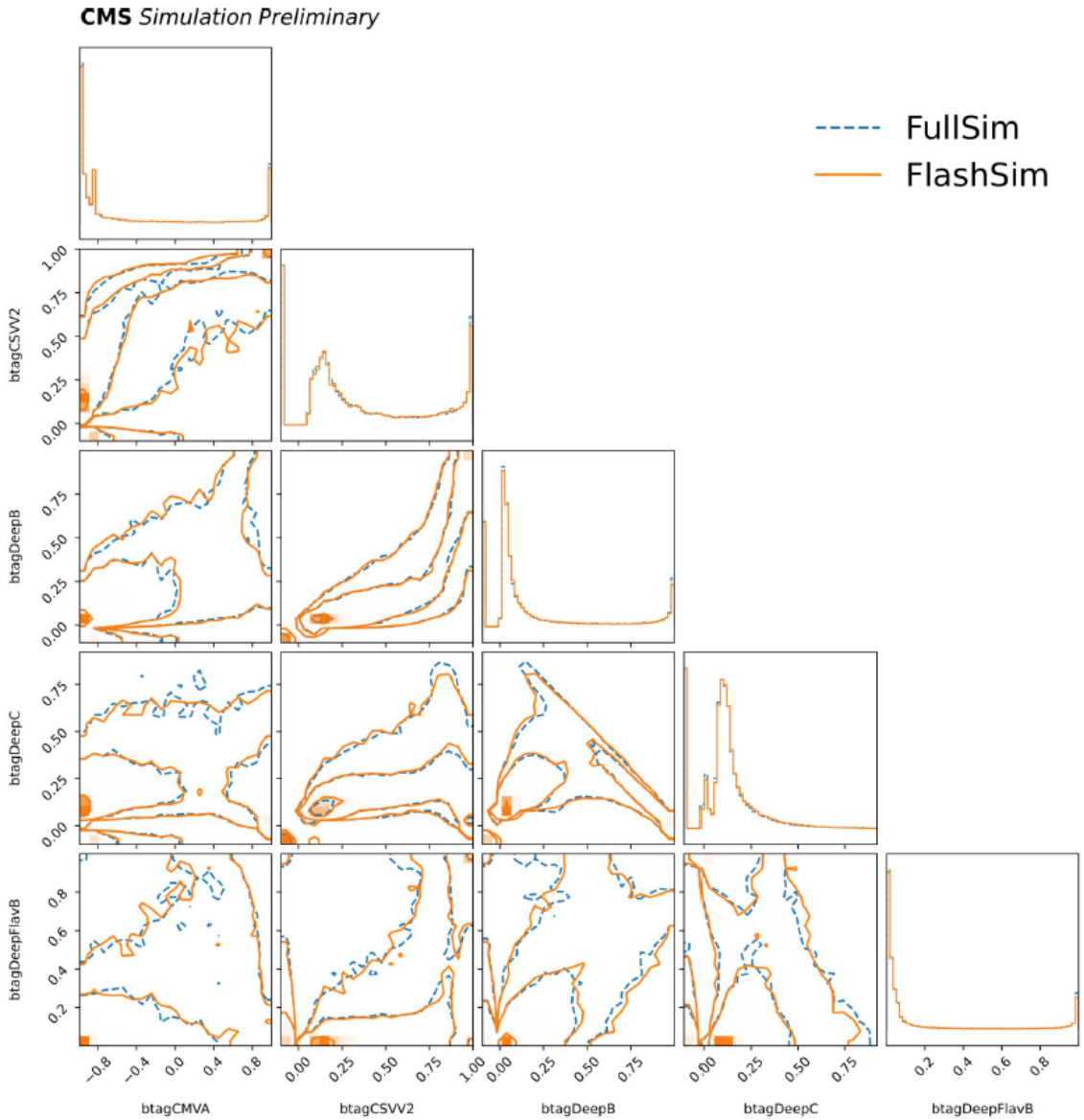
End-to-end: FlashSim for CMS

#476



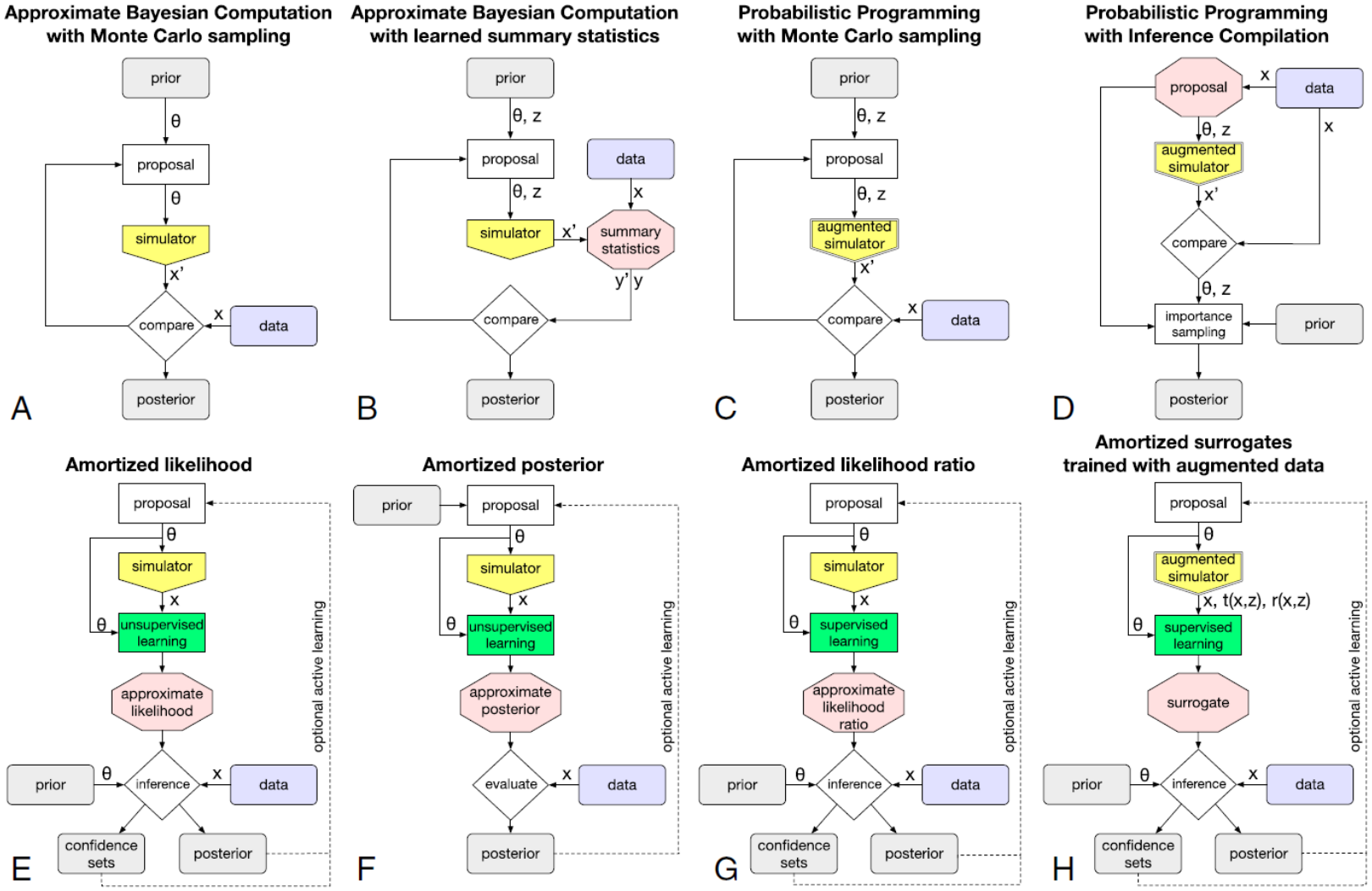
- Normalizing flow to predict high-level analysis quantities from generator-level information
- Reproduces correlations even in ML b-tagging algorithm scores
- Currently covers: jets (real & fake), muons, electrons

- Very promising solution for end-stage analyses
  - Effectively infinite MC → minimize statistical fluctuations
- Complementary w/ SIM-level solutions
  - Need to develop calibrations, algorithms, etc. to produce training data for FlashSim



Simulation-Based Inference

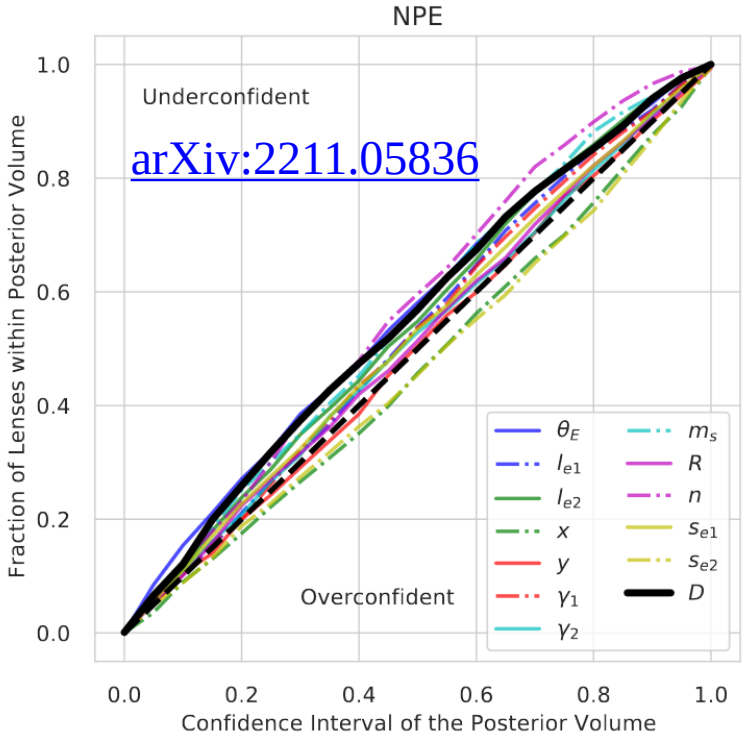
- FlashSim: produces high-level variables for comparison to data
- Only a small (?) step to directly produce likelihood and/or related quantities for inference…
  - Learning how underlying distributions depend on observables/parameters
  - Enable direct use of low-level, high-dimensional data
    - Rather than requiring dimensionality reduction into summary statistics

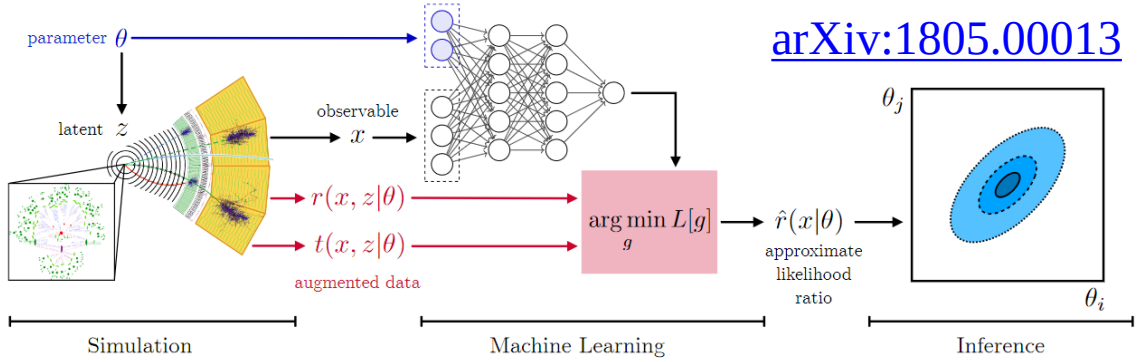


arXiv:1911.01429

SBI for Physics

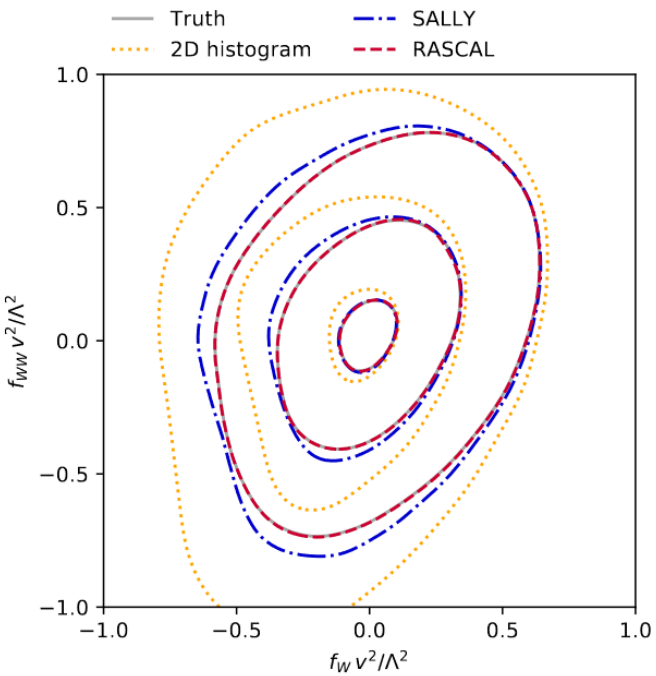
- Growing usage in astrophysics
  - And beyond! http://simulation-based-inference.org/
- Coverage for 12 lens parameters from DES-like dataset using Neural Parameter Estimation (normalizing flow)





arXiv:1805.00013

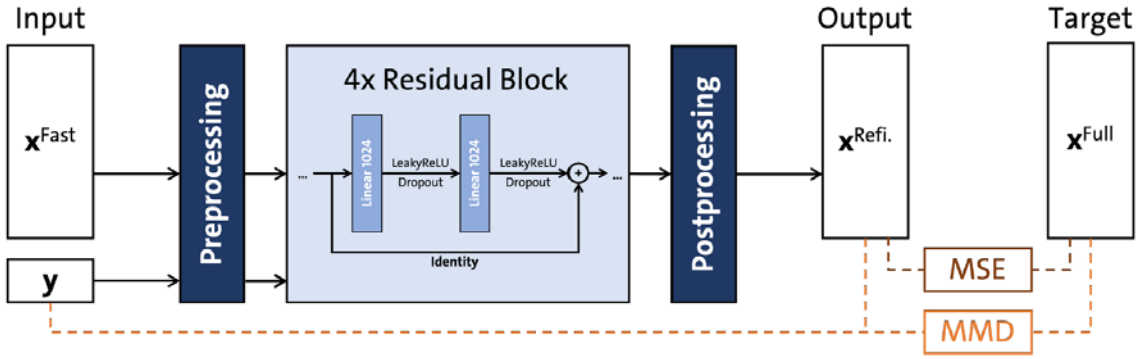
- RASCAL: Ratio And SCore Approximate Likelihood ratio
  - ML version of matrix element method
  - Learn estimator using joint likelihood ratio & joint score
- Matches true contours for SM EFT operators
  - Better agreement than other methods

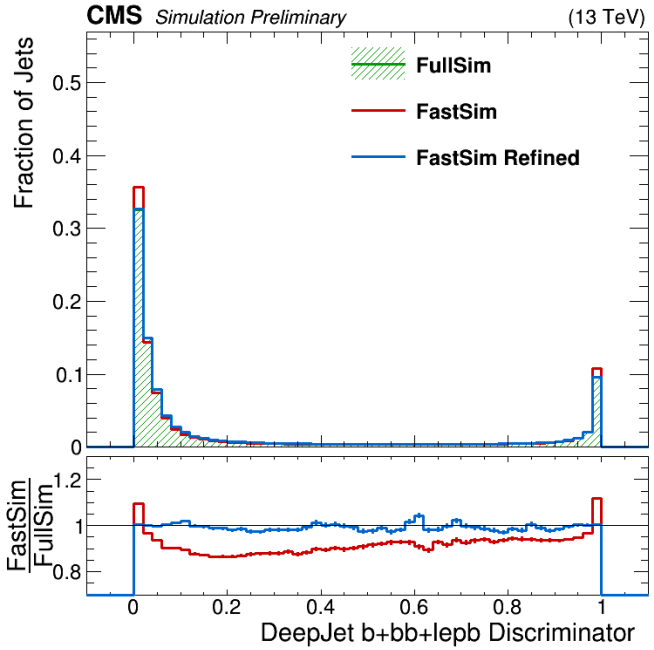


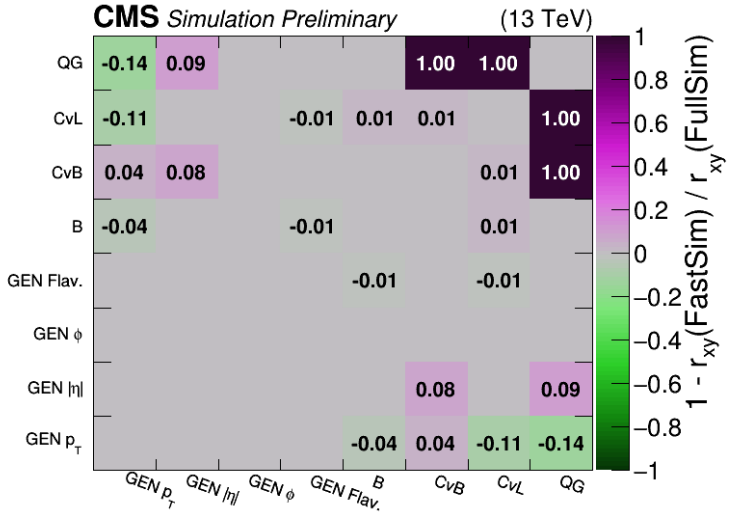
ML Refinement for CMS FastSim

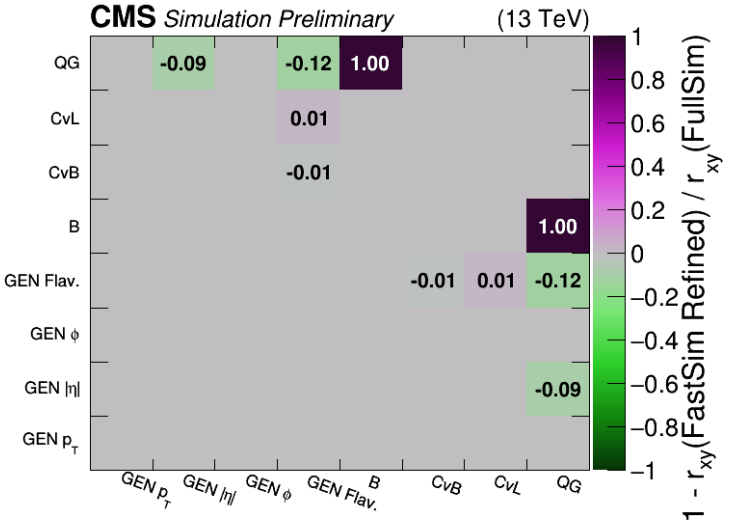
#83

- Alternate approach: ML adjusts high-level quantities from existing FastSim to match FullSim
  - Replaces coarse, manual correction factors
- ResNet-like architecture using skip connections
- Loss functions: both ensemble distribution and object-by-object comparisons
- Improves both 1D distributions and correlations





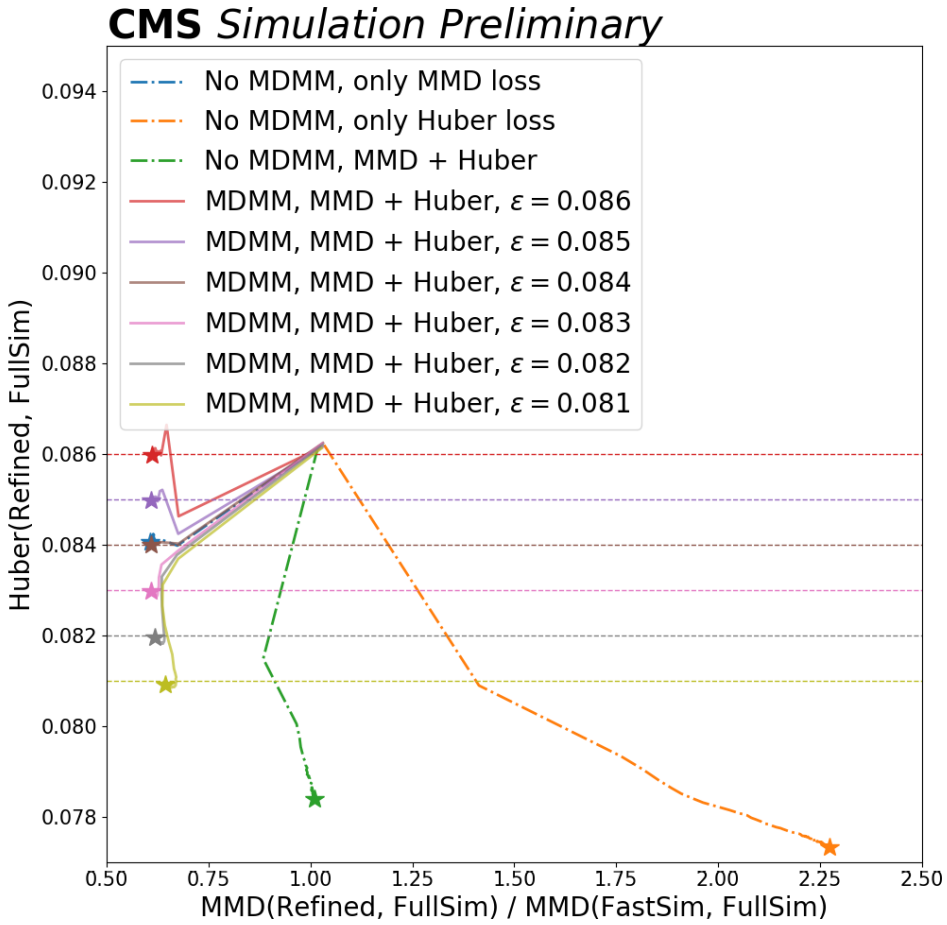




Learning More: Loss Multipliers

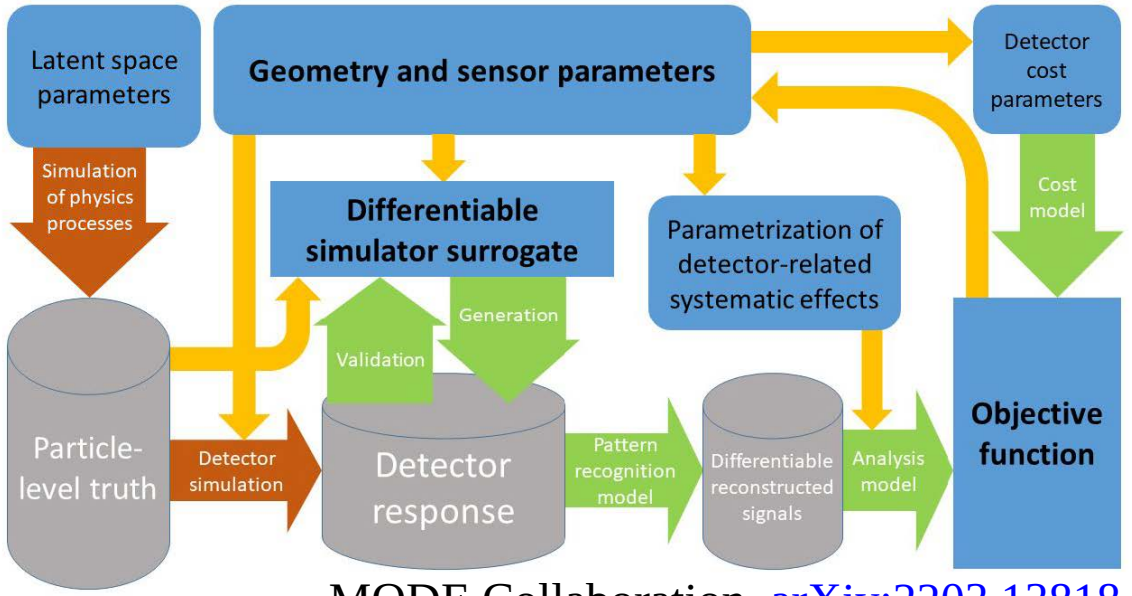
#83

- Increasingly common use of multiple terms in loss function to account for different effects
- Problem: in general, can't optimize for two things at once!
- Instead, optimize one function given a constraint on another function: $\mathcal{L} = f(\theta) - \lambda(\varepsilon - g(\theta))$
- *Learn* loss multipliers λ (instead of just guessing) using modified differential method of multipliers (MDMM)
  - Introduced at NeurIPS in 1987!
- Guarantees convergence on Pareto front
  - Whether convex or concave
  - Without this technique: accidental convergence *at best*
    - No control over *where* (combined) loss converges on convex Pareto front
    - Essentially random stopping on concave Pareto front



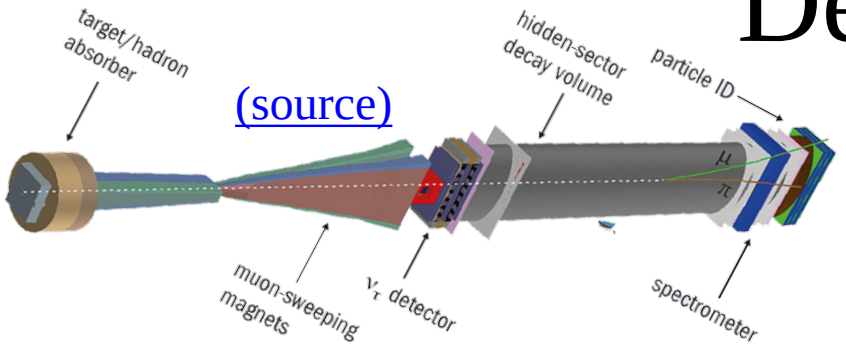
Differentiable Programming

- DNN training via gradient descent: enabled by *automatic differentiation*
  - Associate ∂f(x) with f(x) at machine precision by applying chain rule to basic instructions
- Training procedure can be extended to any differentiable function, given an objective for optimization
  - "Differentiable" is doing a lot of work here! Many functions do not have well-behaved derivatives
  - Alternative to differentiating a difficult function: train a DNN surrogate → ML for simulation
    - Convert discrete quantities to continuous: e.g. mixture density networks, or reinforcement learning
- Obvious utility for designing new experiments
  - Many choices of quantities to optimize!
    - Radiation hardness, physical resolution, cost, signal sensitivity…
  - May need to tune reconstruction algorithms as well as detector properties (geometry, materials, etc.)
    - Can extend even further with differentiable matrix elements, summary statistics, etc.
      - Synergy w/ simulation-based inference



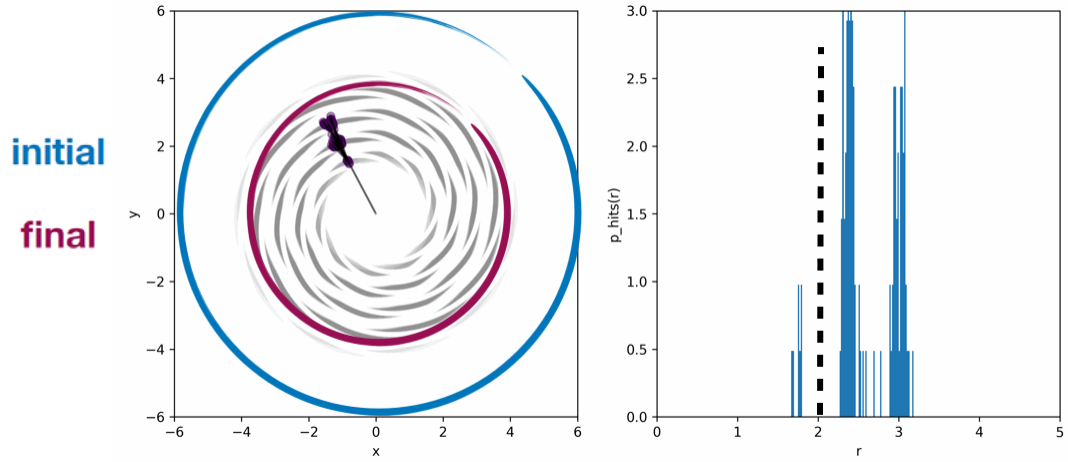
MODE Collaboration, arXiv:2203.13818
(Machine-learned Optimized Design of Experiments)

Detector Optimization

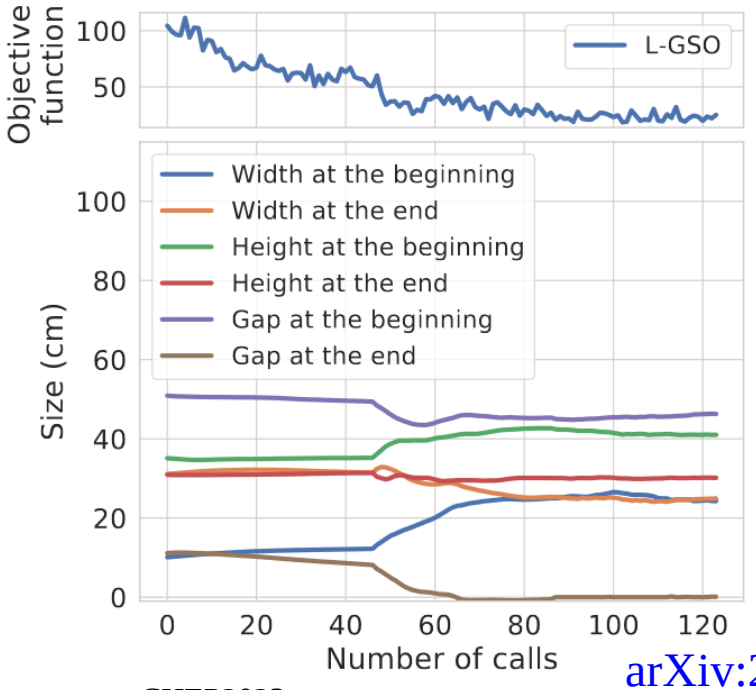


[(source)]

SHiP (Search for Hidden Particles) experiment



- **L-GSO** (local generative surrogate optimization)
  - Optimize multistage magnet parameters to minimize muons hitting detector

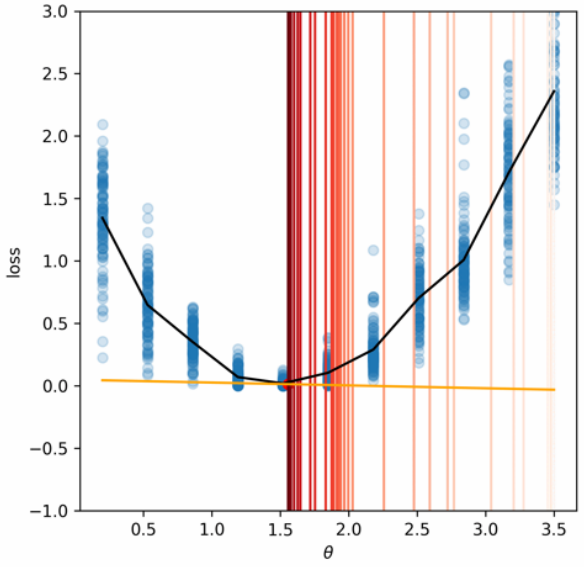


  - Outperforms Bayesian optimization: 25% lower objective w/ smaller magnets & similar computation
  - Can work w/ different surrogate methods

[arXiv:2002.04632]

- Simplified particle detector: modify radial distance of material to achieve mean hit radius = 2
- Successful even w/ noisy gradients
- Multiple new methods for gradient estimation in stochastic problems

[#542]

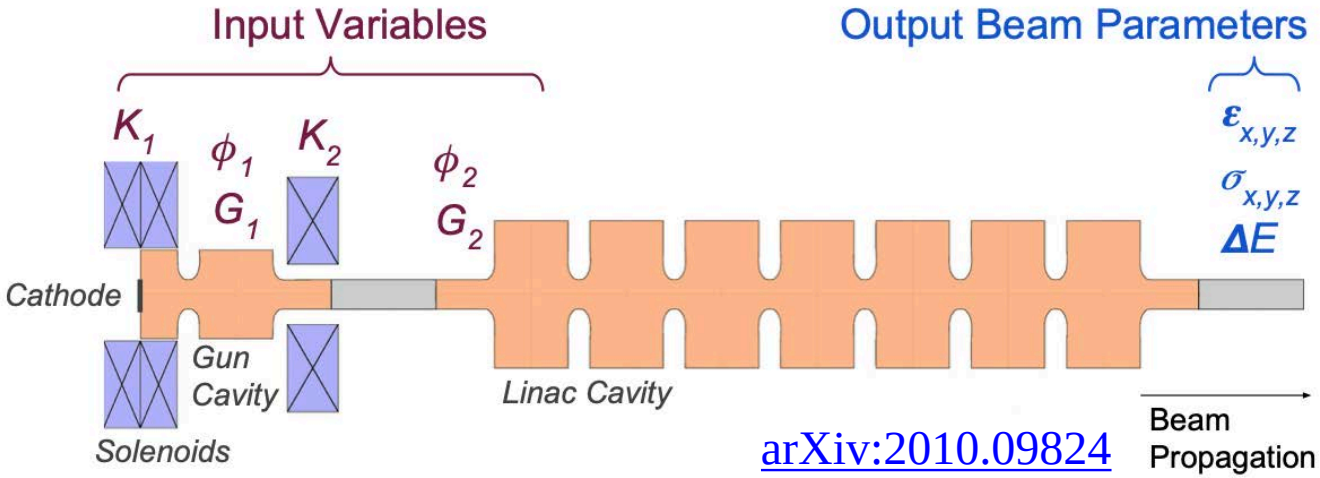


Loss Function

Gradient Estimate

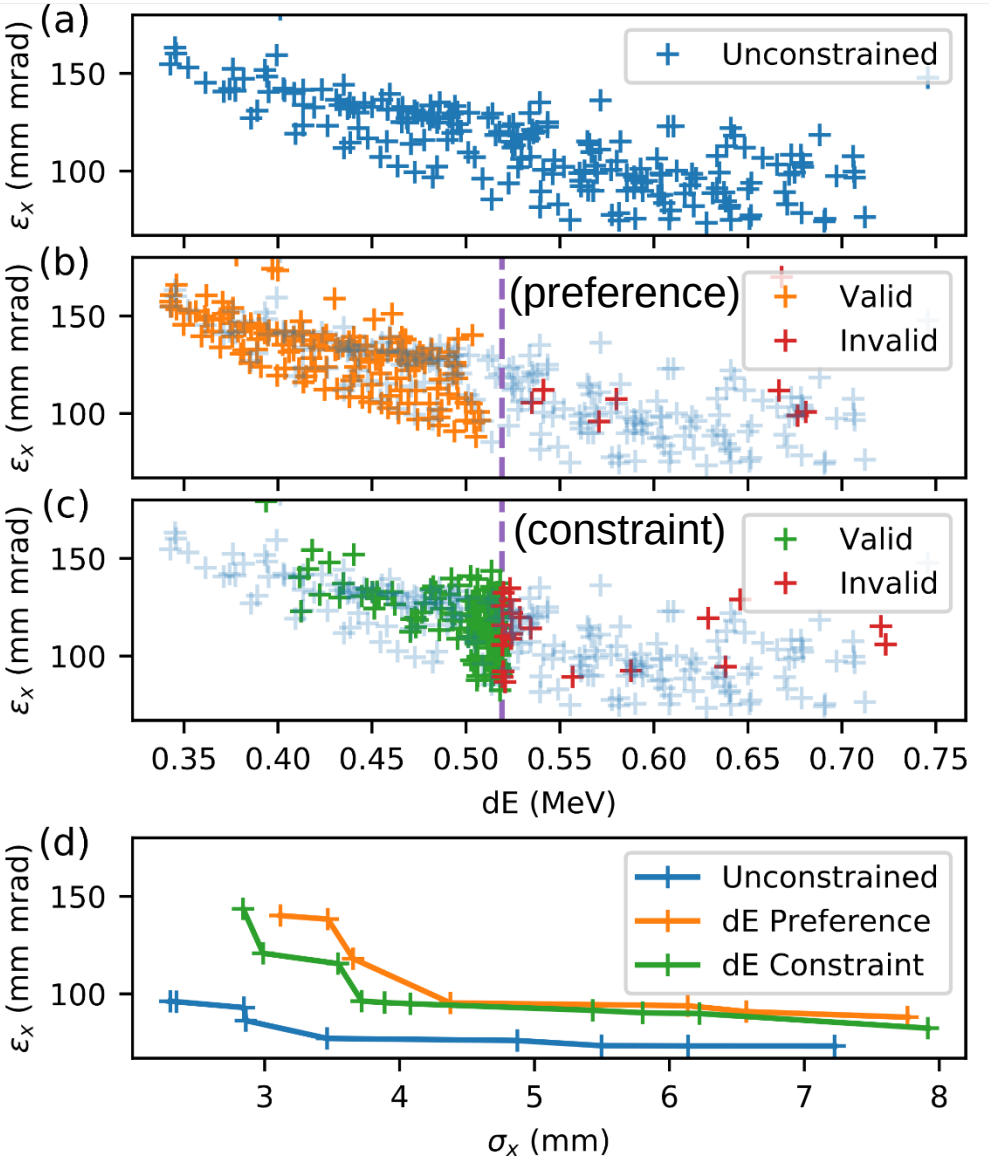
Evaluated Parameters

Accelerator Optimization



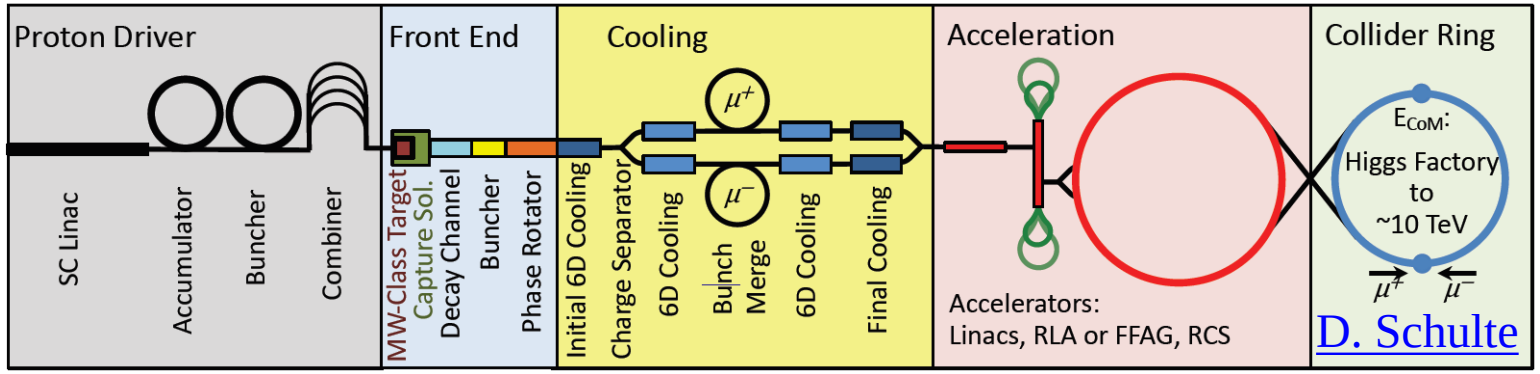
arXiv:2010.09824

- Multi-Objective Bayesian Optimization (**MOBO**) for accelerator performance
  - Based on Argonne Wakefield Accelerator
  - Minimize beam emittance, energy spread, etc. by optimizing injector & LINAC parameters
    - Uses Gaussian Process surrogate for each objective
- Compares different constraint methods
  - Finds Pareto front in each case: better to use MDMM-like constraint method



Optimizing the Energy Frontier: Muon Collider

- Muon collider will reach 10 TeV parton energy scale in 16 km ring (vs. 100 km for hadron collider)
- However, more complex machine: several components need to interface smoothly & efficiently
  - + novel operations like *beam wiggling* to mitigate radiation from TeV neutrino interactions

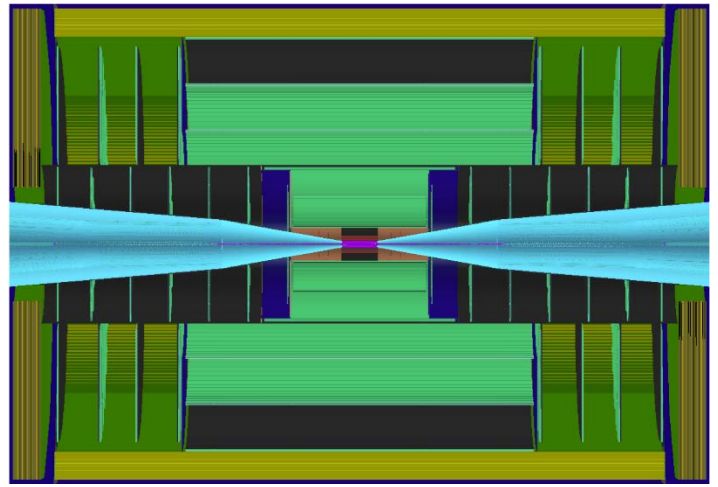


➢ Ideal case for differentiable optimization!

- Also, machine-detector interface very important: *nozzle* to mitigate beam-induced backgrounds from muon decays in flight

❖ Consider simultaneous optimization of accelerator *and* detector

  - A new frontier for simulation and design

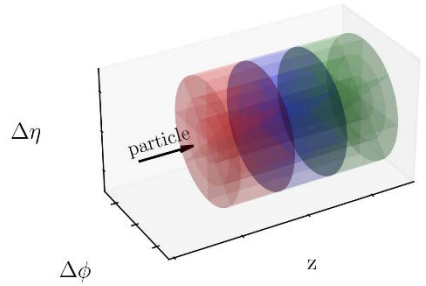


arXiv:2203.07964

A Word From Our Sponsors

- CHEP is chock full of great talks and new ideas about many topics!
  - But if you want even more ML for simulation and differentiable programming…

CaloChallenge workshop:
May 30–31,
Villa Mondragone, Italy



MODE workshop:
July 24–26,
Princeton, USA

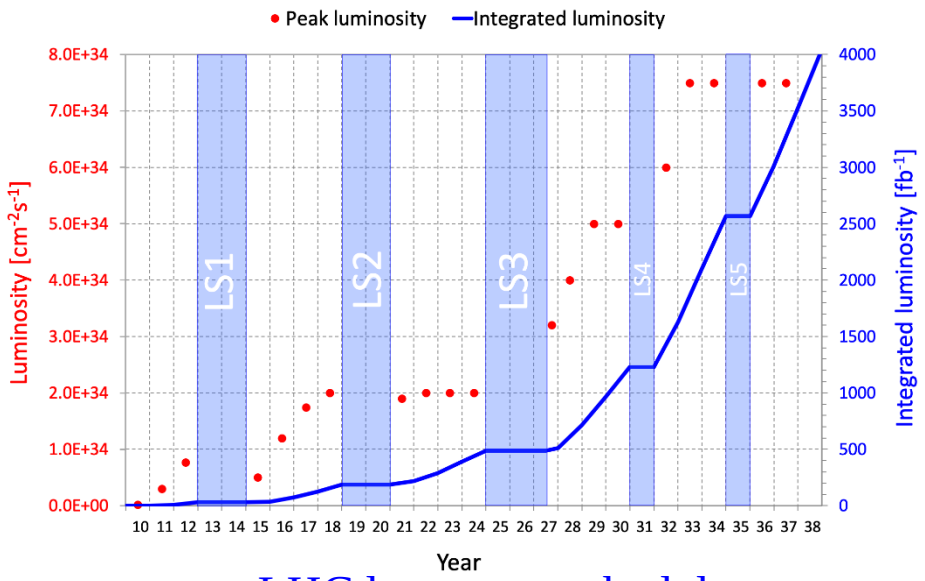
ML4Jets:
November 6–10,
DESY, Germany

Conclusion

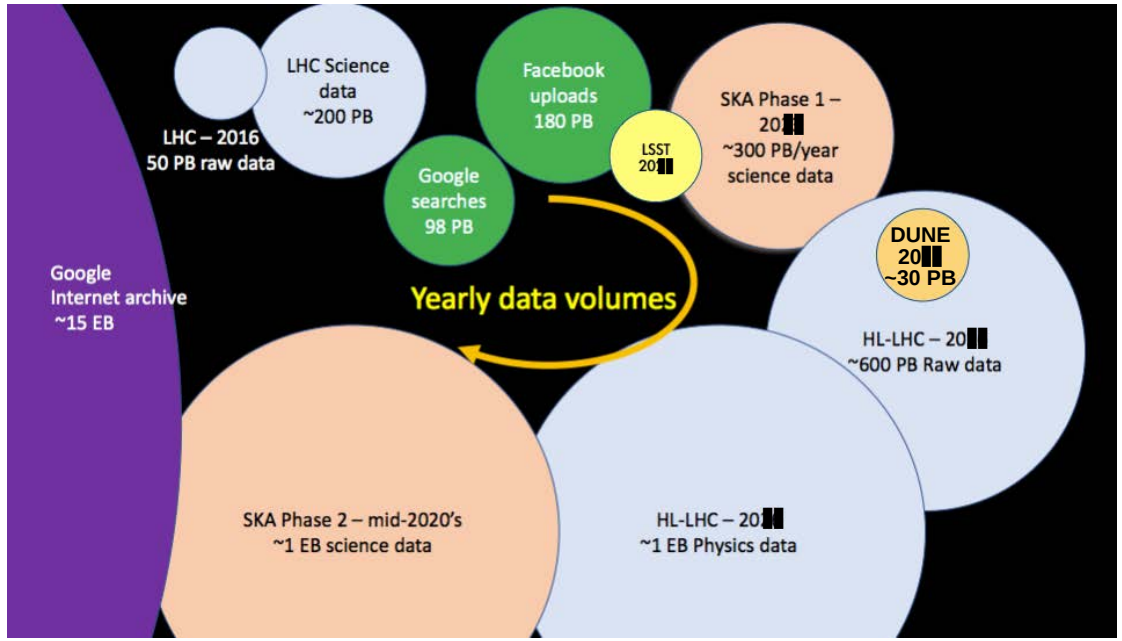
- Simulation is *crucial* and faces severe computing challenges
- AI/ML can simulate events “from scratch” (fully generative) or refine existing fast simulations
  - Full simulation still *essential* as source of truth for training
- Numerous architectures have been explored, with promising new developments ongoing
  - GANs, VAEs, NFs, transformers, diffusion, autoregression…
- Some experiments already *deploying* generative ML in production (ATLAS, LHCb)
- Need to think about *metrics* & *training costs* to assess viability of ML methods
- End-to-end option starts to approach simulation-based inference
- Going *beyond* AI/ML, differentiable programming offers new insights into experiment design
- *Combined* optimization of accelerator & detector design seems very promising; maybe a new frontier!
- Many, many CHEP contributions on these & related topics
  - Click the boxes to learn more about those highlighted here ###
  - Check out parallel & poster sessions for even more!

Backup

Challenges



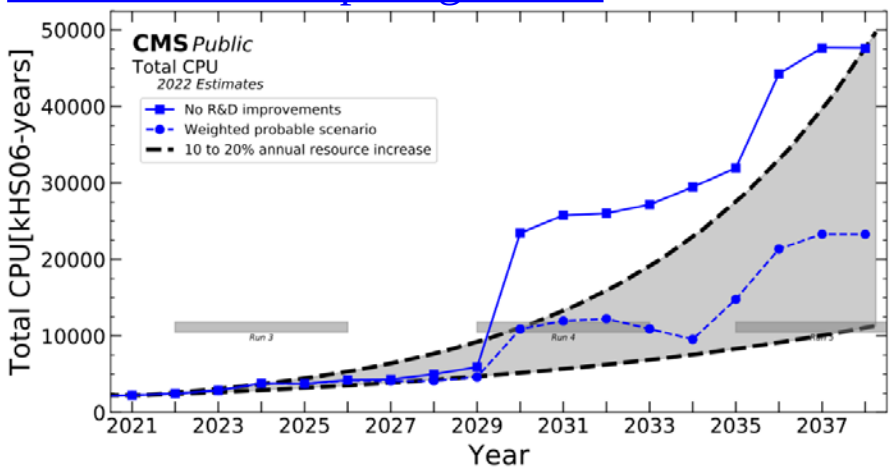
[LHC long term schedule]



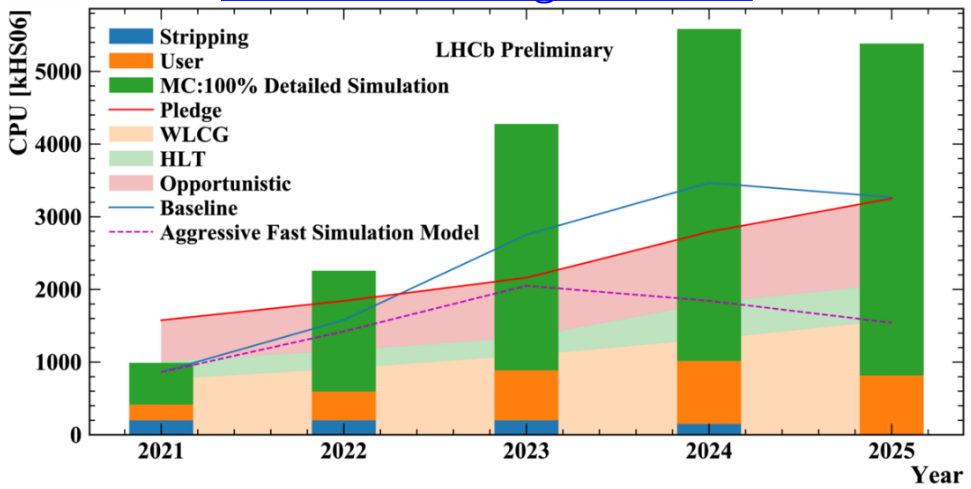
- A new precision era is imminent: HL-LHC, DUNE, LSST, SKA
  - 10× or more data compared to existing experiments
- **Reconstruction** in dense, highly-instrumented environments will need increasing fraction of computing
  - e.g. superlinear scaling with simultaneous collisions (pileup) at HL-LHC (up to 200)
- **Simulation** needs to deliver more events, with more complexity, to match growing data volumes
  - …while using smaller fraction of computing!

Projections

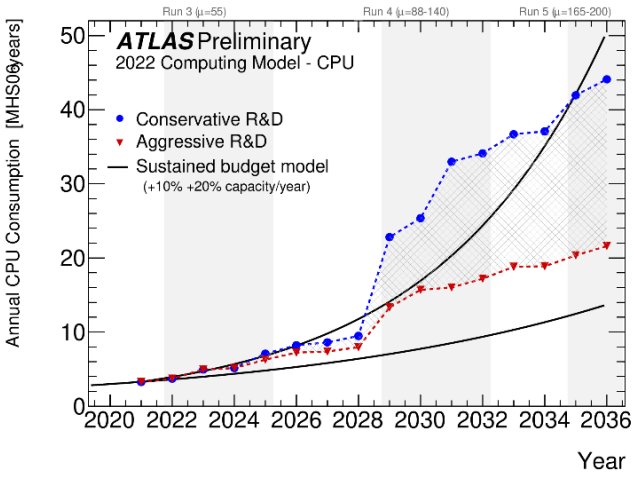
[CMSOfflineComputingResults]

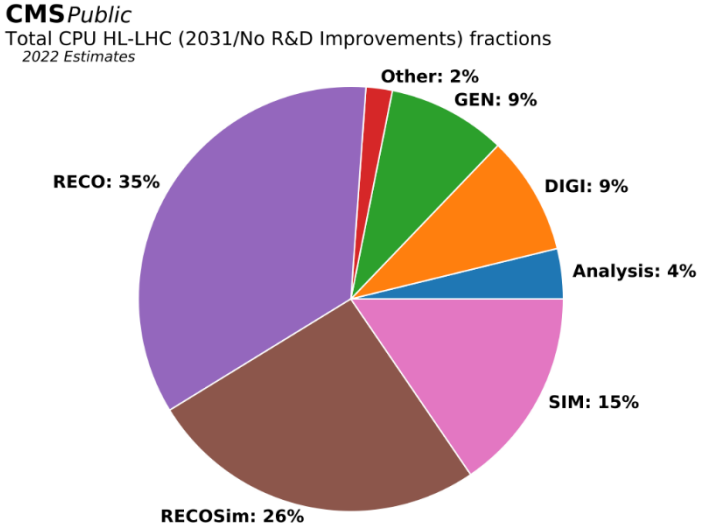


[LHCb CPU Usage Forecast]

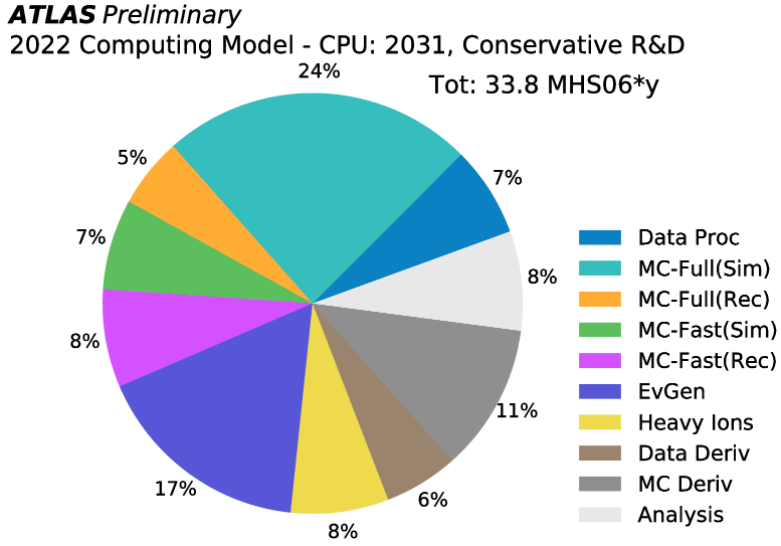


[CERN-LHCC-2022-005]

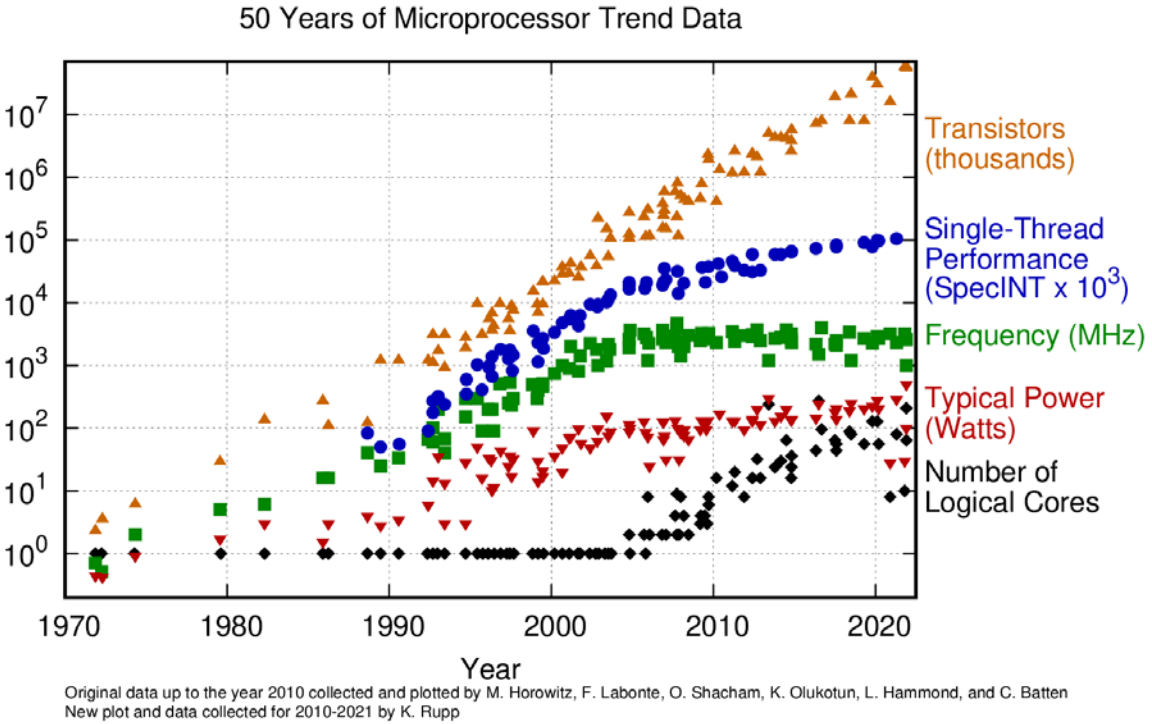




- Run 2: (full) simulation used ~40% (plurality) of grid computing resources for CMS & ATLAS [arXiv:1803.04165]
  - 70% for LHCb! [LHCb-PUB-2022-010]
- Run 4+: limit to ~10–20%, while simulating:
  - Complex detector upgrades
    - e.g. CMS High Granularity Calorimeter
  - More precise physics models
  - More events to reduce statistical uncertainty



Processors: Old and New



- **CPUs**: Moore's Law continues, but Dennard scaling has broken down → stagnant performance/thread

➢ Heterogeneous **revolution**: rise of *specialized coprocessors* attached to general-purpose CPUs

  - **GPUs** (SIMD), **FPGAs** (spatial computing), **ASICs**
  - *Growing taxonomy*: even more specialized processors emerging, e.g. **IPUs** (MIMD for ML)

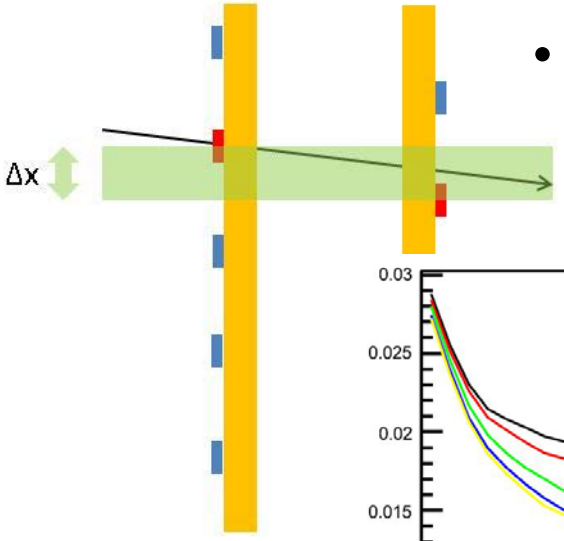
- *Deep learning* uses limited set of mathematical operations: perfect for acceleration on GPUs etc.
  - *Inference as a service*: most general/abstract way to offload tasks to coprocessors

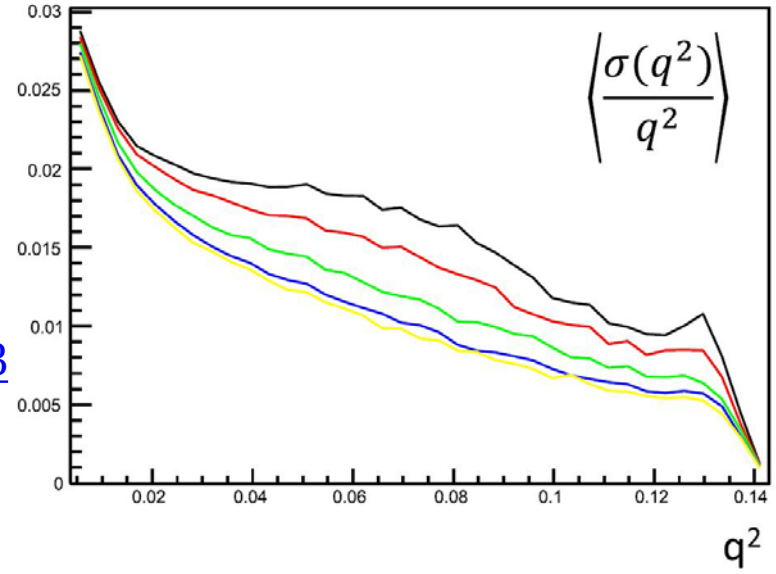
#247

See also: efforts to port FullSim engines to GPUs (Opticks, Celeritas, AdePT)

More Detector Optimization

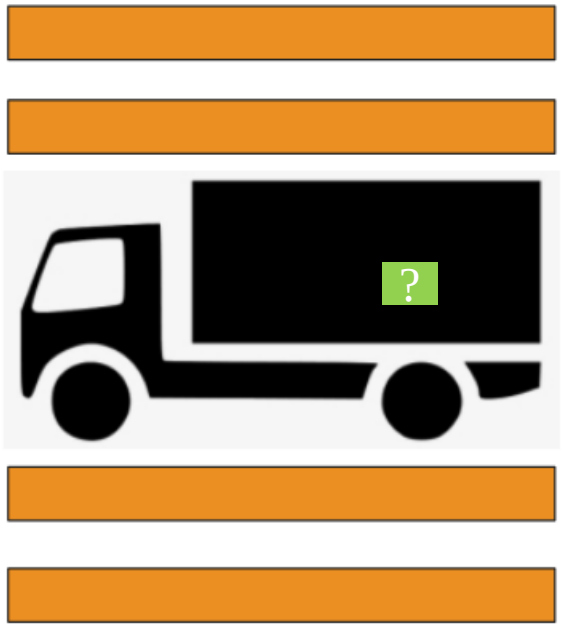
- Proof of concept: grid search (brute force approach) for MUonE detector
  - Measurement of muon-electron elastic scattering vs. q^2 (relevant to muon g–2)
    - Improve resolution by optimizing strip sensor staggering interval

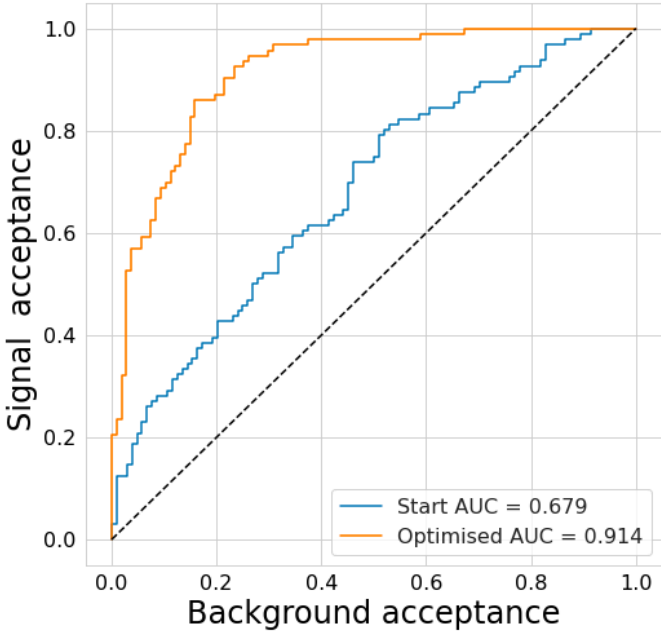




arXiv:2002.09973

- **TomOpt**: optimizing muon tomography detectors
- Example: finding uranium hidden in a truck
  - Minimize classification error (and cost) by adjusting panel positions and sizes
- Result: significant improvement in classification





[(source)](source)