MACHINE LEARNING FOR PARTICLE PHYSICS SIMULATIONS

Raghav Kansal
UCSD, Fermilab

MODE Workshop
24th July 2023

LHC SIMULATIONS

Sources
S. Sekmen, LPC 2017
F. Krauss, Kyoto 2011

Hard process
Showering
Hadronization
Underlying event

→ GenParticles

Energy depositions and interactions along detector

→ SimHits

Digitization
Hit reconstruction

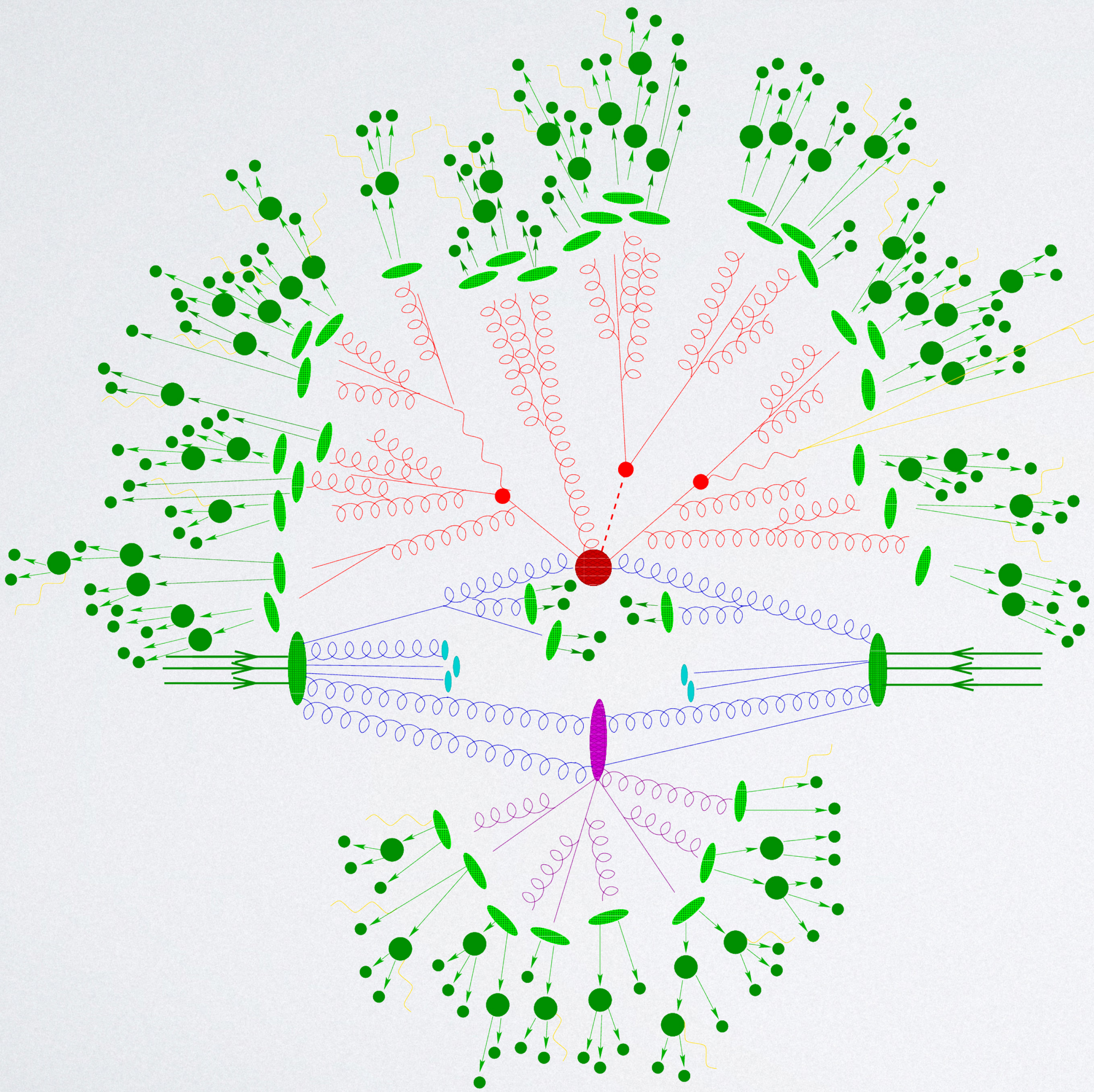
→ RecoHits

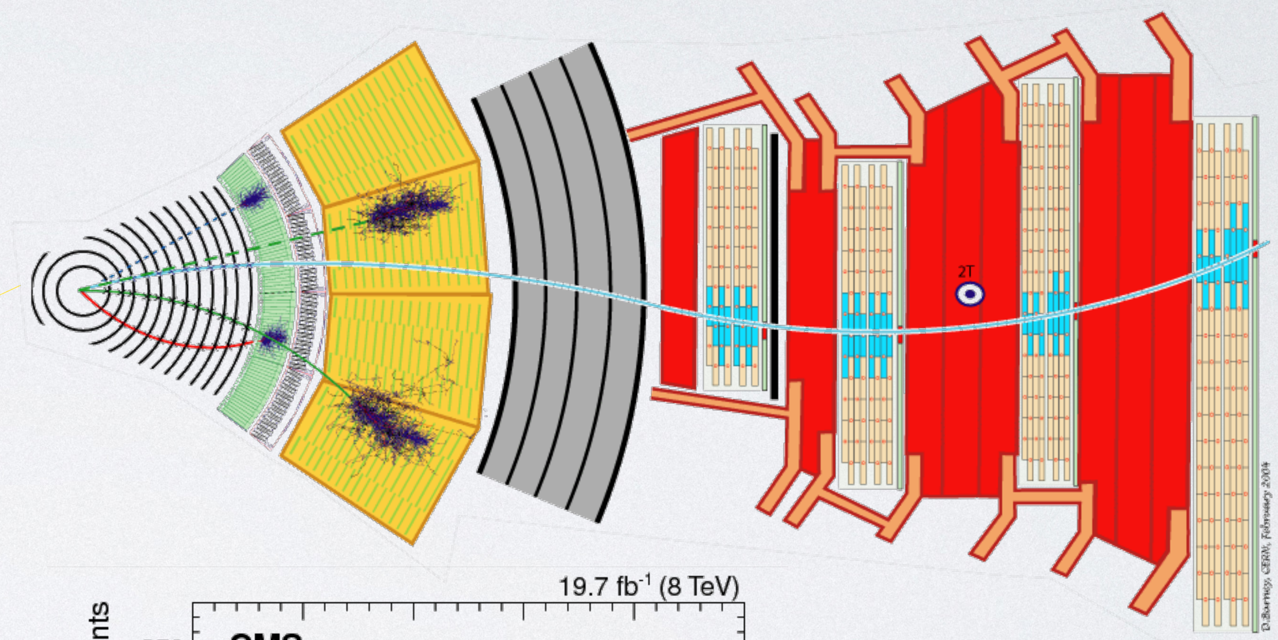
Particle reconstruction

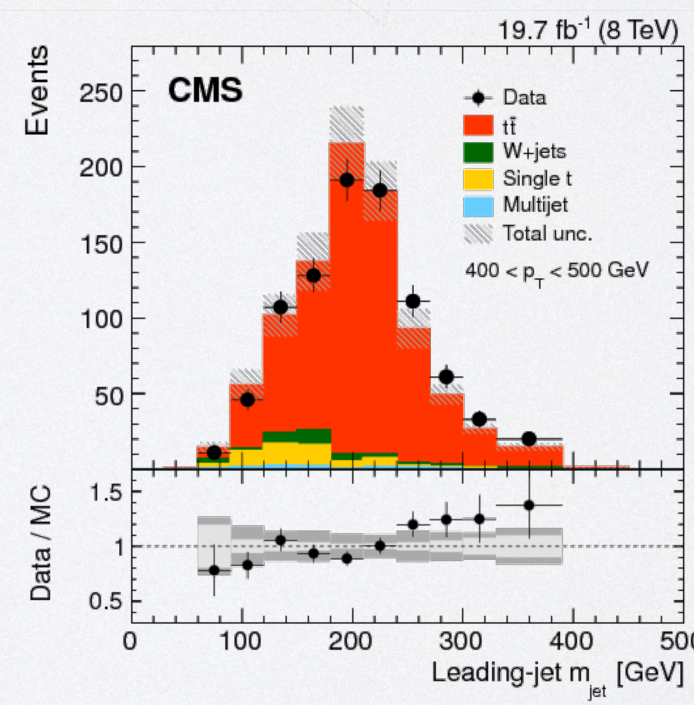
→ Reco Particles

Event-level analysis

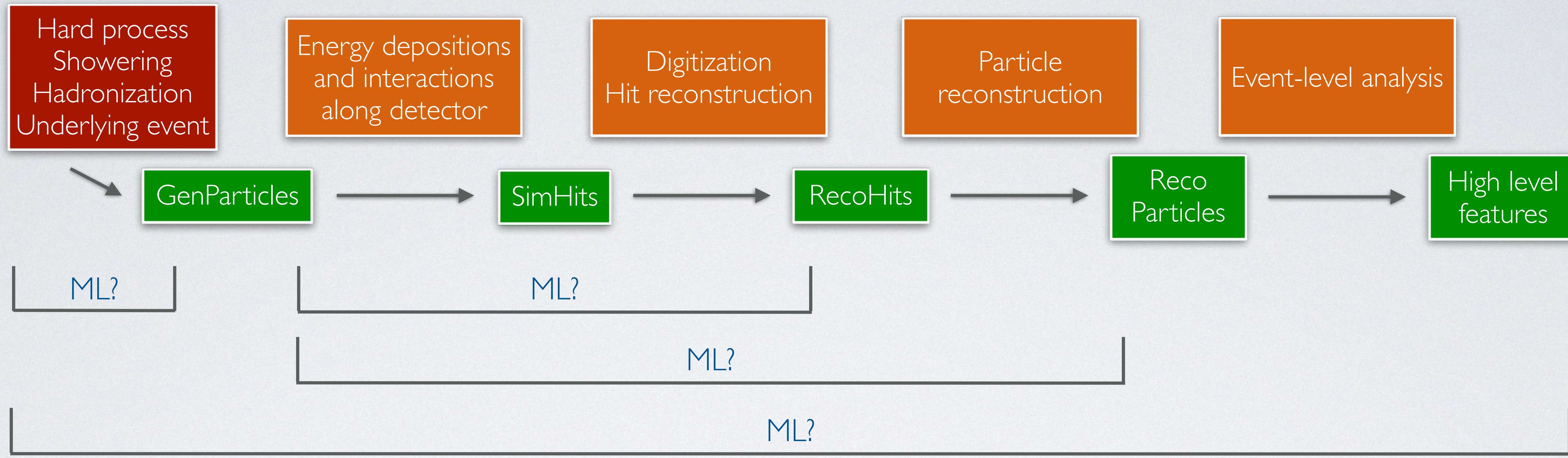
→ High level features



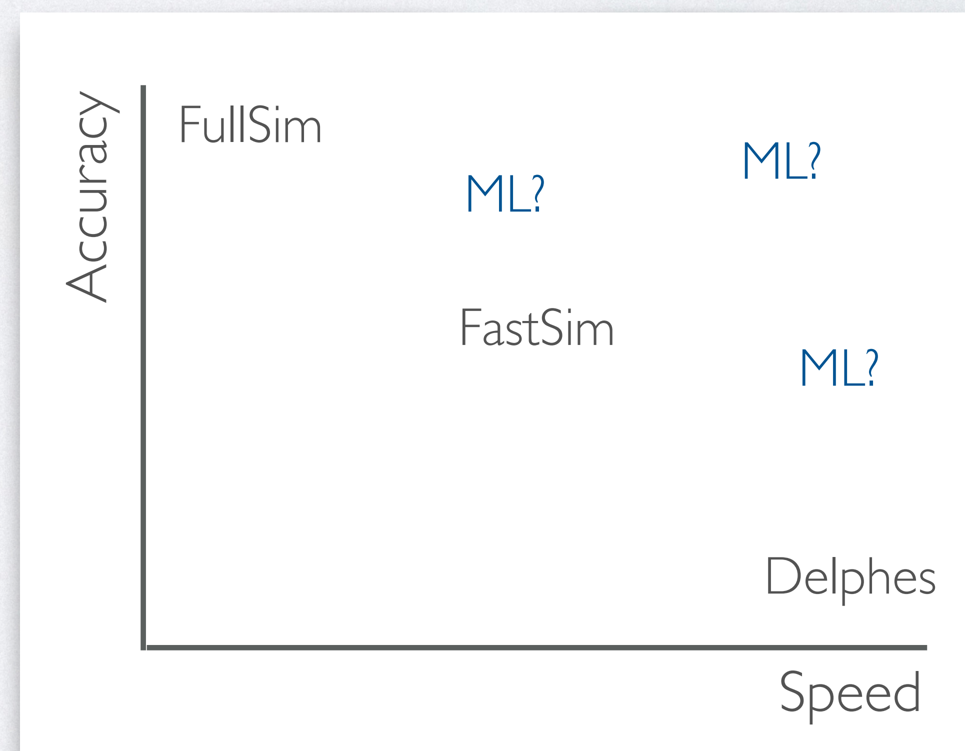




LHC SIMULATIONS

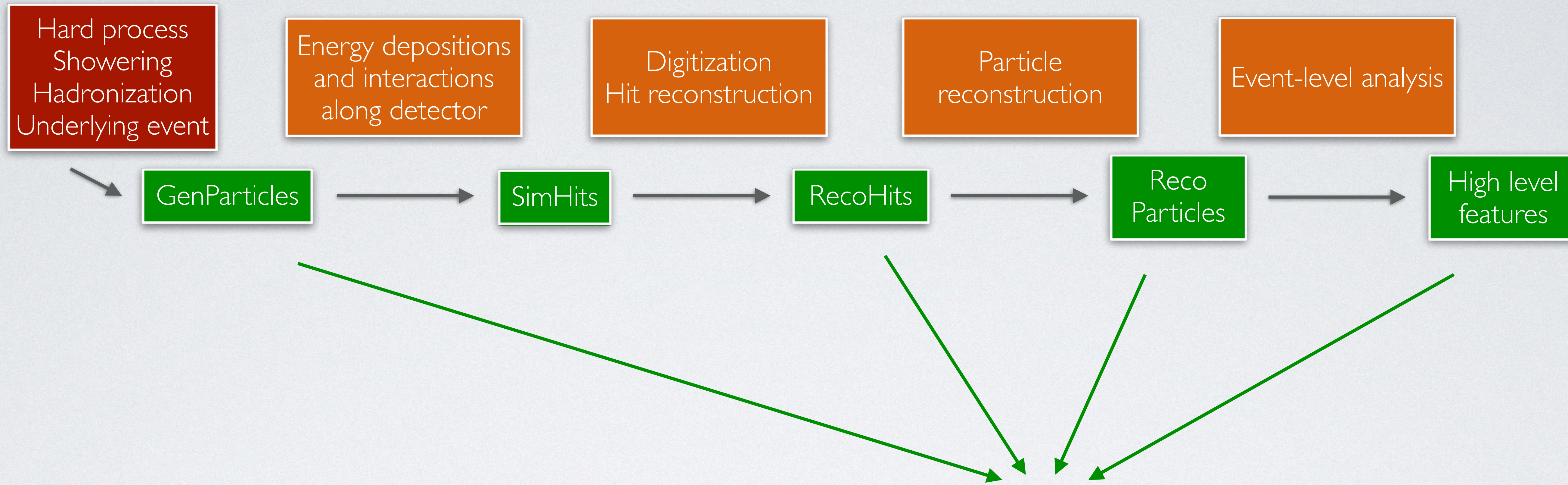


- Opportunity for ML alternatives in many steps
- Trading accuracy of FullSim (GEANT + reco) for speed and differentiability
- Trading interpretability/trust for # of steps



K. Pedro, HSF 2020

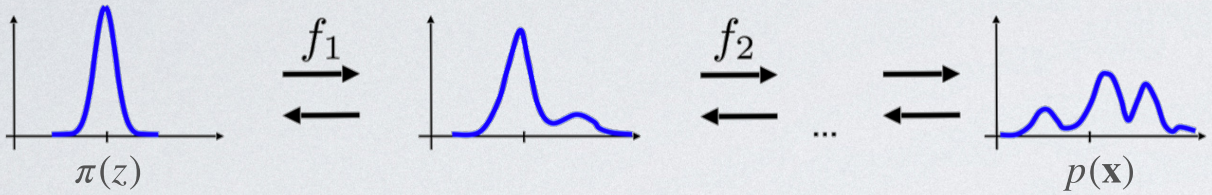
LHC SIMULATIONS



- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$
- Rich area in machine learning: **deep generative models**
  - $p_\theta(\mathbf{x})$ typically modelled using high-capacity DNNs

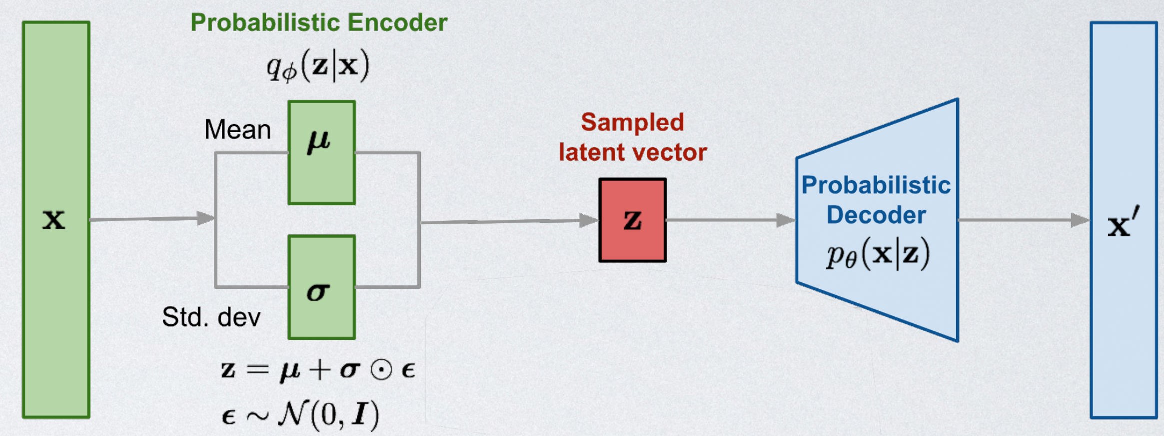
ML LANDSCAPE

Flows



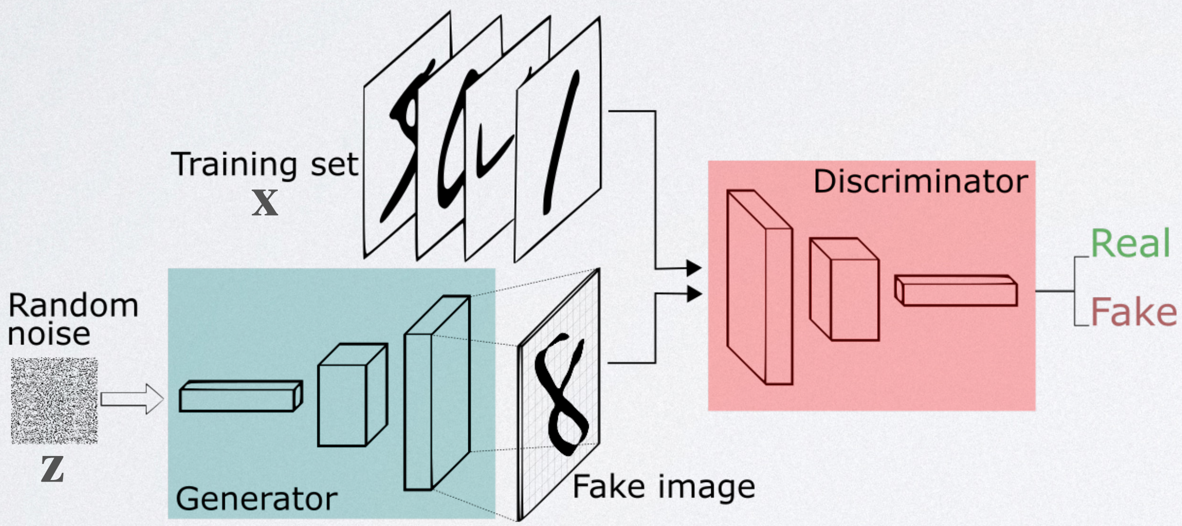
- Learn transformations from simpler $\pi(z)$ to $p(\mathbf{x})$
- Access to exact likelihood
- But restrictive

Variational Autoencoders (VAEs)



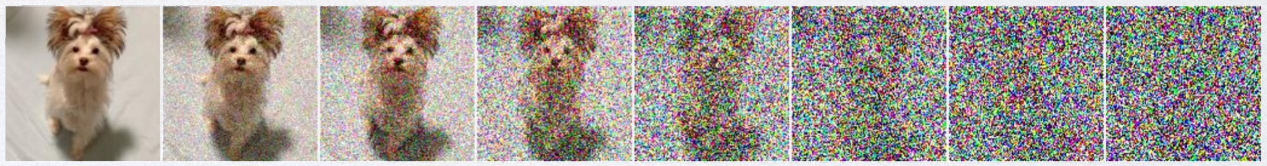
- Maximise an approximation to the likelihood
- Can also be restrictive

Generative Adversarial Networks (GANs)



- Minimise loss wrt to discriminator classifying real or fake
- Less restrictive, generally higher performing
- Difficult to train

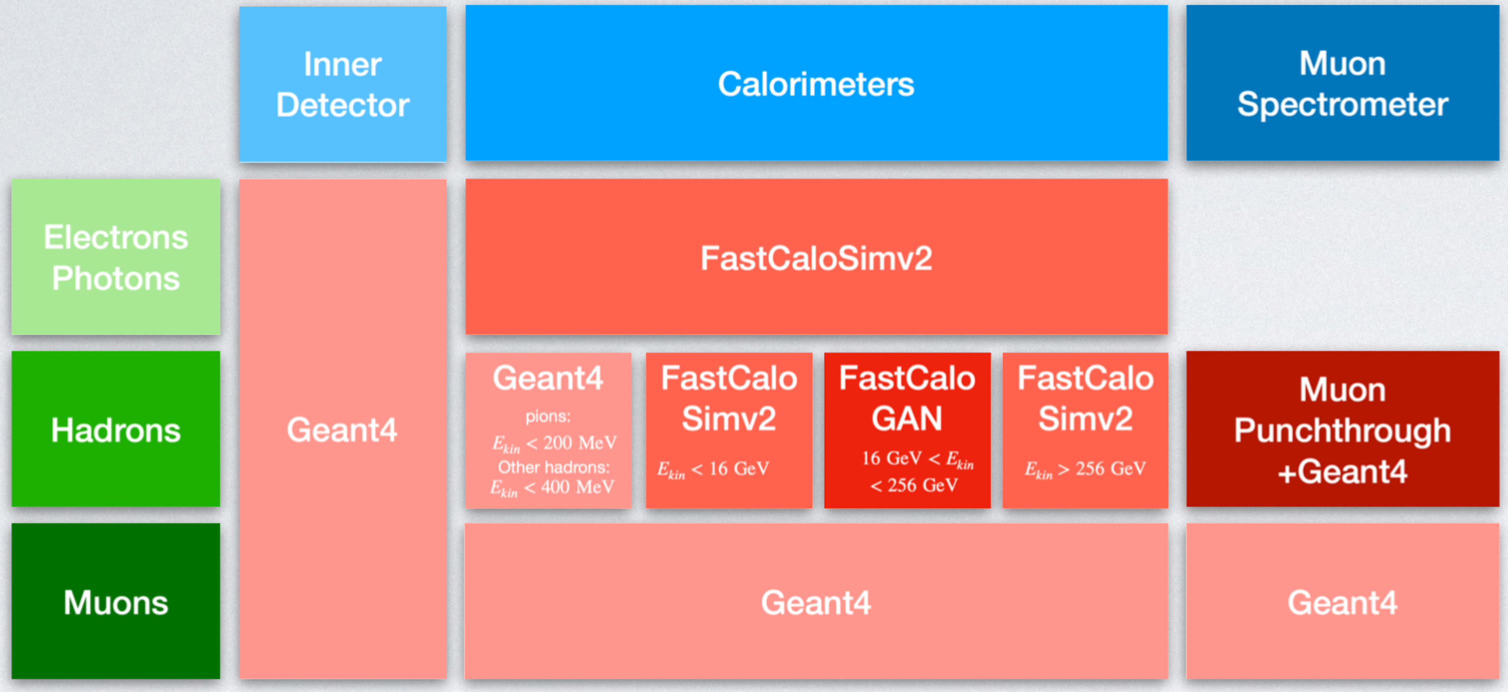
(Score-based) Diffusion

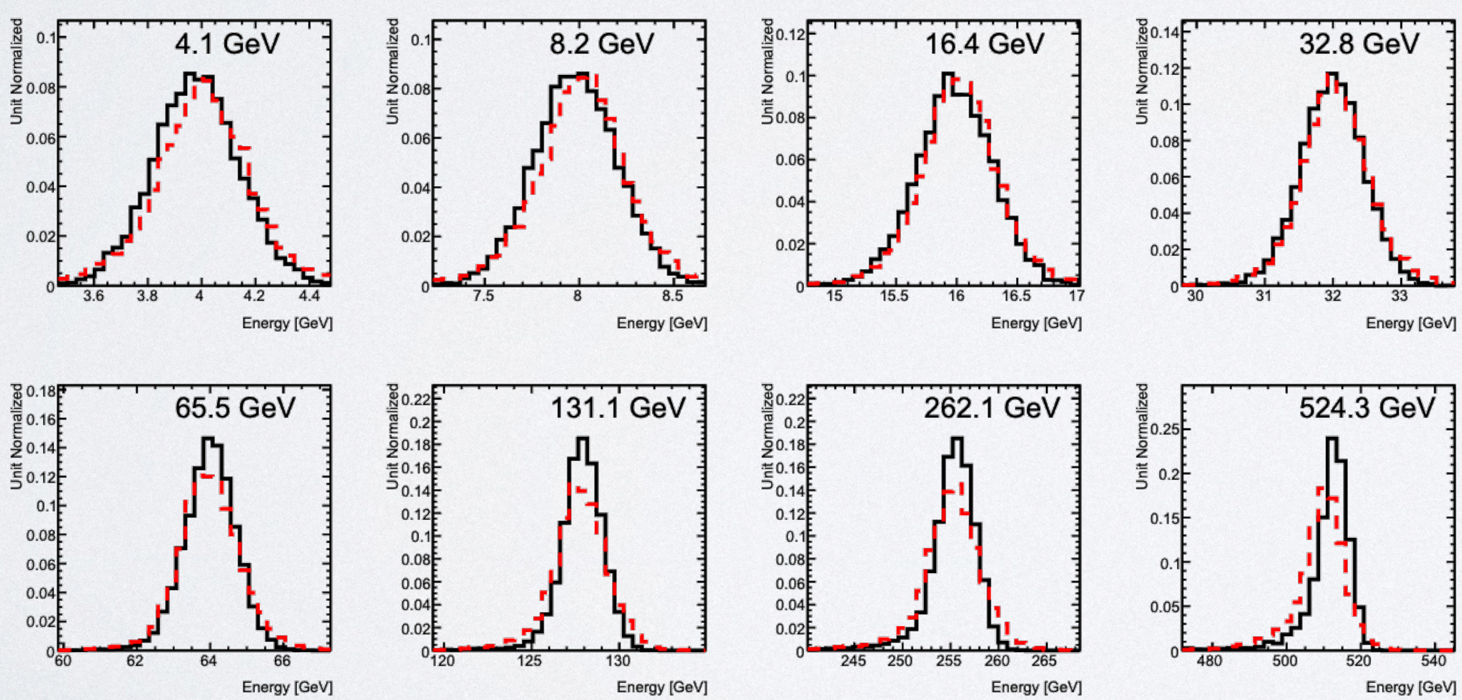


- Learn $-\partial \ln p(\mathbf{x}) / \partial \mathbf{x}$ (score) instead of $p(\mathbf{x})$ directly
- Less restrictive, score doesn't need to be normalised
- Current industry SOTA (DALL-E, StableDiffusion etc.)
- But slow - need O(100)s of steps along the score

APPROACHES

ATLAS FASTSIM



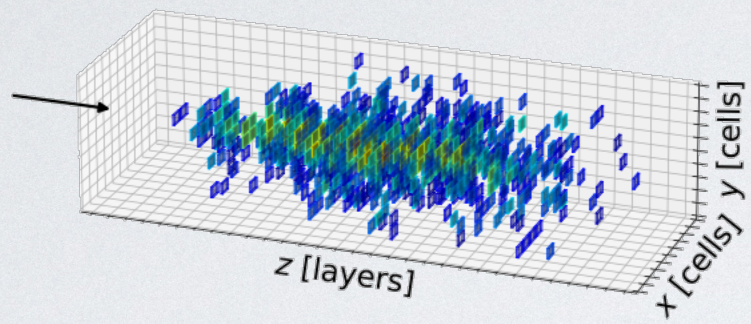


- GANs used already for fast sim
  - One component of "AtlFast3"
  - 7B events for Run 2 analyses!
- Trained on hadron shower images
- Reasonable performance but:
  - Room for improvement
  - "Voxelisation" to deal with sparsity and high granularity
  - 300 GANs trained for each E, η bin

CALORIMETER SHOWERS

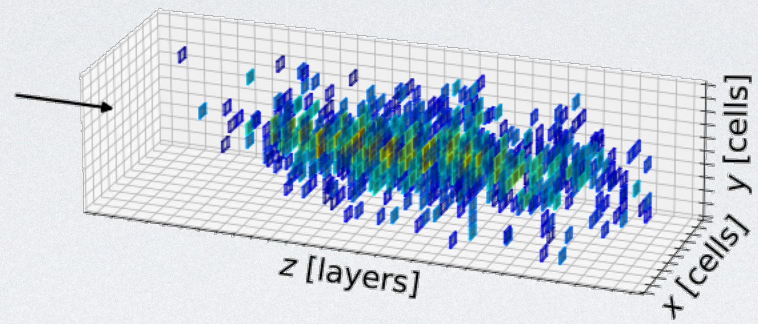
- Idea: learn distribution of hits per gen particle i.e., replacing GEANT

BIB-AE (2112.09709)



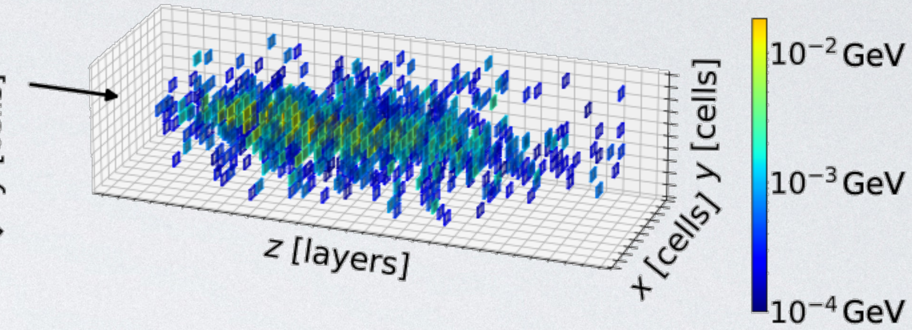
- VAE + GAN + postprocessing
- 0.1 μs/shower*

Sample GEANT shower



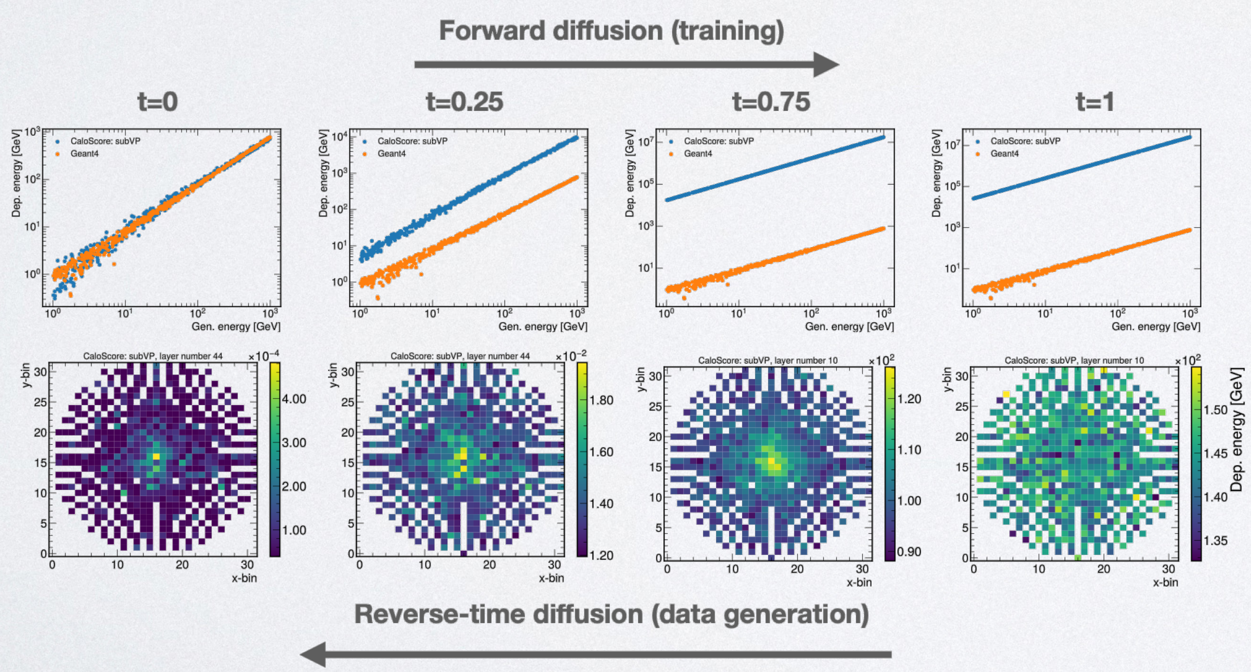
- Reproducing shower images
- Good agreement with simplified ILD-like single γ showers

L2LFlows (2302.11594)

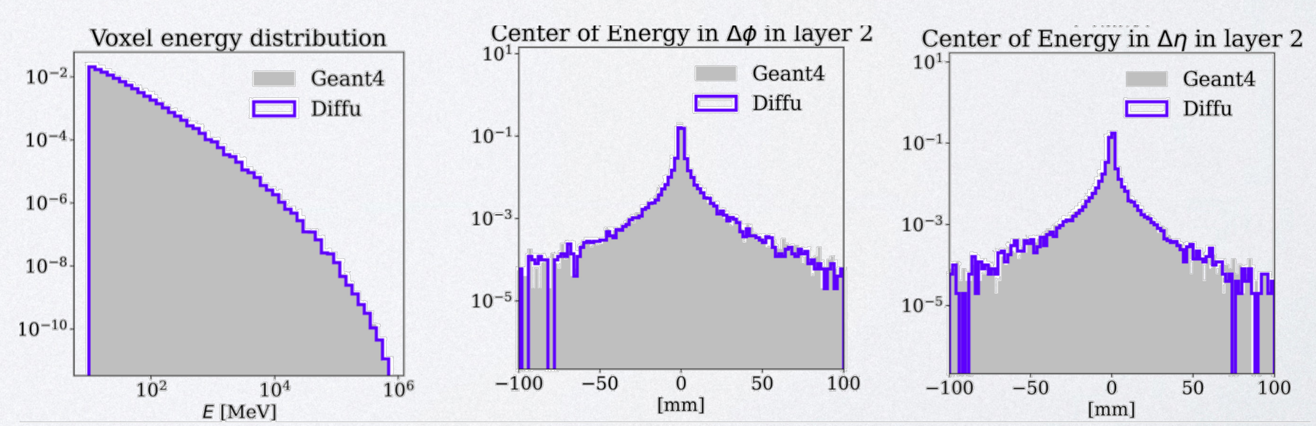


- Series of normalising flows
- 7μs/shower*
- Compared $O(10s)$/shower with GEANT

CaloScore (2206.11898) **CaloDiffusion** (CHEP 2023)



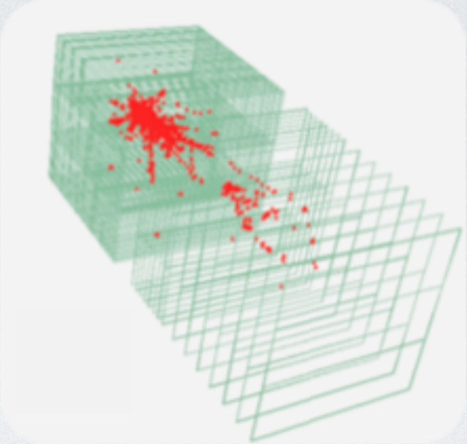
- Diffusion based models
- Good agreement with different datasets
- Generally slower than GANs
  - 0.3s/shower*



DATA REPRESENTATIONS

- Properties of LHC data:
    - Sparsity
    - High granularity
    - Irregular geometry
    - No fixed ordering
- Point clouds:
    - Store only the hits/particles
    - Retain feature precision
    - Are flexible, work for any geometry
    - Have no ordering

$\Rightarrow$ Natural representation for HEP data



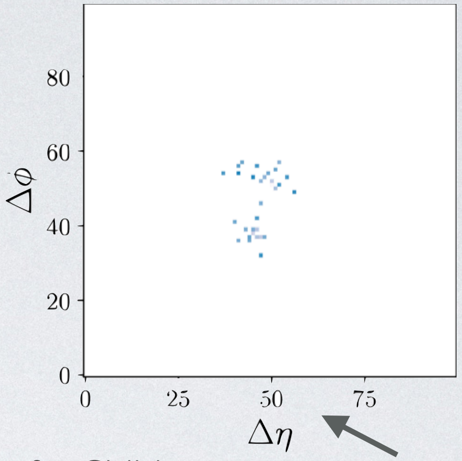
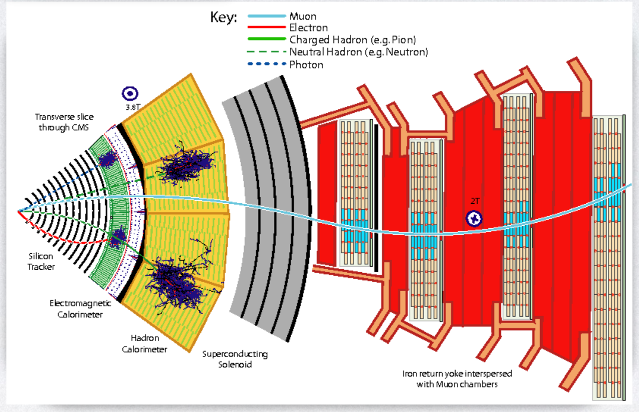
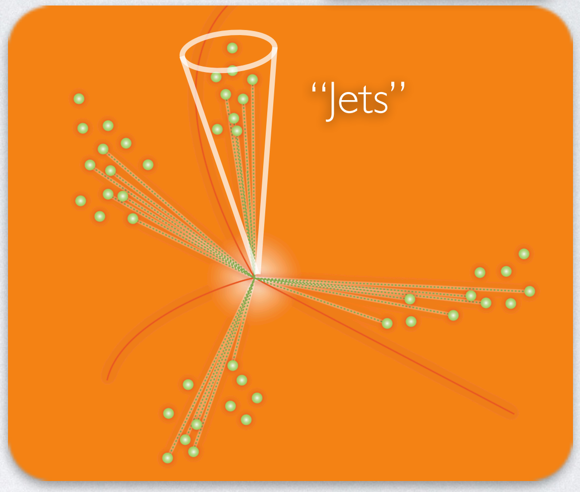


Image repr. for CNNs

RecoHits



Reco Particles



Hit coordinates, energy

Particle coords, momentum

or η φ p_T

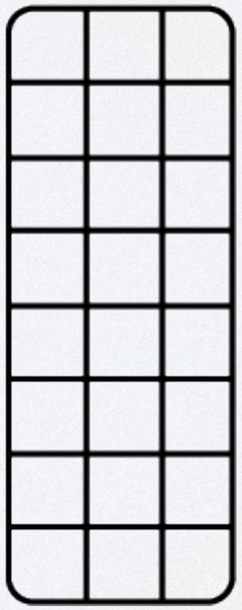
p_x p_y p_z

Particle/Hit 1

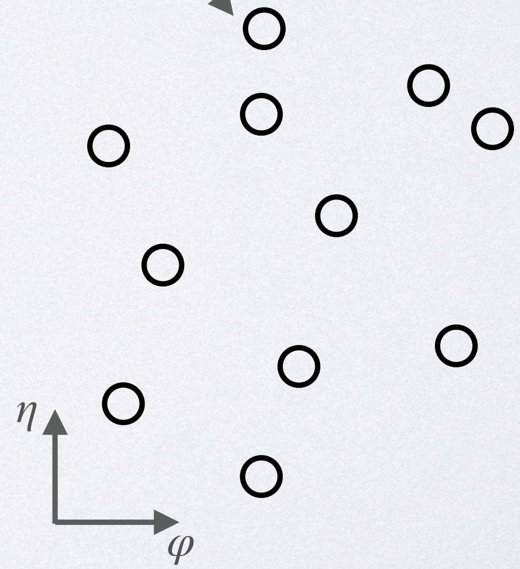
Particle/Hit 2

⋮

Row entry



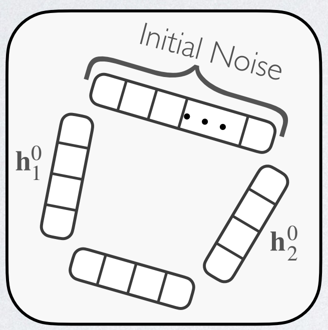
Energy/p_T node features

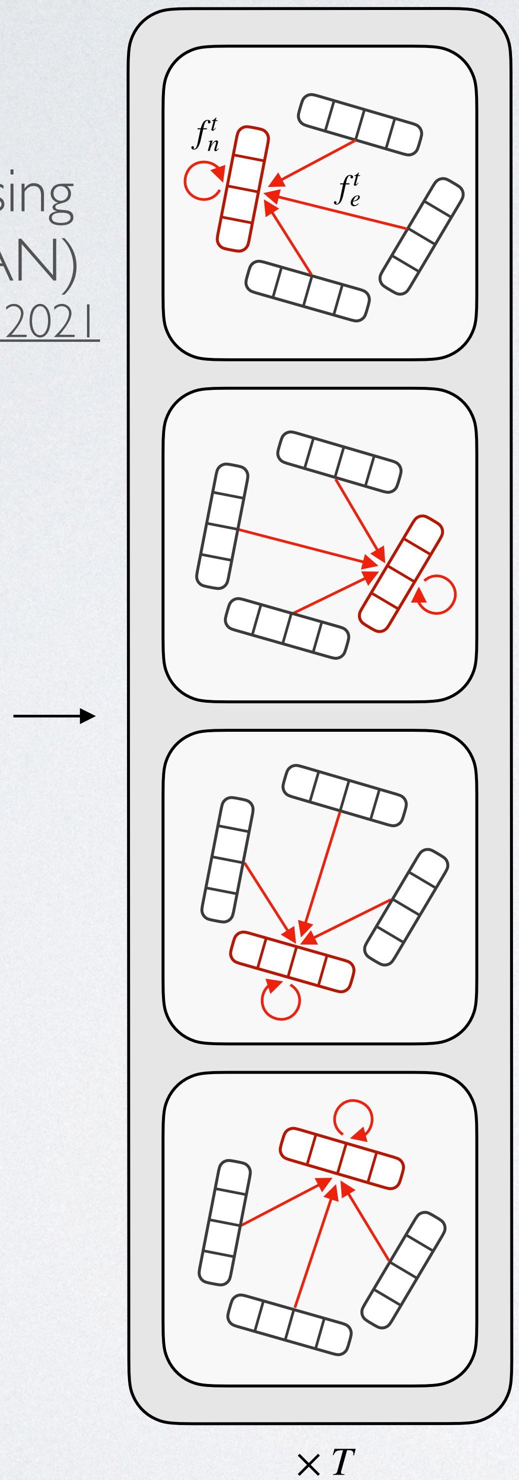


POINT CLOUD MODELS

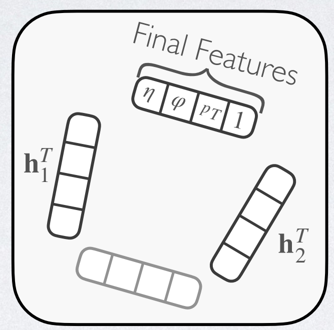
- Want to learn global features and inter-particle correlations
- And speed it up with transformers

Message Passing GAN (MPGAN)
RK et al., NeurIPS 2021



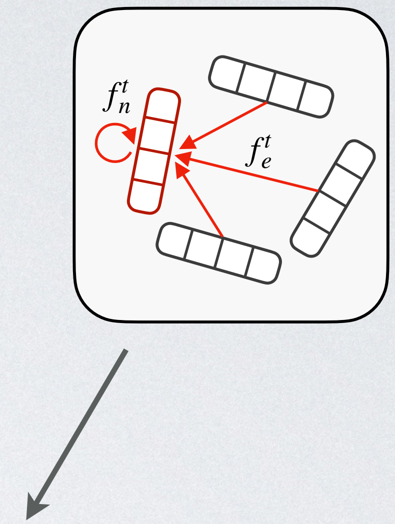


Generated Particle Cloud

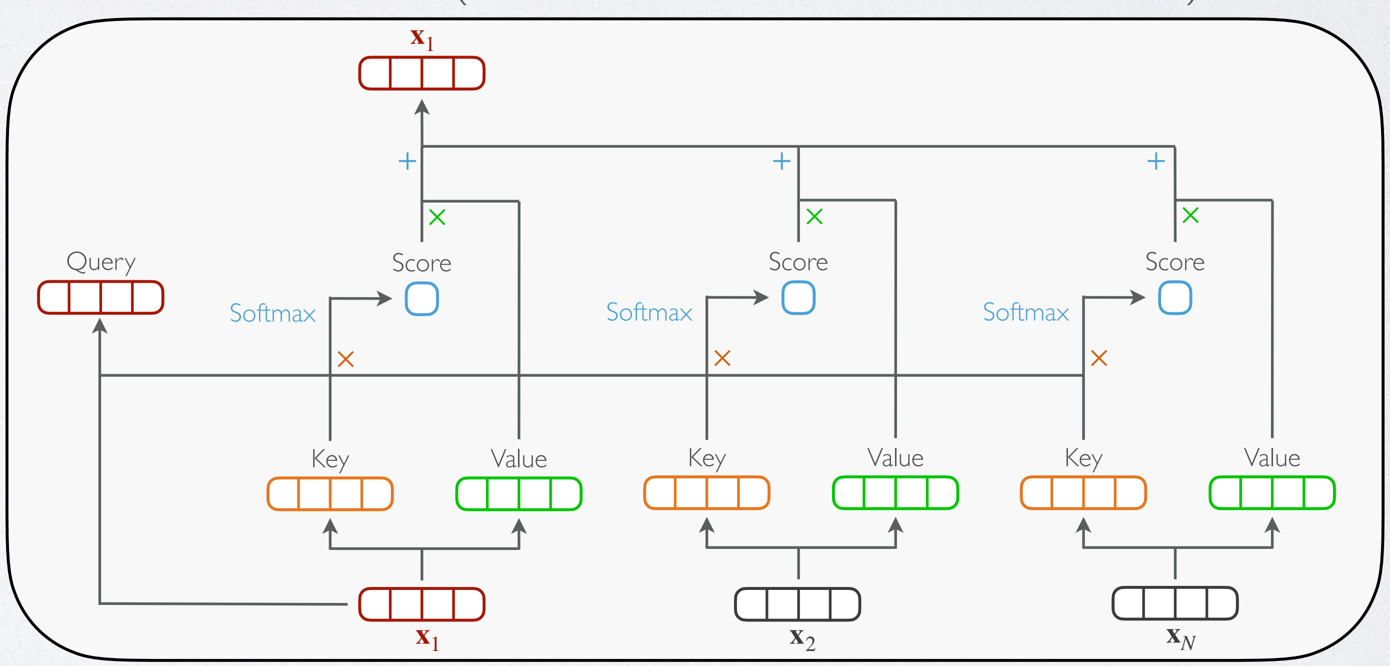


Generative Adversarial Particle Transformer (GAPT)
RK et al., PRD 2023

Message passing



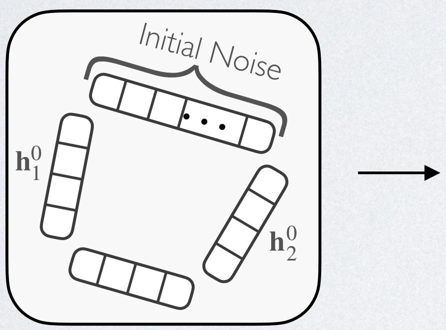
Self-attention (set transformers, Lee et al. ICML 2019)

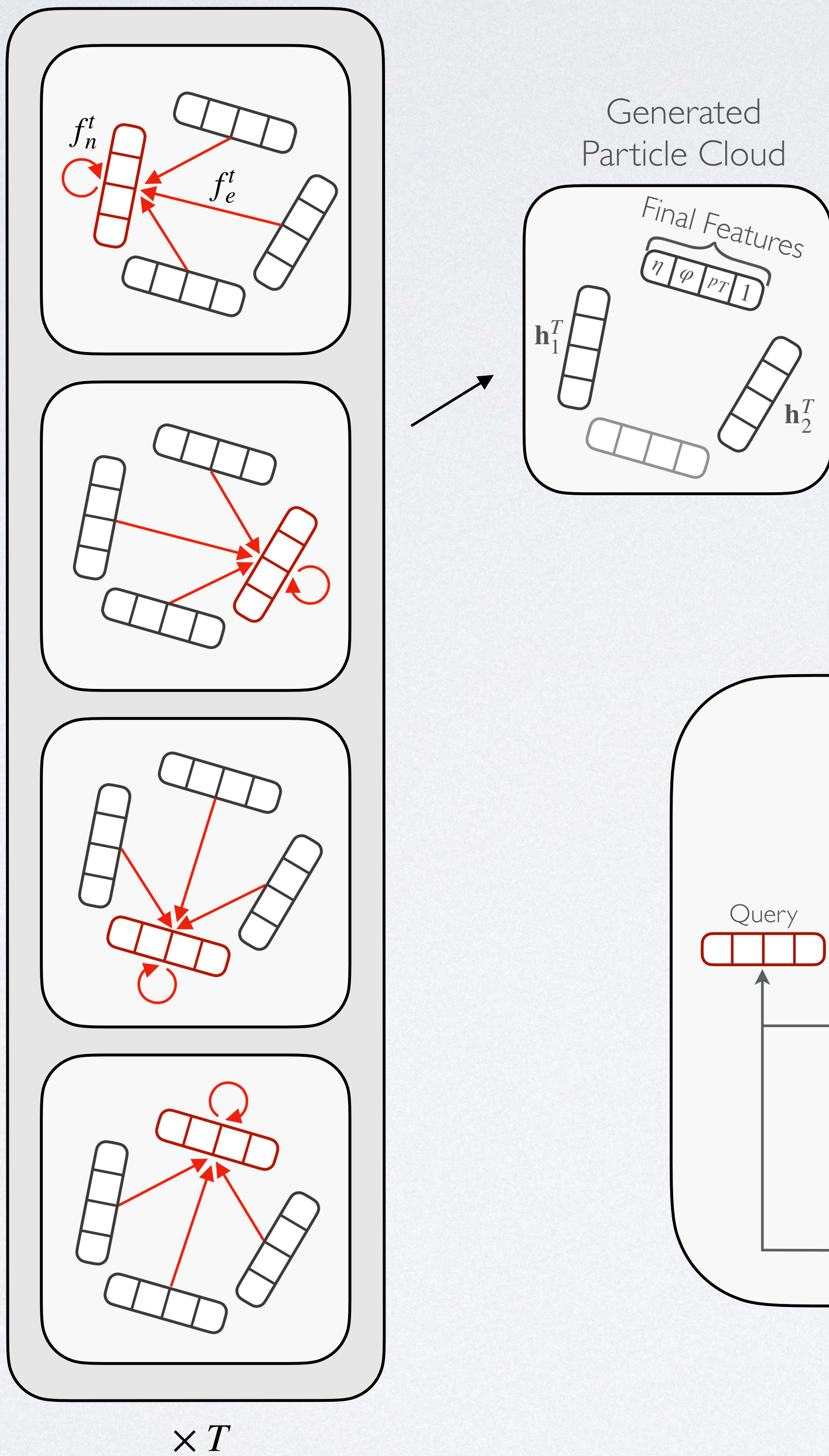


POINT CLOUD MODELS

- Want to learn global features and inter-particle correlations
- And speed it up with transformers

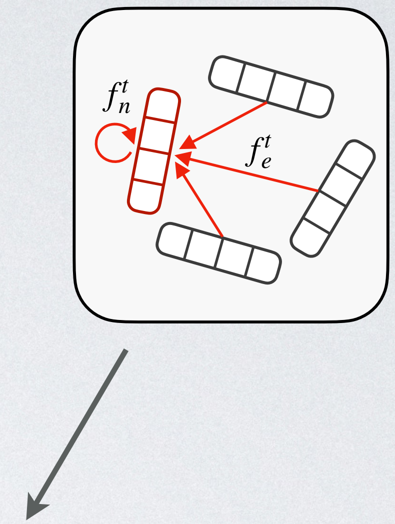
Message Passing GAN (MPGAN)
RK et al., NeurIPS 2021



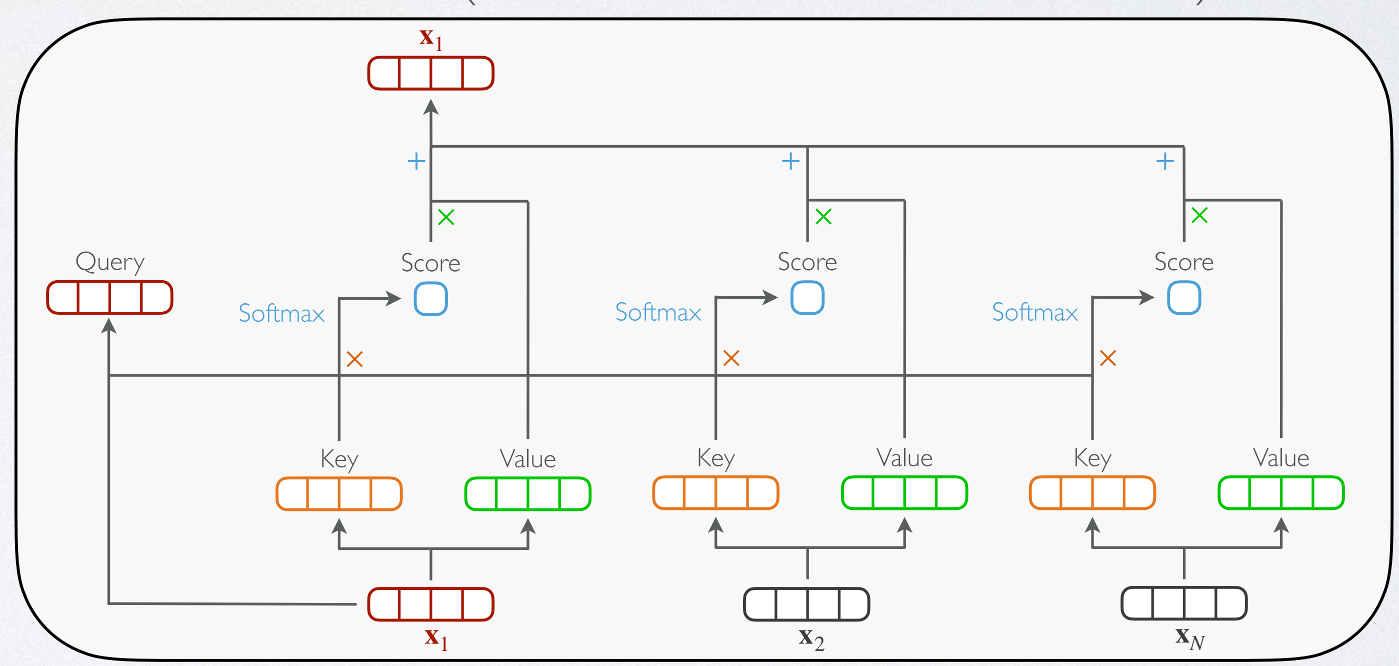


Generative Adversarial Particle Transformer (GAPT)
RK et al., PRD 2023

Message passing



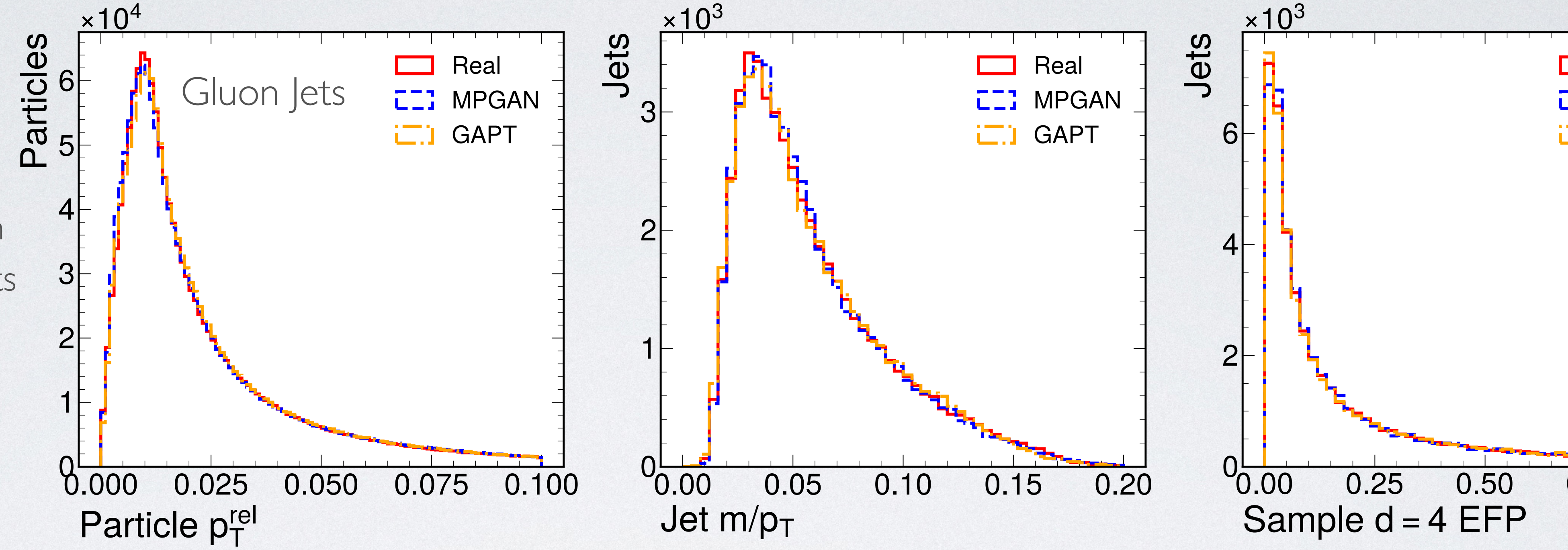
Self-attention (set transformers, Lee et al. ICML 2019)

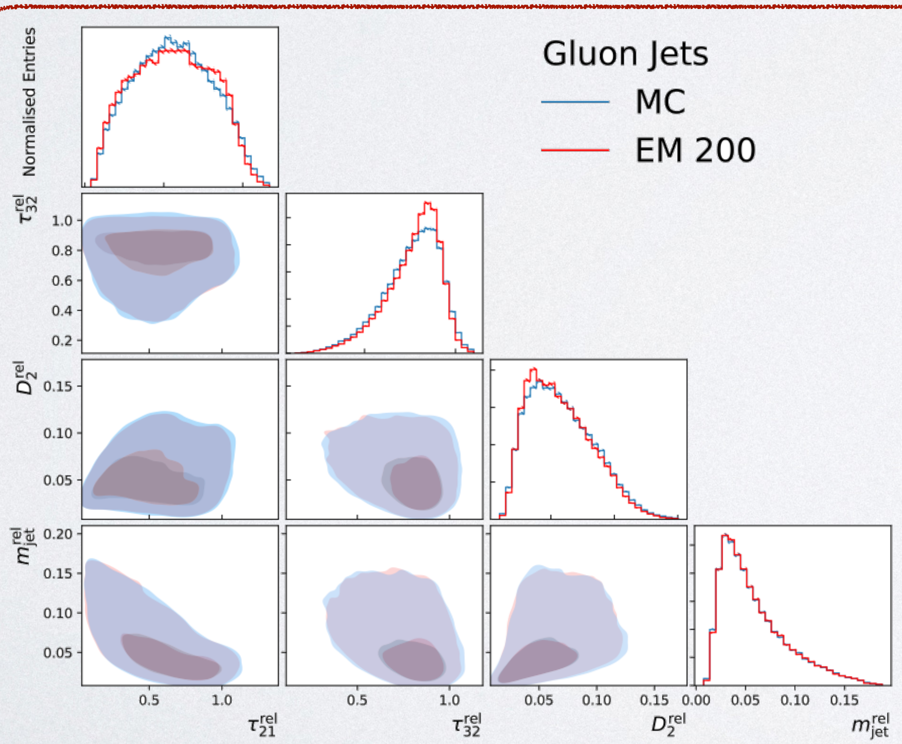


JETS

- Idea: learn distribution of PF cands per gen parton i.e., replacing Pythia + GEANT + reco

- GANs
- Good agreement with simplified CMS-like jets
- MPGAN: 41μs/jet*
  GAPT: 9μs/jet*





PC-JeDi (2303.05376)

FPCD (2304.01266)

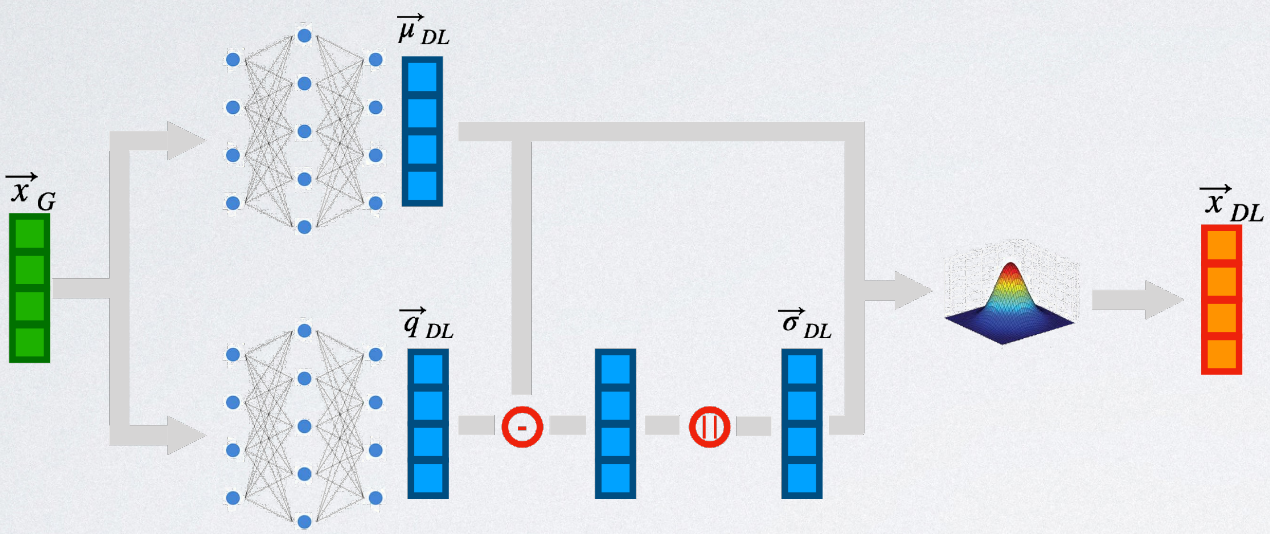
- Diffusion models
- Also point-cloud based
- Good agreement
- Slower than GANs
  - $\mathcal{O}$(1ms)/jet*
  - But distillation brings this down to $\mathcal{O}$(10μs)/jet

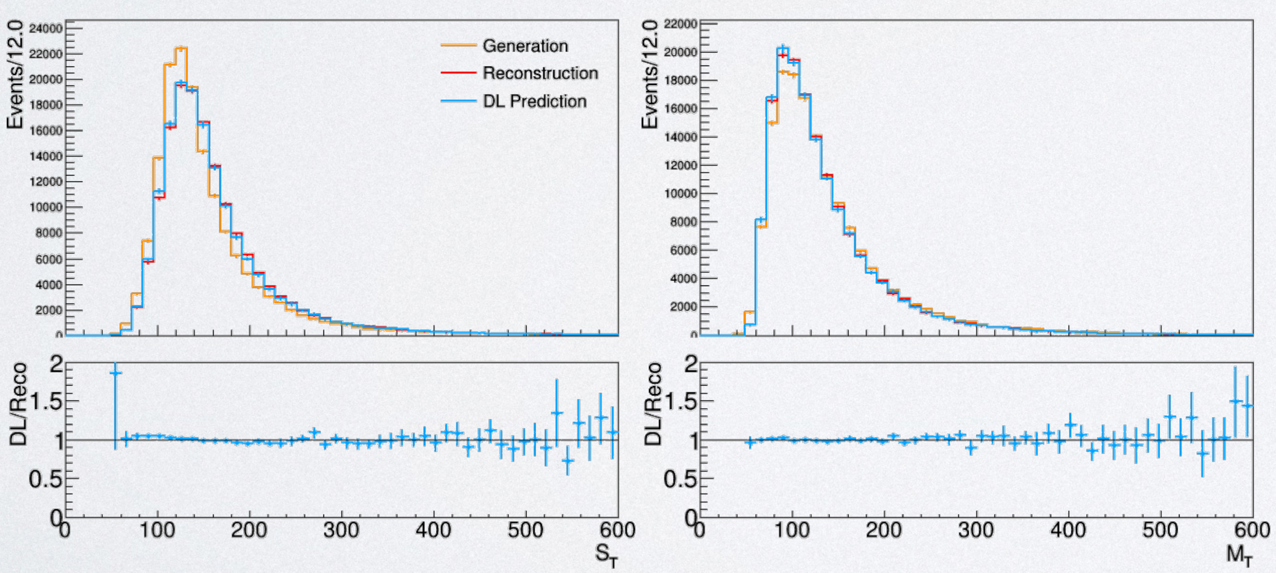
END-TO-END

- Idea: directly from gen-level features (jet $\vec{\mathbf{p}}$, MET etc.) to reconstructed features

Analysis-Specific FastSim

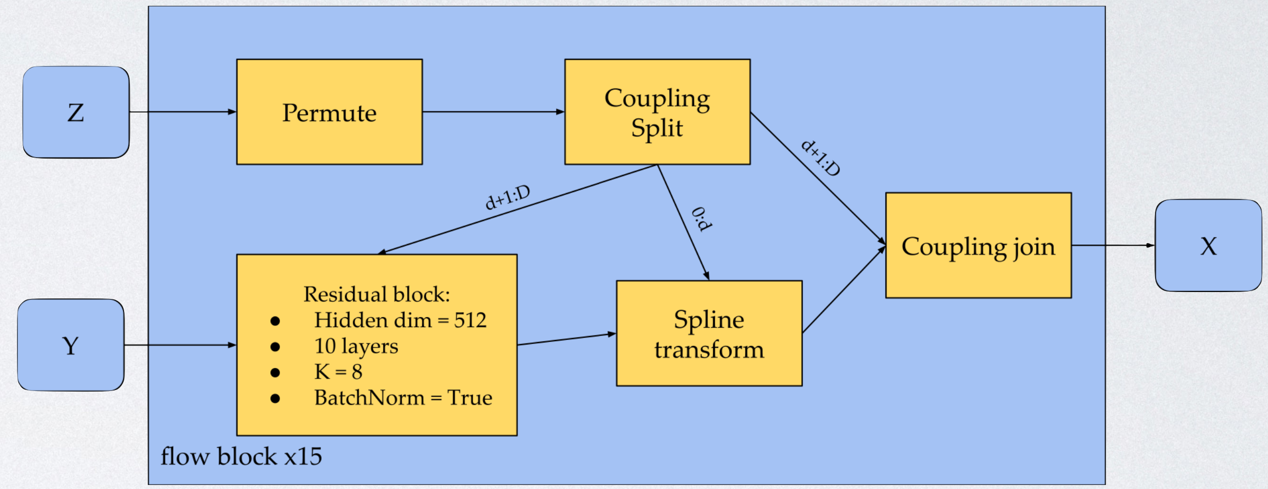
(Chen et al., CSBS 2021)

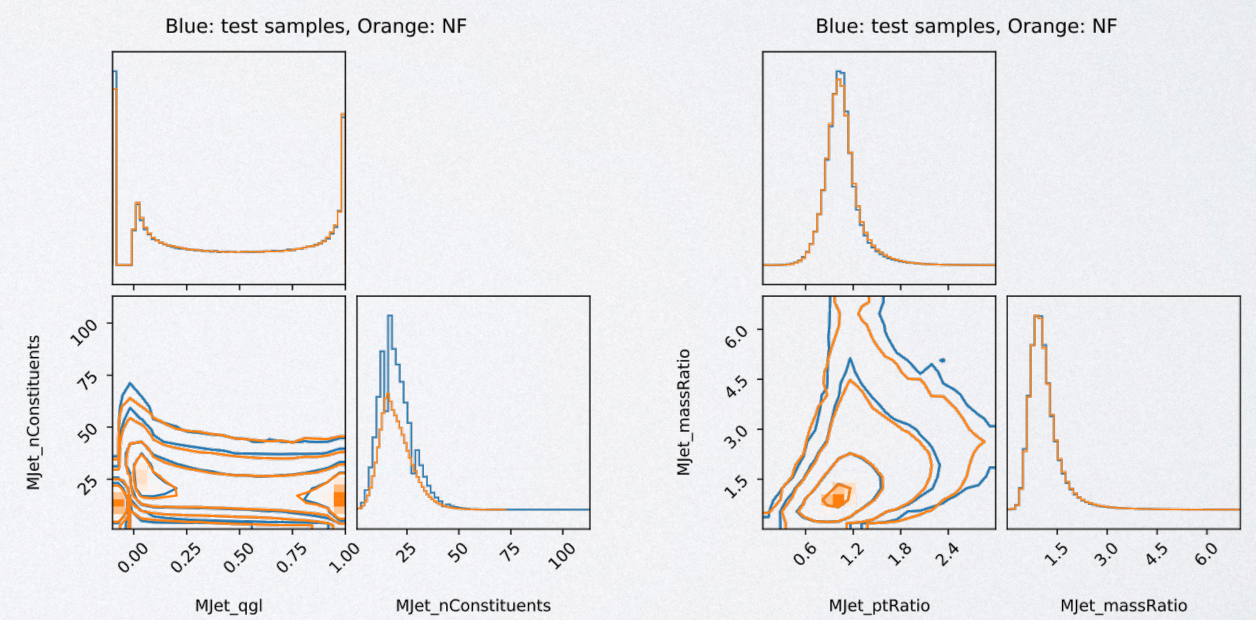




FlashSim

(Vaselli et al., *in prep*)



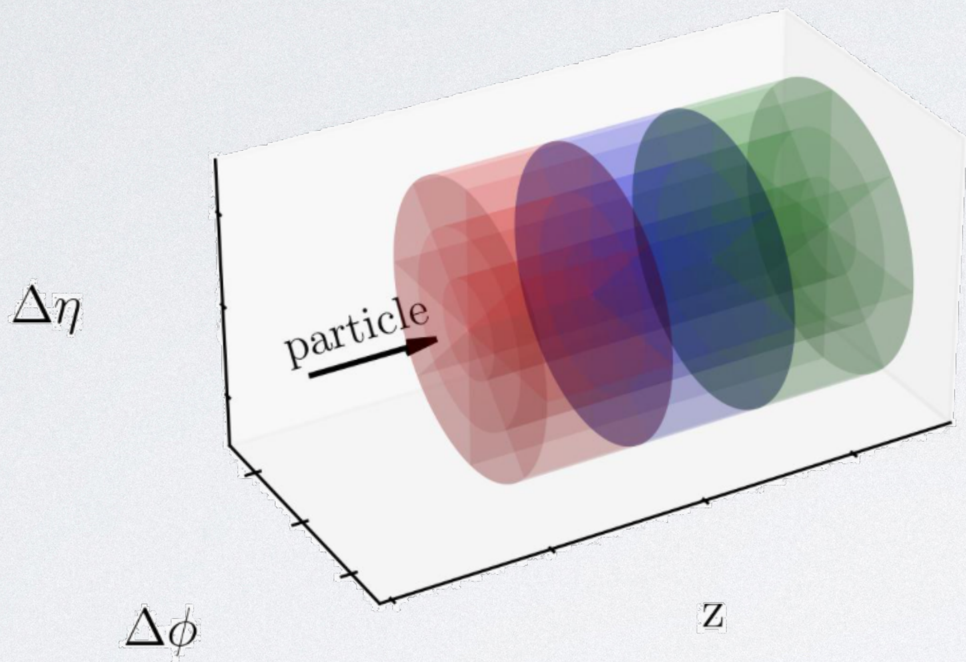


HOW DO WE CONVERGE?

COMMON DATASETS

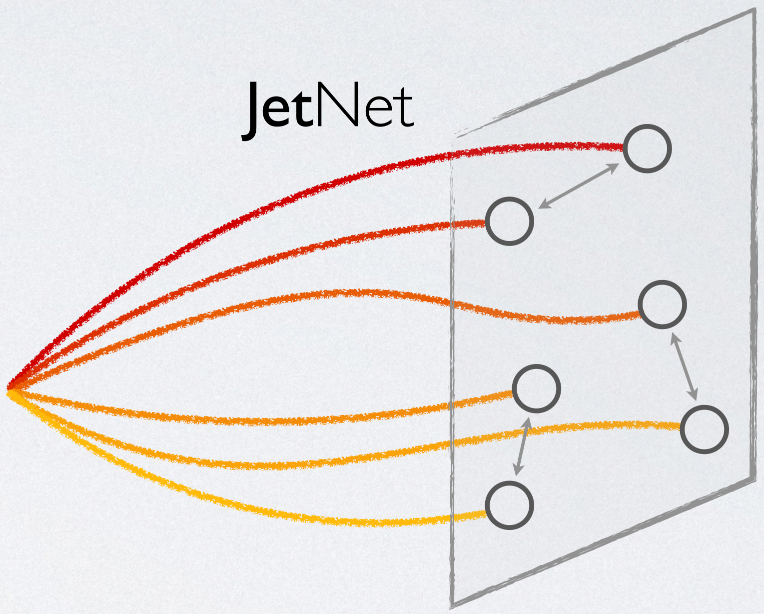
- Need common datasets to fairly compare models

CaloChallenge 2022-23



- Public "challenge" for calorimeter simulations
- 3 image-based datasets based on ATLAS-like and general detectors

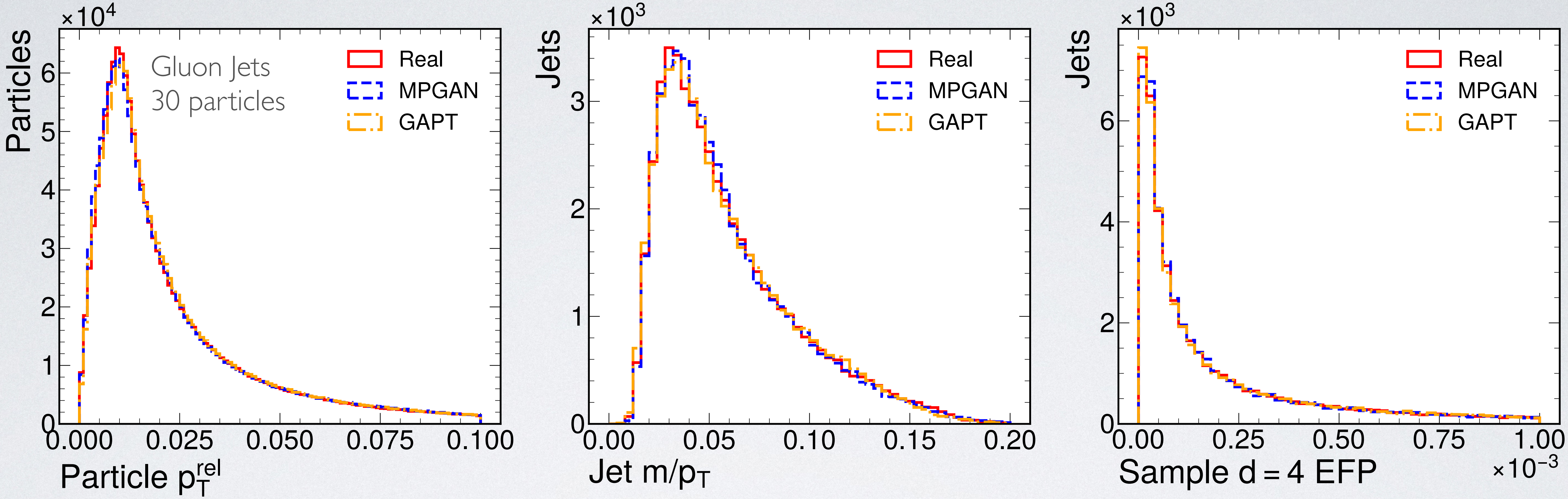
JetNet



- Public library and (collection of) jet datasets
- All point-cloud based, simplified reco
- Basis for majority of recent work on jets

See talk by M. Giannelli!

PROBLEM: MPGAN VS GAPT



- Both are very well performing on same dataset
- How do we decide which is better?
- How do we decide if either can replace existing simulations?

SOLUTION: COMMON METRICS

- Need standard, quantitative metrics to compare, validate, and trust models
- Studied in detail in 2211.10295 in terms of two-sample GoF tests
  - Traditional method is looking at 1 or 2D histograms
    - Should be quantified, can miss correlations
  - Many multivariate GoF tests studied
  - Fréchet and kernel physics distances found to be most sensitive
- Starting to be adopted for jets

	FPD $\times 10^3$	KPD $\times 10^3$	W_1^M $\times 10^3$	$W_{1p}^{p_T^{\mathrm{rel}}}$ $\times 10^3$
Truth	0.08 ± 0.03	-0.006 ± 0.005	0.28 ± 0.05	0.44 ± 0.09
MPGAN	$\mathbf{0.30 \pm 0.06}$	$\mathbf{-0.001 \pm 0.004}$	$\mathbf{0.54 \pm 0.06}$	0.6 ± 0.2
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08	$\mathbf{0.51 \pm 0.09}$

- Aim is to establish recommendations for validating ML simulations
  - See talk at PHYSTAT

CONCLUSION

- Expect fast simulations to be necessary for HL-LHC (if not sooner)
- Exciting potential to speed up full sim/reco pipeline with generative ML
  - Bonus: differentiable!
- Many approaches now in HEP
- Starting to converge:
  - Need to test on realistic detector datasets (or real data!)
  - Need to validate rigorously

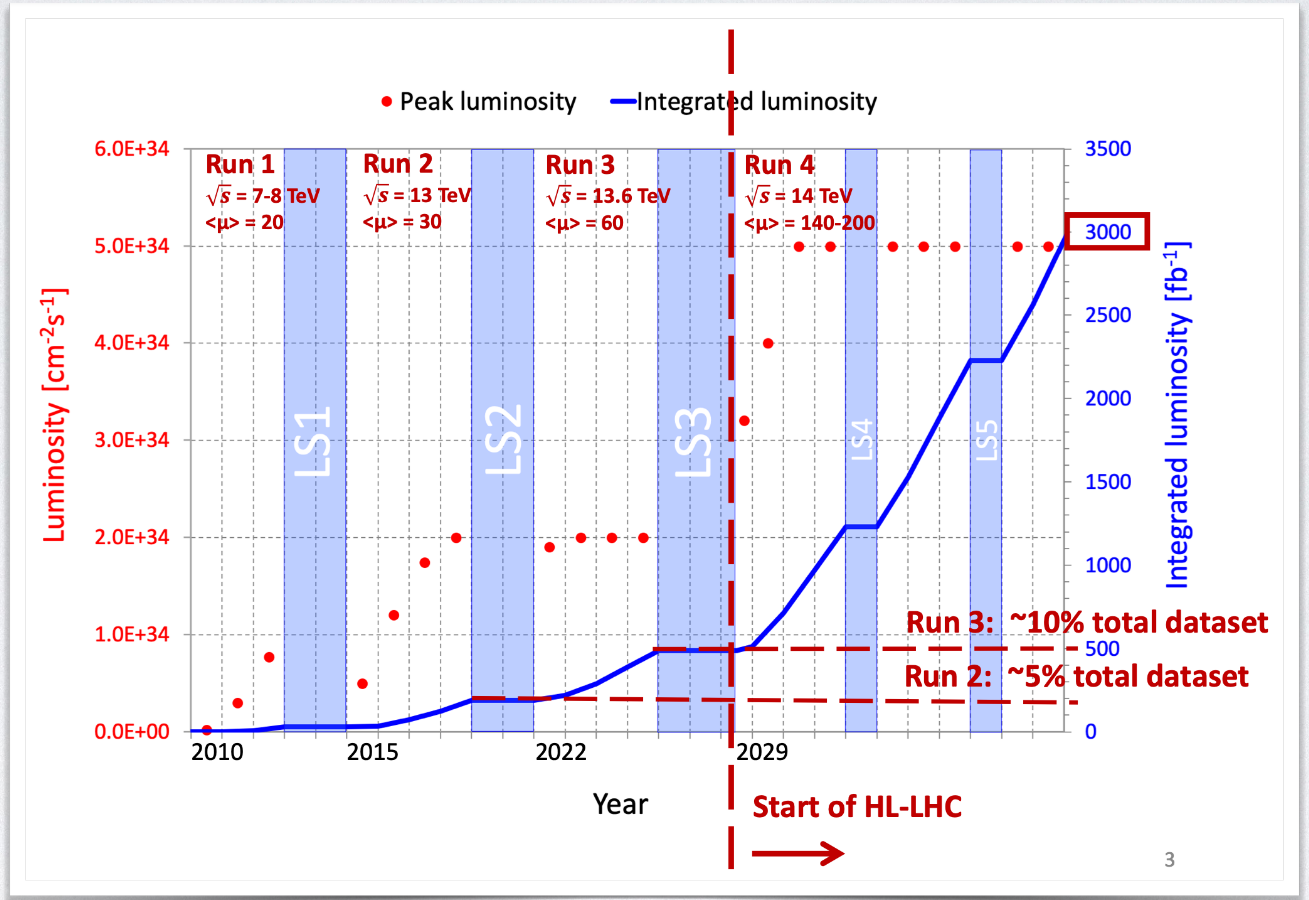
BACKUP

LHC SIMULATIONS

Sources
K. Pedro, HSF 2020
J. Duarte, ANL 2021, Video

- Full detector simulation takes ~40% of grid CPU resources

Luminosity (~ how much data is collected) is only going to increase



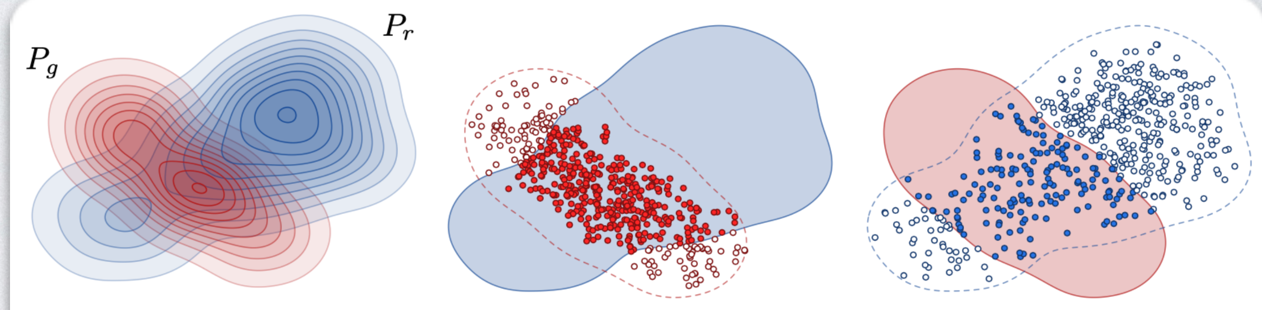
- Order-of-magnitude more simulations needed in the next decade

- Complexity of the simulations will also increase

- ML a possible solution?

MORE METRICS

- Precision and recall (Kynkäänniemi et al 2019)
  - Estimate real and generated manifold
  - Can disentangle quality and diversity

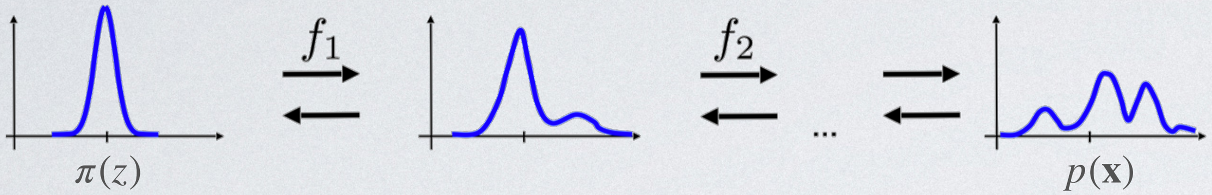


(a) Example distributions (b) Precision (c) Recall

- Classifier-based metrics: train a classifier between real and generated data
  Friedman 2003, Paz and Oquab 2017 (C2ST), Krause and Shih (2021)
  - Can be powerful test of quality and diversity
  - Practical limitations: interpretability, generalising to conditional generation, standardising a specific architecture for all alternative hypotheses, reproducability of trainings, inefficiency
  - In terms of GOF testing: comparing different test statistics for different models

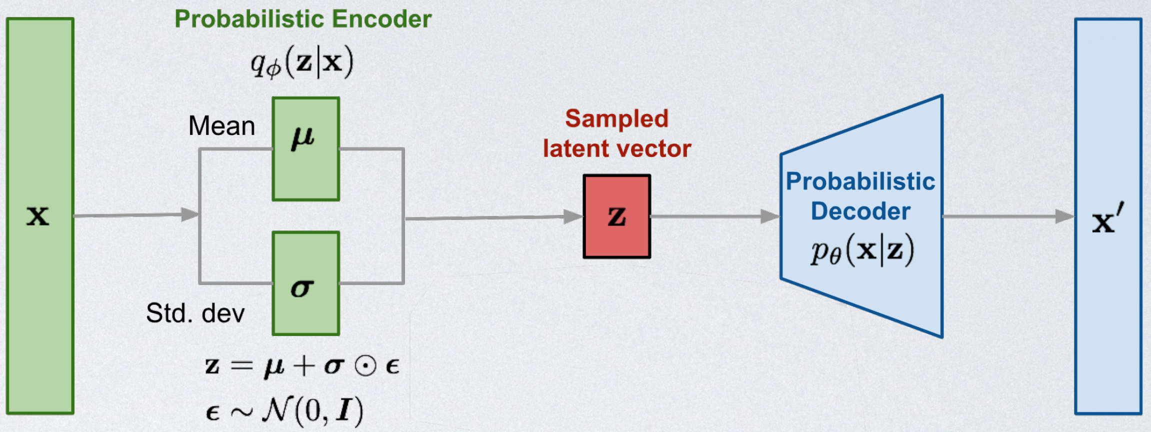
ML LANDSCAPE

Flows



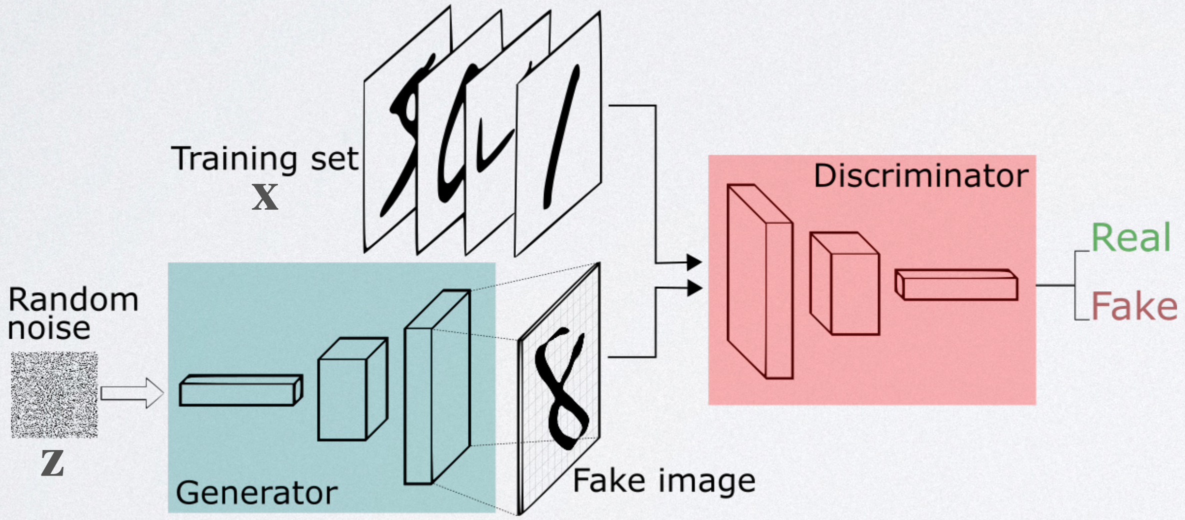
- Learn transformations from simpler $\pi(z)$ to $p(\mathbf{x})$
- Access to exact likelihood
- But restrictive

Variational Autoencoders (VAEs)



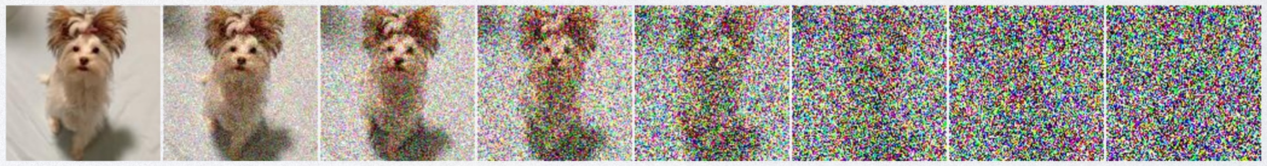
- Maximise an approximation to the likelihood
- Can also be restrictive

Generative Adversarial Networks (GANs)



- Minimise loss wrt to discriminator classifying real or fake
- Less restrictive, generally higher performing
- Difficult to train

(Score-based) Diffusion

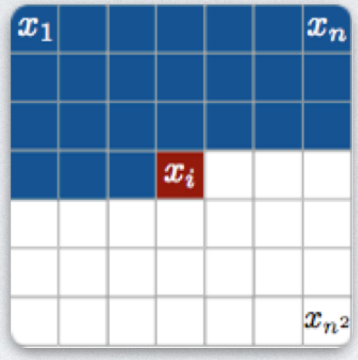


- Learn $-\partial \ln p(\mathbf{x})/\partial \mathbf{x}$ (score) instead of $p(\mathbf{x})$ directly
- Less restrictive, score doesn't need to be normalised
- Current industry SOTA (DALL-E, StableDiffusion etc.)
- But slow - need O(100)s of steps along the score

DIRECT MODELLING

Autoregressive

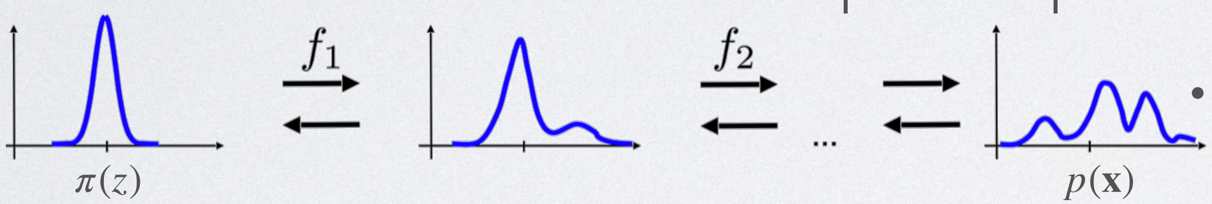
$$p(\mathbf{x}) = p(x_1)\prod_{i=2}^{N-1} p(x_i \mid x_{<i})$$



- Model assumption: x_i can be predicted entirely by $x_{<i}$
- Iteratively output x_i based on $x_{<i}$ using learnt $p(x_i \mid x_{<i})$

Flow-based

$$p(\mathbf{x}) = \pi\left(\mathbf{x} = f^{-1}(z)\right)\left|\frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right|$$

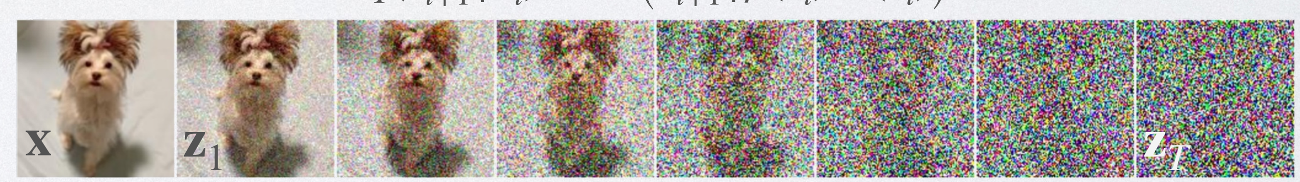


- Backward: learn transformation from $p(\mathbf{x})$ to simpler $\pi(z)$
- Forward: sample $\pi(z)$, **invert** transformation, get $p(\mathbf{x})$

Diffusion-based

- Backward: iteratively add gaussian noise
- Forward: $p(\mathbf{z}_i \mid \mathbf{z}_{i+1})$ network learns to denoise

$$q(\mathbf{z}_{i+1} \mid \mathbf{z}_i) = \mathcal{N}\left(\mathbf{z}_{i+1} \mid \mu(\mathbf{z}_i), \sigma(\mathbf{z}_i)\right)$$



$$p(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{z}_1)\left(\prod_{i=1}^{T-1} p(\mathbf{z}_i \mid \mathbf{z}_{i+1})\right)p(\mathbf{z}_T)$$

1. Access to exact likelihood
2. Simple $-\ln p(\mathbf{x})$ loss
3. Stable training

- **But** in practice they are typically outperformed by GANs (next slide)

LATENT VARIABLE MODELS

- Assume high dimensional data $\mathbf{x}$ can be characterised by lower dimensional 'latent' (hidden) features $\mathbf{z}$
- Generative process: sample from simpler prior $\mathbf{z} \sim p(\mathbf{z})$ and learn $p(\mathbf{x} \,|\, \mathbf{z}) \Rightarrow p(\mathbf{x}) = \int p(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z}) d\mathbf{z}$

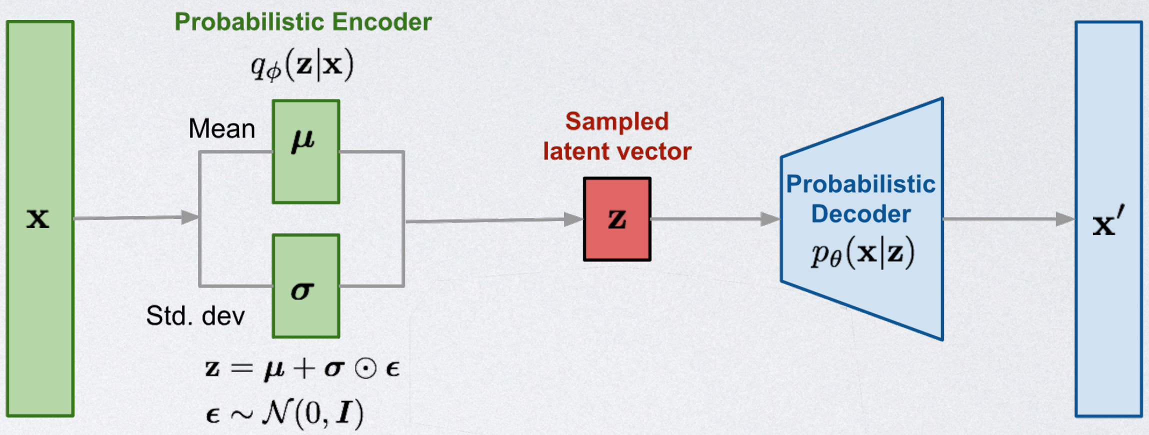
Variational Autoencoder

- 'Evidence-based Lower BOund' (ELBO) of $\ln p(\mathbf{x})$:

$$\mathrm{ELBO} = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}[\ln p(\mathbf{x} \,|\, \mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z})\right]$$

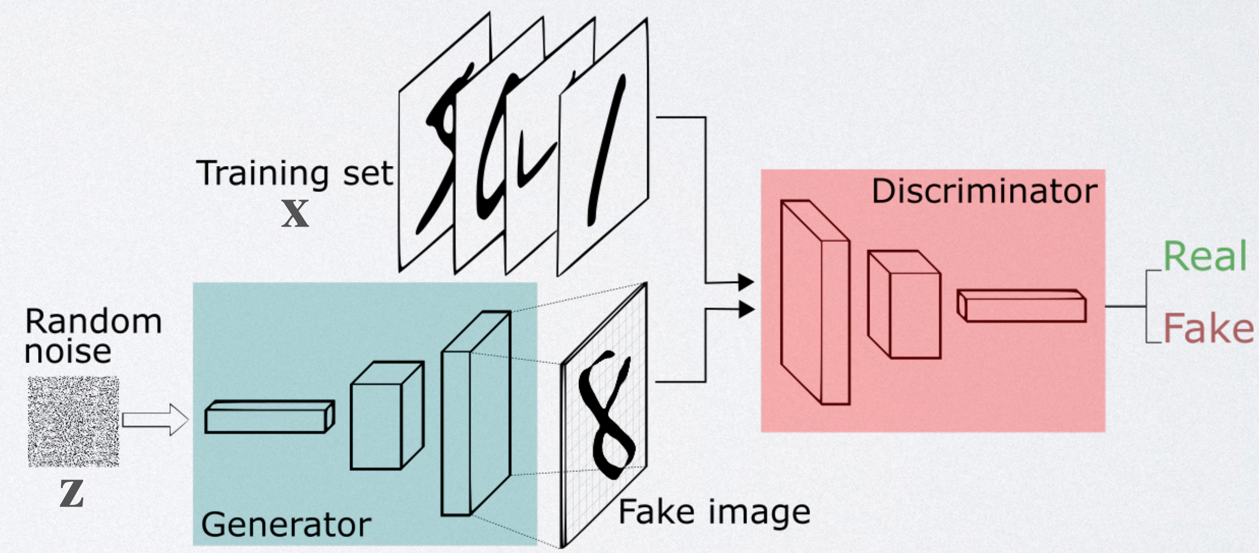
Autoencoder reconstruction loss

Divergence between variational posterior and assumed prior



Generative Adversarial Networks

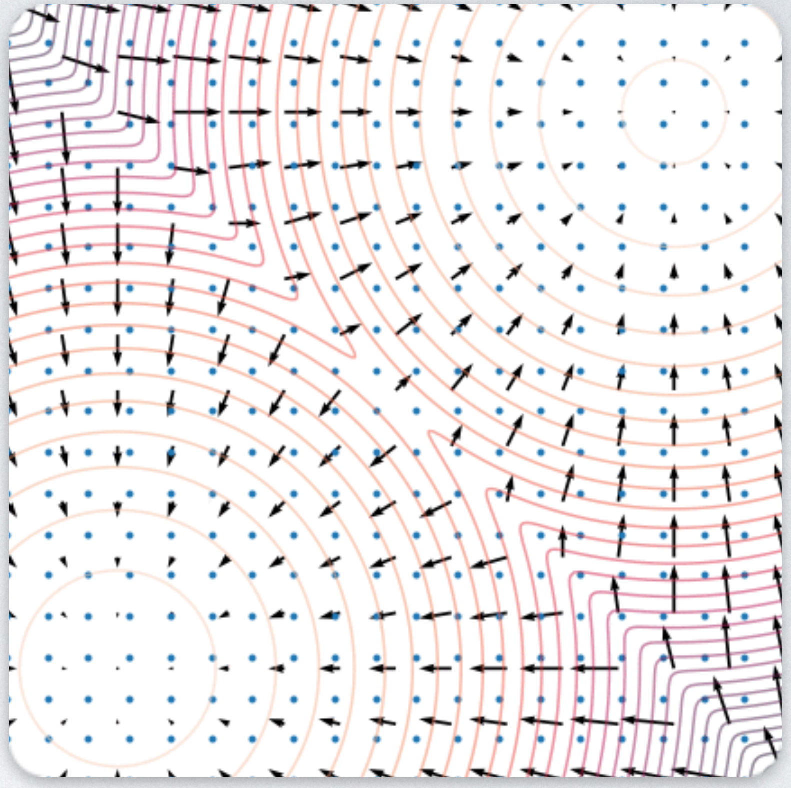
- Abandon likelihood-based loss approach
- Iteratively train 'discriminator' network as an **adversarial loss** for the 'generator'
- 1) Hard to train; 2) lose likelihood; 3) adversarial; **but** when done right tends to be most performant*



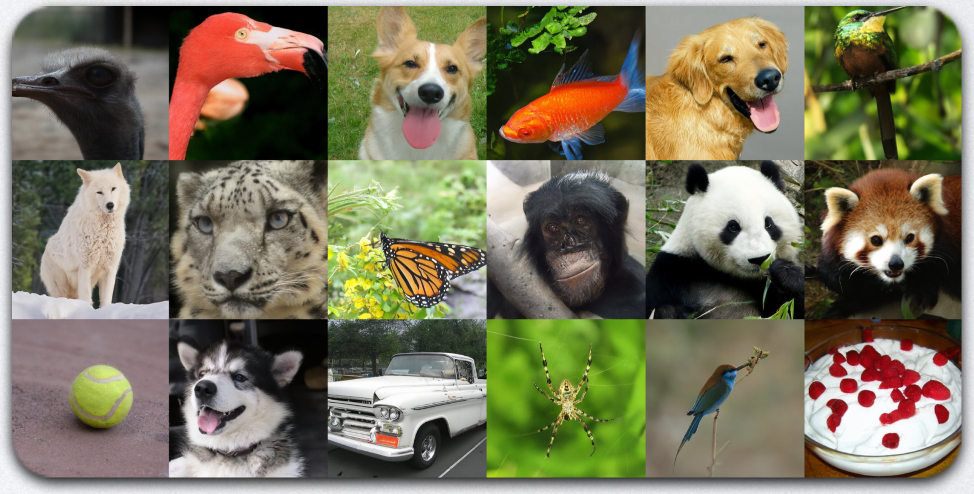
- ***UNTIL** last year where **score-based** diffusion models started beating GANs for the first time! (next slide)

SCORE-BASED MODELS

- Modelling $\nabla_{\mathbf{x}} \ln p(\mathbf{x})$ (the '**score** function') instead of $p(\mathbf{x})$
- Retain access to likelihood, no need for normalisation
- Many methods for learning $\nabla_{\mathbf{x}} \ln p(\mathbf{x})$: '**score-matching**'
  - e.g. **diffusion models** (but interpreted differently)
- With lots of clever tricks, such models beat GANs
- **But** very slow, and very new so will require model development and tuning - still, very interesting area!



$p(\mathbf{x})$ can be accessed via the score function (source)



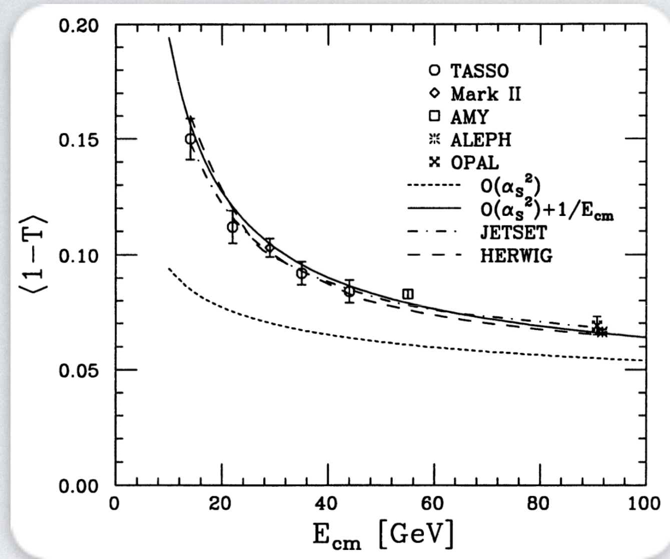
Diffusion Models Beat GANs on Image Synthesis (Dhariwal and Nichol 2021)

TEST CRITERIA

- To **trust** generated data, tests should be:
  - Sensitive to quality
  - Sensitive to diversity
  - Multivariate (for correlations & conditional generation)
  - Interpretable
- To **compare** generative models, tests should be:
  - Standardised
  - Reproducible
  - ~Efficient

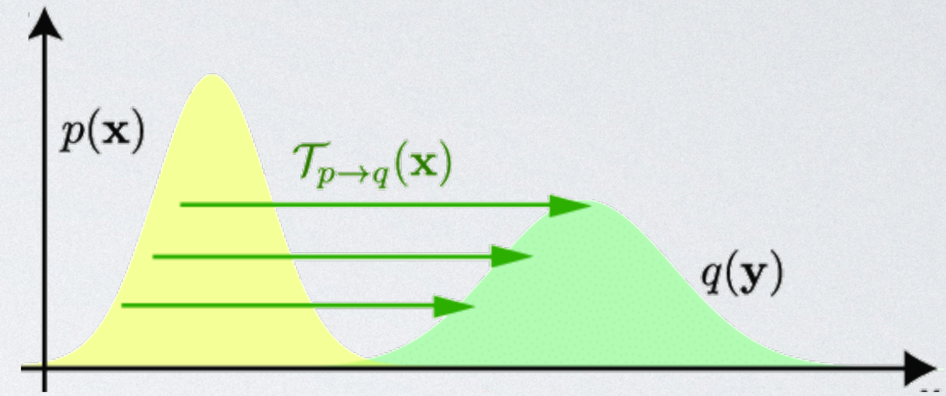
IDEAS

MC generator evaluation (Ellis et al '96)



- Look at histograms
  - Useful, but need to quantify
  - Only 1 or 2D
- Multi-dim goodness-of-fit tests?

- Wasserstein distance
  - Sensitive, but hard to estimate



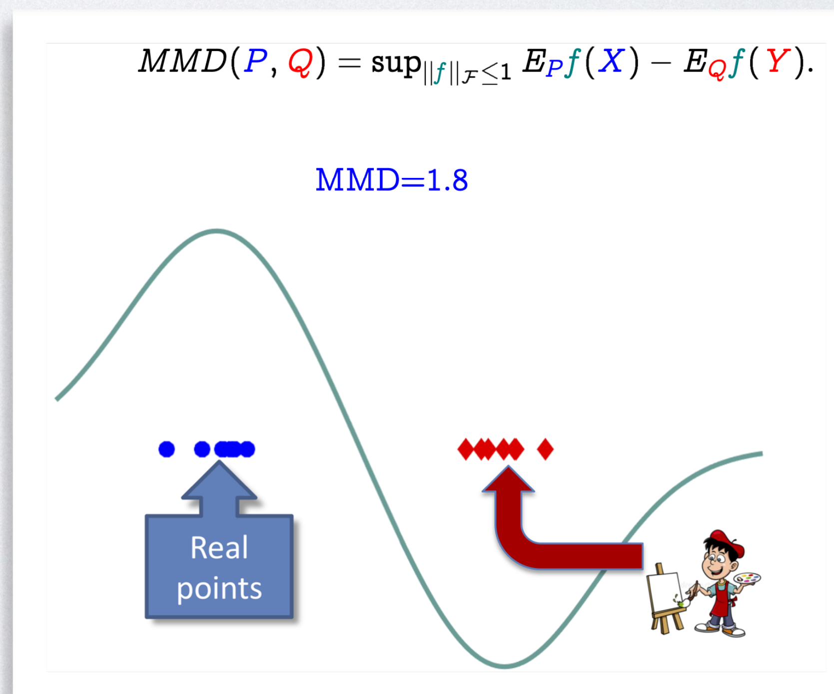
- Fréchet distance
  - Approximate Wasserstein
  - Standard in ML (FID)

$$\text{FGD} = \text{Frechet}(\mathcal{N}(\mu_r, \boldsymbol{\Sigma}_r), \mathcal{N}(\mu_g, \boldsymbol{\Sigma}_g))$$

$\{\mathbf{x}_{\text{real}}\}$ $\{\mathbf{x}_{\text{gen}}\}$

- Maximum mean discrepancy (MMD)
  - Embed samples in kernel space
  - Distance in embedding space
  - Also used in ML (KID)

Gretton 2020



MORE ON IPMS

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

- Fréchet Gaussian Distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID)
  - Computationally efficient
  - Gaussian assumption
  - ~~Biased~~ (FGD_∞ - extrapolate to infinity)

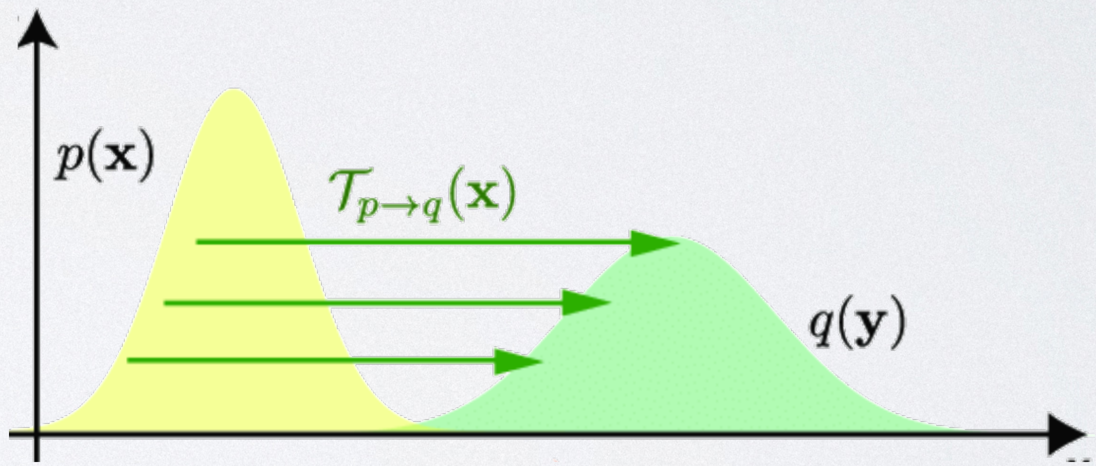
$$\text{FGD} = \text{Frechet}(\mathcal{N}(\mu_{\text{r}}, \mathbf{\Sigma}_{\text{r}}), \mathcal{N}(\mu_{\text{g}}, \mathbf{\Sigma}_{\text{g}}))$$

$\{\mathbf{x}_{\text{real}}\}$ $\{\mathbf{x}_{\text{gen}}\}$

MORE ON IPMS

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

- Wasserstein $p-$distances (W_p):
    - $\mathcal{F}$ is all K-Lipschitz functions
    - "Work" needed to transport probability mass
    - Sensitive to quality and diversity
    - Computationally challenging for large N, D
    - Biased estimators

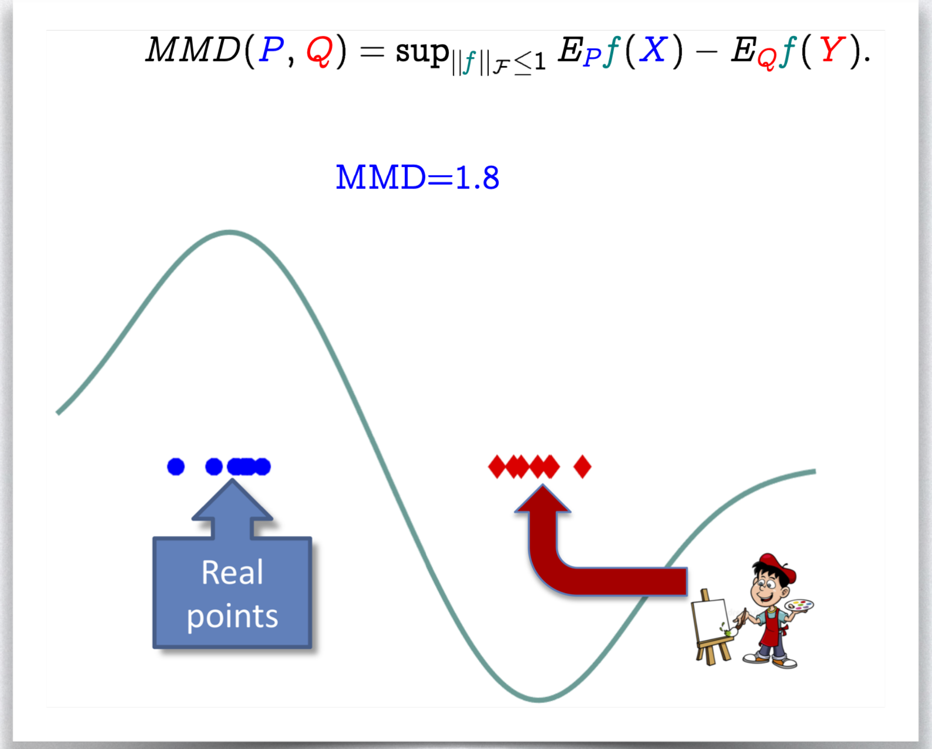


MORE ON IPMS

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

- Maximum mean discrepancy (MMD)
  - $\mathcal{F}$ is reproducing Kernel Hilbert space (RKHS) for a chosen kernel $k(x, y)$
  - Distance between embeddings of p_{real} and p_{gen} in $\mathcal{F}$
  - Proposed in computer vision (KID), 3rd order polynomial kernel
  - Unbiased estimators
  - Kernel dependent

Gretton 2020

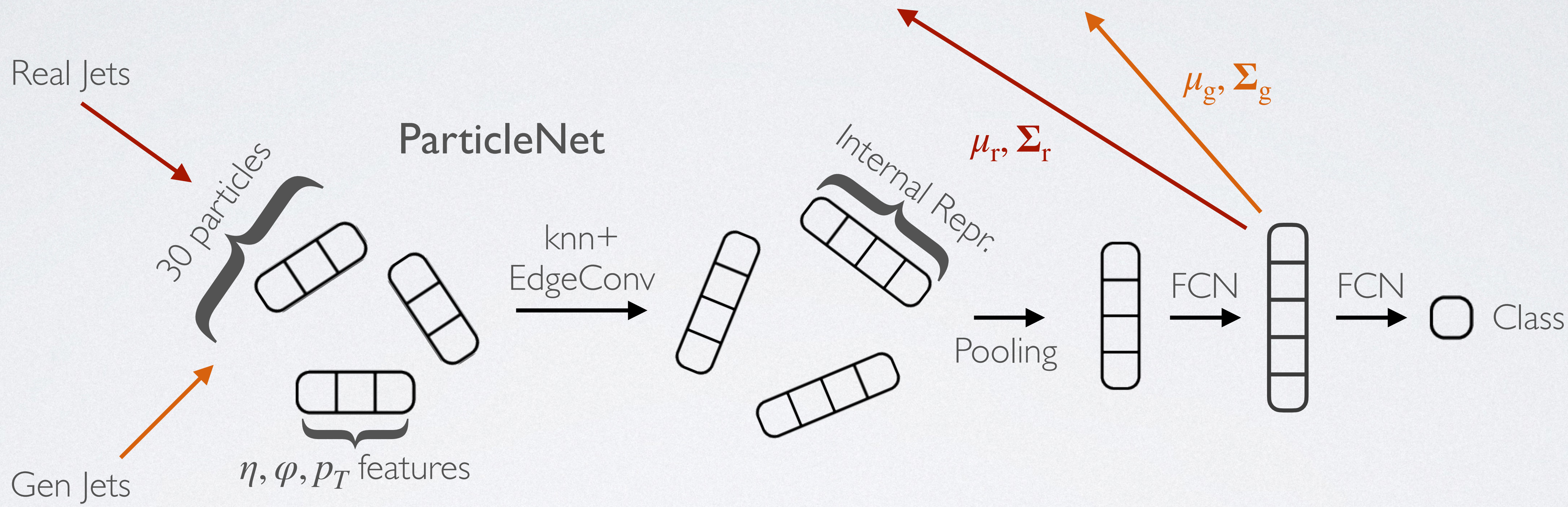


FRÉCHET <CLASSIFIER> DISTANCES

- Machine learning version of this: use classifier hidden features instead!   Kansal et al., NeurIPS 2021

- Example: apply to jet generation using pre-trained ParticleNet graph classifier:

$$\mathrm{FGD} = \mathrm{Frechet}(\mathcal{N}(\mu_{\mathrm{r}}, \boldsymbol{\Sigma}_{\mathrm{r}}), \mathcal{N}(\mu_{\mathrm{g}}, \boldsymbol{\Sigma}_{\mathrm{g}})) = ||\mu_{\mathrm{r}} - \mu_{\mathrm{g}}||^2 + \mathrm{Tr}\left[\boldsymbol{\Sigma}_{\mathrm{r}} + \boldsymbol{\Sigma}_{\mathrm{g}} - 2(\boldsymbol{\Sigma}_{\mathrm{r}}\boldsymbol{\Sigma}_{\mathrm{g}})^{1/2}\right]$$



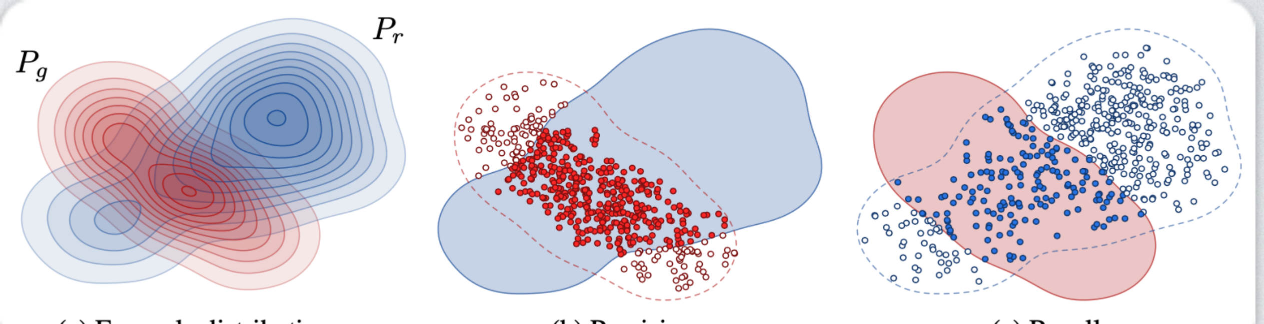
- High-performing classifier learns salient hidden features from data

- Retain sensitivity to **quality, diversity** from W_1, **reproducible** and **efficient** plus:
    - Single aggregate score, correlations (Σ) between features, easy to scale

MAXIMUM MEAN DISCREPANCY

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

TESTS FOR QUALITY / DIVERSITY



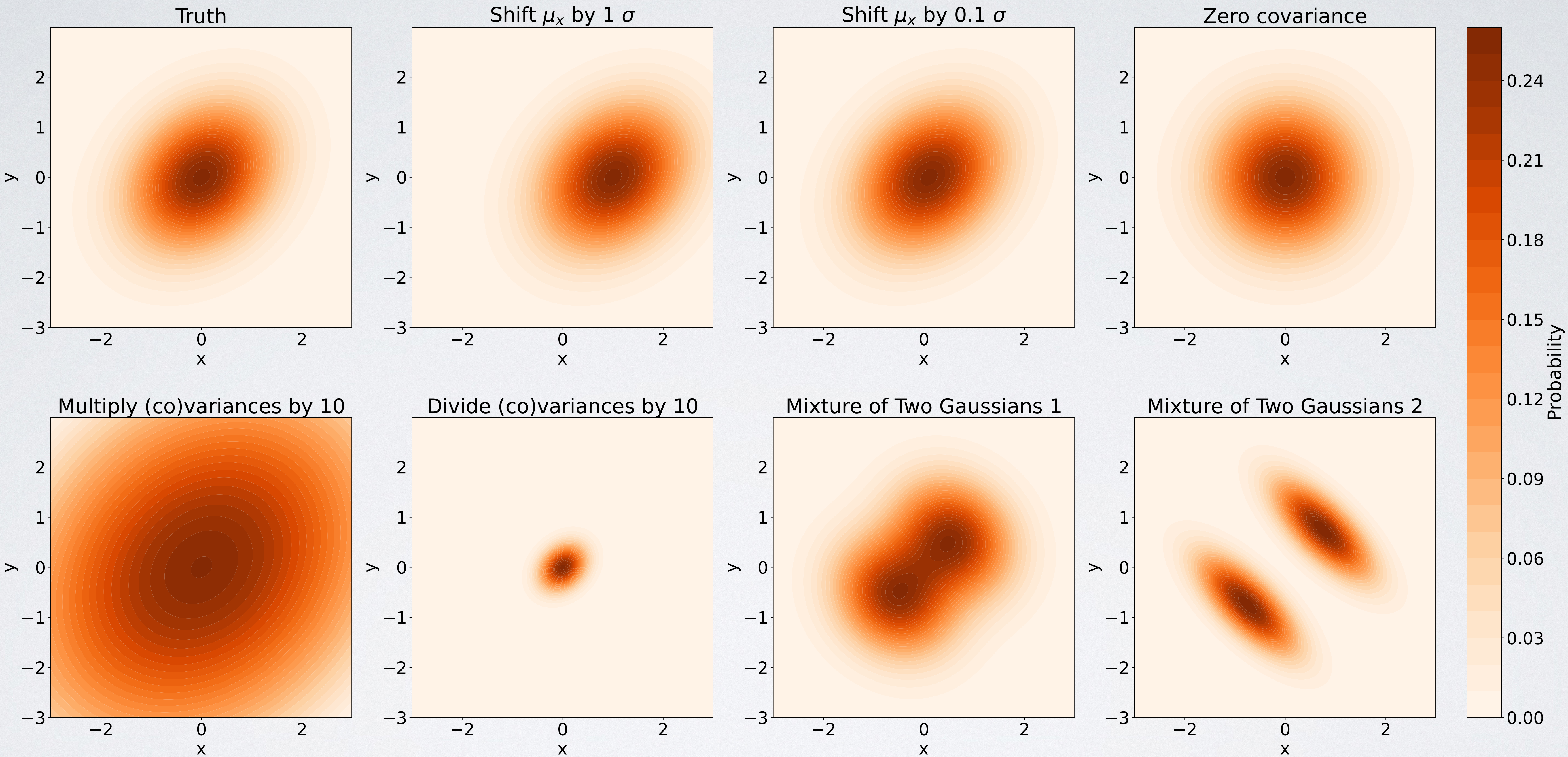
(a) Example distributions (b) Precision (c) Recall

- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)
  - Estimate real and generated manifold using k-nearest-neighbours
  - Precision: fraction of generated samples lying within real manifold (quality)
  - Recall: fraction of real samples which lying within gen manifold (diversity)
- Density & Coverage (Naeem et al 2020)
  - Like P&R, but takes into account density of real manifold

TOY DISTRIBUTIONS

- First testing on toy Gaussian distributions

Tests if metrics are sensitive to correlations



Tests sensitivity to quality

Tests sensitivity to diversity

Mixture with same mean, variance and covariance as truth: Tests sensitivity to shape of distribution

Same statistics, but easier to distinguish (by eye)

RESULTS

Metric	Truth	Shift μ_x by 1σ	Shift μ_x by 0.1σ	Zero covariance	Multiply (co)variances by 10	Divide (co)variances by 10	Mixture of Two Gaussians 1	Mixture of Two Gaussians 2
Wasserstein Significance								
$\text{FGD}_\infty \times 10^3$ Significance								
MMD Significance								
Precision Significance								
Recall Significance								
Diversity Significance								
Coverage Significance								

Most sensitive metric per distribution in bold

RESULTS

Metric	Truth	Shift μ_x by 1σ	Shift μ_x by 0.1σ	Zero covariance	Multiply (co)variances by 10	Divide (co)variances by 10	Mixture of Two Gaussians 1	Mixture of Two Gaussians 2
Wasserstein	0.016 ± 0.004	1.14 ± 0.02	0.043 ± 0.008	0.077 ± 0.006	9.8 ± 0.1	0.97 ± 0.01	$\mathbf{0.036 \pm 0.003}$	0.191 ± 0.005
Significance		284 ± 6	7 ± 1	16 ± 1	2460 ± 30	241 ± 3	$\mathbf{5.2 \pm 0.5}$	44 ± 1
FGD_∞ $\times 10^3$	0.27 ± 0.08	$\mathbf{1002 \pm 4}$	$\mathbf{11.5 \pm 0.5}$	$\mathbf{28.4 \pm 0.5}$	9400 ± 20	$\mathbf{941 \pm 2}$	0.4 ± 0.1	0.21 ± 0.07
Significance		$\mathbf{11960 \pm 40}$	$\mathbf{134 \pm 6}$	$\mathbf{336 \pm 6}$	112300 ± 200	$\mathbf{11230 \pm 20}$	1.3 ± 0.4	0
MMD	0.01 ± 0.02	16.4 ± 0.9	0.07 ± 0.04	0.40 ± 0.08	$\mathbf{19}\text{k} \pm \mathbf{1}\text{k}$	4.3 ± 0.1	0.06 ± 0.02	0.35 ± 0.03
Significance		790 ± 40	3 ± 2	19 ± 4	$\mathbf{920}\text{k} \pm \mathbf{70}\text{k}$	204 ± 6	2.3 ± 0.8	16 ± 1
Precision	0.972 ± 0.005	0.91 ± 0.01	0.976 ± 0.004	0.969 ± 0.006	0.34 ± 0.01	1.0 ± 0.0	0.975 ± 0.003	0.998 ± 0.001
Significance		12.2 ± 0.1	0	0.440 ± 0.003	119 ± 4	0	0	0
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.998 ± 0.001	0.998 ± 0.001	0.58 ± 0.02	0.996 ± 0.001	0.997 ± 0.001
Significance		5.38 ± 0.02	0.227 ± 0.000	0	0	420 ± 10	0.762 ± 0.001	0
Diversity	0.979 ± 0.005	0.969 ± 0.007	0.980 ± 0.005	0.977 ± 0.006	0.486 ± 0.007	0.98 ± 0.01	0.981 ± 0.007	0.98 ± 0.01
Significance		2.11 ± 0.02	0	0.335 ± 0.002	109 ± 2	0	0	0.654 ± 0.007
Coverage	0.946 ± 0.004	0.791 ± 0.008	0.944 ± 0.002	0.939 ± 0.002	0.580 ± 0.003	0.367 ± 0.004	0.942 ± 0.003	$\mathbf{0.717 \pm 0.003}$
Significance		43.8 ± 0.4	0.493 ± 0.001	2.047 ± 0.004	103.6 ± 0.6	164 ± 2	1.094 ± 0.004	$\mathbf{64.9 \pm 0.3}$

- Wasserstein, $\mathbf{FGD}_\infty$, MMD find all alternatives discrepant, except $\mathbf{FGD}_\infty$ on mixtures

Most sensitive metric per distribution in bold

RESULTS

Metric	Truth	Shift μ_x by 1σ	Shift μ_x by 0.1σ	Zero covariance	Multiply (co)variances by 10	Divide (co)variances by 10	Mixture of Two Gaussians 1	Mixture of Two Gaussians 2
Wasserstein	0.016 ± 0.004	1.14 ± 0.02	0.043 ± 0.008	0.077 ± 0.006	9.8 ± 0.1	0.97 ± 0.01	$\mathbf{0.036 \pm 0.003}$	0.191 ± 0.005
Significance		284 ± 6	7 ± 1	16 ± 1	2460 ± 30	241 ± 3	$\mathbf{5.2 \pm 0.5}$	44 ± 1
FGD_∞ $\times 10^3$	0.27 ± 0.08	$\mathbf{1002 \pm 4}$	$\mathbf{11.5 \pm 0.5}$	$\mathbf{28.4 \pm 0.5}$	9400 ± 20	$\mathbf{941 \pm 2}$	0.4 ± 0.1	0.21 ± 0.07
Significance		$\mathbf{11960 \pm 40}$	$\mathbf{134 \pm 6}$	$\mathbf{336 \pm 6}$	112300 ± 200	$\mathbf{11230 \pm 20}$	1.3 ± 0.4	0
MMD	0.01 ± 0.02	16.4 ± 0.9	0.07 ± 0.04	0.40 ± 0.08	$\mathbf{19k \pm 1k}$	4.3 ± 0.1	0.06 ± 0.02	0.35 ± 0.03
Significance		790 ± 40	3 ± 2	19 ± 4	$\mathbf{920k \pm 70k}$	204 ± 6	2.3 ± 0.8	16 ± 1
Precision	0.972 ± 0.005	0.91 ± 0.01	0.976 ± 0.004	0.969 ± 0.006	0.34 ± 0.01	1.0 ± 0.0	0.975 ± 0.003	0.998 ± 0.001
Significance		12.2 ± 0.1	0	0.440 ± 0.003	119 ± 4	0	0	0
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.998 ± 0.001	0.998 ± 0.001	0.58 ± 0.02	0.996 ± 0.001	0.997 ± 0.001
Significance		5.38 ± 0.02	0.227 ± 0.000	0	0	420 ± 10	0.762 ± 0.001	0
Diversity	0.979 ± 0.005	0.969 ± 0.007	0.980 ± 0.005	0.977 ± 0.006	0.486 ± 0.007	0.98 ± 0.01	0.981 ± 0.007	0.98 ± 0.01
Significance		2.11 ± 0.02	0	0.335 ± 0.002	109 ± 2	0	0	0.654 ± 0.007
Coverage	0.946 ± 0.004	0.791 ± 0.008	0.944 ± 0.002	0.939 ± 0.002	0.580 ± 0.003	0.367 ± 0.004	0.942 ± 0.003	$\mathbf{0.717 \pm 0.003}$
Significance		43.8 ± 0.4	0.493 ± 0.001	2.047 ± 0.004	103.6 ± 0.6	164 ± 2	1.094 ± 0.004	$\mathbf{64.9 \pm 0.3}$

- Wasserstein, $\mathbf{FGD}_\infty$, MMD find all alternatives discrepant, except $\mathbf{FGD}_\infty$ on mixtures
- $\mathbf{FGD}_\infty$ generally the most sensitive otherwise, but misses shape distortions

Most sensitive metric per distribution in bold

RESULTS

Metric	Truth	Shift μ_x by 1σ	Shift μ_x by 0.1σ	Zero covariance	Multiply (co)variances by 10	Divide (co)variances by 10	Mixture of Two Gaussians 1	Mixture of Two Gaussians 2
Wasserstein	0.016 ± 0.004	1.14 ± 0.02	0.043 ± 0.008	0.077 ± 0.006	9.8 ± 0.1	0.97 ± 0.01	$\mathbf{0.036 \pm 0.003}$	0.191 ± 0.005
Significance		284 ± 6	7 ± 1	16 ± 1	2460 ± 30	241 ± 3	$\mathbf{5.2 \pm 0.5}$	44 ± 1
$\text{FGD}_\infty \times 10^3$	0.27 ± 0.08	$\mathbf{1002 \pm 4}$	$\mathbf{11.5 \pm 0.5}$	$\mathbf{28.4 \pm 0.5}$	9400 ± 20	$\mathbf{941 \pm 2}$	0.4 ± 0.1	0.21 ± 0.07
Significance		$\mathbf{11960 \pm 40}$	$\mathbf{134 \pm 6}$	$\mathbf{336 \pm 6}$	112300 ± 200	$\mathbf{11230 \pm 20}$	1.3 ± 0.4	0
MMD	0.01 ± 0.02	16.4 ± 0.9	0.07 ± 0.04	0.40 ± 0.08	$\mathbf{19k \pm 1k}$	4.3 ± 0.1	0.06 ± 0.02	0.35 ± 0.03
Significance		790 ± 40	3 ± 2	19 ± 4	$\mathbf{920k \pm 70k}$	204 ± 6	2.3 ± 0.8	16 ± 1
Precision	0.972 ± 0.005	0.91 ± 0.01	0.976 ± 0.004	0.969 ± 0.006	0.34 ± 0.01	1.0 ± 0.0	0.975 ± 0.003	0.998 ± 0.001
Significance		12.2 ± 0.1	0	0.440 ± 0.003	119 ± 4	0	0	0
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.998 ± 0.001	0.998 ± 0.001	0.58 ± 0.02	0.996 ± 0.001	0.997 ± 0.001
Significance		5.38 ± 0.02	0.227 ± 0.000	0	0	420 ± 10	0.762 ± 0.001	0
Diversity	0.979 ± 0.005	0.969 ± 0.007	0.980 ± 0.005	0.977 ± 0.006	0.486 ± 0.007	0.98 ± 0.01	0.981 ± 0.007	0.98 ± 0.01
Significance		2.11 ± 0.02	0	0.335 ± 0.002	109 ± 2	0	0	0.654 ± 0.007
Coverage	0.946 ± 0.004	0.791 ± 0.008	0.944 ± 0.002	0.939 ± 0.002	0.580 ± 0.003	0.367 ± 0.004	0.942 ± 0.003	$\mathbf{0.717 \pm 0.003}$
Significance		43.8 ± 0.4	0.493 ± 0.001	2.047 ± 0.004	103.6 ± 0.6	164 ± 2	1.094 ± 0.004	$\mathbf{64.9 \pm 0.3}$

- Wasserstein, $\mathbf{FGD}_\infty$, MMD find all alternatives discrepant, except $\mathbf{FGD}_\infty$ on mixtures
- $\mathbf{FGD}_\infty$ generally the most sensitive otherwise, but misses shape distortions
- Precision and recall do their job, density and coverage give unintuitive results

Most sensitive metric per distribution in bold

RESULTS

RK et al., PRD 2023

Metric	Truth	Smeared	Shifted	Removing tail	Particle features smeared	Particle η^{rel} smeared	Particle $p_{\mathrm{T}}^{\mathrm{rel}}$ smeared	Particle $p_{\mathrm{T}}^{\mathrm{rel}}$ shifted
$W_1^M \times 10^3$	0.28 ± 0.05	2.1 ± 0.2	6.0 ± 0.3	0.6 ± 0.2	1.7 ± 0.2	0.9 ± 0.3	0.5 ± 0.2	5.8 ± 0.2
Sign.	—	37 ± 3	114 ± 6	7 ± 2	28 ± 3	12 ± 4	4 ± 1	111 ± 3
Wasserstein EFP	0.02 ± 0.01	0.09 ± 0.05	0.10 ± 0.02	0.016 ± 0.007	0.19 ± 0.08	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.02
Sign.	—	6 ± 4	7 ± 1	0.06 ± 0.02	14 ± 6	0.8 ± 0.4	0.9 ± 0.6	4 ± 1
FGD_∞ EFP $\times 10^3$	0.08 ± 0.03	$\mathbf{20 \pm 1}$	$\mathbf{26.6 \pm 0.9}$	$\mathbf{2.4 \pm 0.1}$	21 ± 2	$\mathbf{3.6 \pm 0.3}$	2.3 ± 0.2	29.1 ± 0.4
Sign.	—	$\mathbf{580 \pm 30}$	$\mathbf{760 \pm 20}$	$\mathbf{66 \pm 4}$	610 ± 40	$\mathbf{103 \pm 8}$	64 ± 4	830 ± 10
MMD EFP $\times 10^3$	-0.006 ± 0.005	0.17 ± 0.06	0.9 ± 0.1	0.03 ± 0.02	0.35 ± 0.09	0.08 ± 0.05	0.01 ± 0.02	1.8 ± 0.1
Sign.	—	30 ± 10	170 ± 20	6 ± 4	70 ± 10	10 ± 10	3 ± 5	360 ± 20
Precision EFP	0.9 ± 0.1	0.94 ± 0.04	0.978 ± 0.005	0.88 ± 0.08	0.7 ± 0.1	0.94 ± 0.06	0.7 ± 0.1	0.79 ± 0.09
Sign.	—	0	0	0.109 ± 0.009	1.9 ± 0.3	0	2.0 ± 0.3	0.9 ± 0.1
Recall EFP	0.9 ± 0.1	0.88 ± 0.07	0.97 ± 0.01	0.92 ± 0.06	0.83 ± 0.05	0.92 ± 0.07	0.8 ± 0.1	0.8 ± 0.1
Sign.	—	0.16 ± 0.01	0	0	0.58 ± 0.04	0	0.8 ± 0.1	1.1 ± 0.2
Wasserstein PN	1.65 ± 0.06	1.7 ± 0.1	2.4 ± 0.4	1.71 ± 0.08	4.5 ± 0.1	1.79 ± 0.05	4.0 ± 0.4	7.6 ± 0.2
Sign.	—	0.84 ± 0.05	12 ± 2	0.97 ± 0.05	45 ± 1	2.26 ± 0.06	37 ± 3	95 ± 3
FGD_∞ PN $\times 10^3$	0.6 ± 0.4	37 ± 2	202 ± 4	4.3 ± 0.4	$\mathbf{1220 \pm 10}$	20 ± 1	$\mathbf{1230 \pm 10}$	$\mathbf{3630 \pm 10}$
Sign.	—	98 ± 4	540 ± 0	9.8 ± 0.9	$\mathbf{3320 \pm 20}$	51 ± 3	$\mathbf{3340 \pm 30}$	$\mathbf{9870 \pm 30}$
MMD PN $\times 10^3$	-2 ± 2	4 ± 8	80 ± 10	-1 ± 4	500 ± 100	3 ± 2	560 ± 60	1100 ± 40
Sign.	—	3 ± 6	40 ± 10	0 ± 3	280 ± 70	3 ± 2	310 ± 30	610 ± 20
Precision PN	0.68 ± 0.07	0.64 ± 0.04	0.71 ± 0.06	0.73 ± 0.03	0.09 ± 0.04	0.75 ± 0.08	0.08 ± 0.04	0.39 ± 0.08
Sign.	—	0.57 ± 0.04	0	0	8 ± 4	0	8 ± 5	4.0 ± 0.8
Recall PN	0.70 ± 0.05	0.61 ± 0.04	0.61 ± 0.08	0.73 ± 0.06	0.014 ± 0.009	0.7 ± 0.1	0.01 ± 0.01	0.57 ± 0.09
Sign.	—	1.8 ± 0.1	1.8 ± 0.2	0	14 ± 9	0	10 ± 10	2.6 ± 0.4
Classifier LLF AUC	0.50	0.52	0.54	0.50	0.97	0.81	0.93	0.99
Classifier HLF AUC	0.50	0.53	0.55	0.50	0.84	0.64	0.74	0.92

- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing

Most sensitive metric per distribution in bold

RESULTS

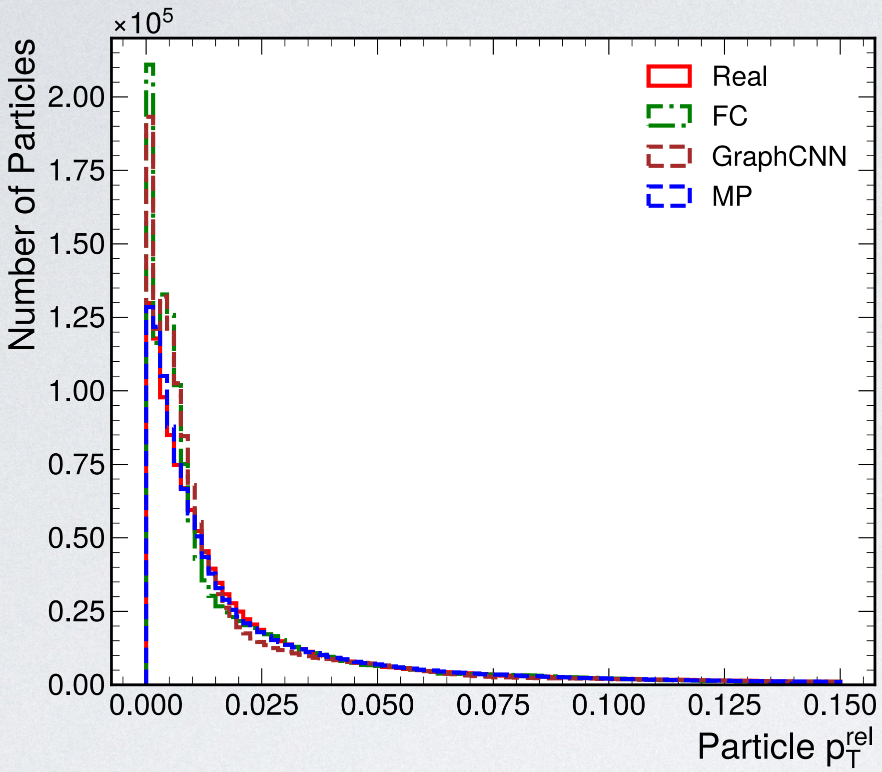
RK et al., PRD 2023

Metric	Truth	Smeared	Shifted	Removing tail	Particle features smeared	Particle η^{rel} smeared	Particle $p_{\mathrm{T}}^{\mathrm{rel}}$ smeared	Particle $p_{\mathrm{T}}^{\mathrm{rel}}$ shifted
$W_1^M \times 10^3$	0.28 ± 0.05	2.1 ± 0.2	6.0 ± 0.3	0.6 ± 0.2	1.7 ± 0.2	0.9 ± 0.3	0.5 ± 0.2	5.8 ± 0.2
Sign.	—	37 ± 3	114 ± 6	7 ± 2	28 ± 3	12 ± 4	4 ± 1	111 ± 3
Wasserstein EFP	0.02 ± 0.01	0.09 ± 0.05	0.10 ± 0.02	0.016 ± 0.007	0.19 ± 0.08	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.02
Sign.	—	6 ± 4	7 ± 1	0.06 ± 0.02	14 ± 6	0.8 ± 0.4	0.9 ± 0.6	4 ± 1
FGD_∞ EFP $\times 10^3$	0.08 ± 0.03	$\mathbf{20 \pm 1}$	$\mathbf{26.6 \pm 0.9}$	$\mathbf{2.4 \pm 0.1}$	21 ± 2	$\mathbf{3.6 \pm 0.3}$	2.3 ± 0.2	29.1 ± 0.4
Sign.	—	$\mathbf{580 \pm 30}$	$\mathbf{760 \pm 20}$	$\mathbf{66 \pm 4}$	610 ± 40	$\mathbf{103 \pm 8}$	64 ± 4	830 ± 10
MMD EFP $\times 10^3$	-0.006 ± 0.005	0.17 ± 0.06	0.9 ± 0.1	0.03 ± 0.02	0.35 ± 0.09	0.08 ± 0.05	0.01 ± 0.02	1.8 ± 0.1
Sign.	—	30 ± 10	170 ± 20	6 ± 4	70 ± 10	10 ± 10	3 ± 5	360 ± 20
Precision EFP	0.9 ± 0.1	0.94 ± 0.04	0.978 ± 0.005	0.88 ± 0.08	0.7 ± 0.1	0.94 ± 0.06	0.7 ± 0.1	0.79 ± 0.09
Sign.	—	0	0	0.109 ± 0.009	1.9 ± 0.3	0	2.0 ± 0.3	0.9 ± 0.1
Recall EFP	0.9 ± 0.1	0.88 ± 0.07	0.97 ± 0.01	0.92 ± 0.06	0.83 ± 0.05	0.92 ± 0.07	0.8 ± 0.1	0.8 ± 0.1
Sign.	—	0.16 ± 0.01	0	0	0.58 ± 0.04	0	0.8 ± 0.1	1.1 ± 0.2
Wasserstein PN	1.65 ± 0.06	1.7 ± 0.1	2.4 ± 0.4	1.71 ± 0.08	4.5 ± 0.1	1.79 ± 0.05	4.0 ± 0.4	7.6 ± 0.2
Sign.	—	0.84 ± 0.05	12 ± 2	0.97 ± 0.05	45 ± 1	2.26 ± 0.06	37 ± 3	95 ± 3
FGD_∞ PN $\times 10^3$	0.6 ± 0.4	37 ± 2	202 ± 4	4.3 ± 0.4	$\mathbf{1220 \pm 10}$	20 ± 1	$\mathbf{1230 \pm 10}$	$\mathbf{3630 \pm 10}$
Sign.	—	98 ± 4	540 ± 0	9.8 ± 0.9	$\mathbf{3320 \pm 20}$	51 ± 3	$\mathbf{3340 \pm 30}$	$\mathbf{9870 \pm 30}$
MMD PN $\times 10^3$	-2 ± 2	4 ± 8	80 ± 10	-1 ± 4	500 ± 100	3 ± 2	560 ± 60	1100 ± 40
Sign.	—	3 ± 6	40 ± 10	0 ± 3	280 ± 70	3 ± 2	310 ± 30	610 ± 20
Precision PN	0.68 ± 0.07	0.64 ± 0.04	0.71 ± 0.06	0.73 ± 0.03	0.09 ± 0.04	0.75 ± 0.08	0.08 ± 0.04	0.39 ± 0.08
Sign.	—	0.57 ± 0.04	0	0	8 ± 4	0	8 ± 5	4.0 ± 0.8
Recall PN	0.70 ± 0.05	0.61 ± 0.04	0.61 ± 0.08	0.73 ± 0.06	0.014 ± 0.009	0.7 ± 0.1	0.01 ± 0.01	0.57 ± 0.09
Sign.	—	1.8 ± 0.1	1.8 ± 0.2	0	14 ± 9	0	10 ± 10	2.6 ± 0.4
Classifier LLF AUC	0.50	0.52	0.54	0.50	0.97	0.81	0.93	0.99
Classifier HLF AUC	0.50	0.53	0.55	0.50	0.84	0.64	0.74	0.92

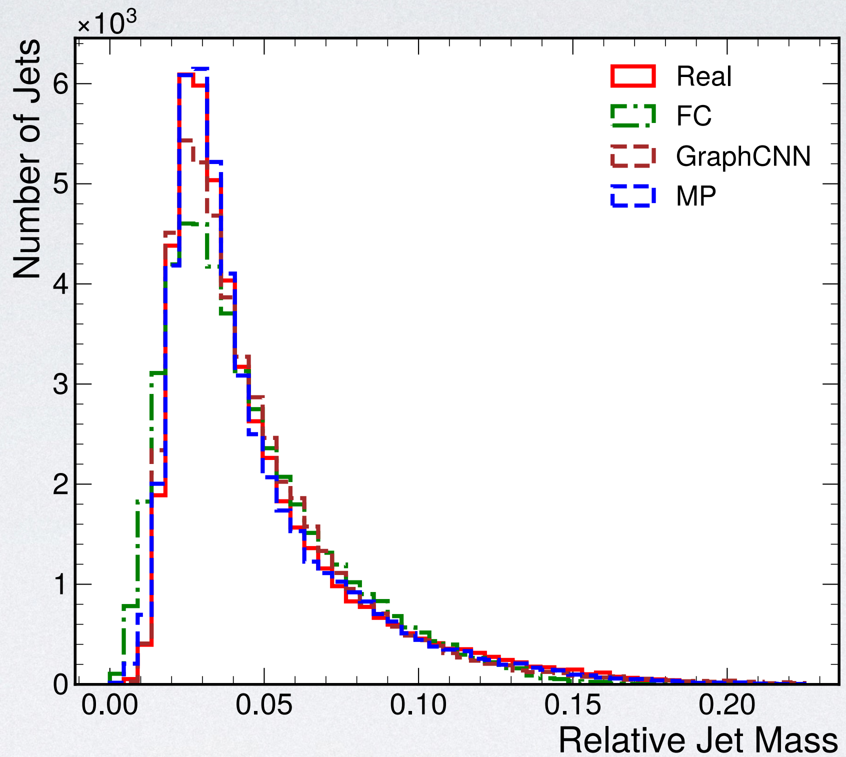
- Classifiers, low-level (LLF) and high-level features (HLF), identify particle feature distortions but miss distribution-level discrepancies

RESULTS: LIGHT QUARK JETS

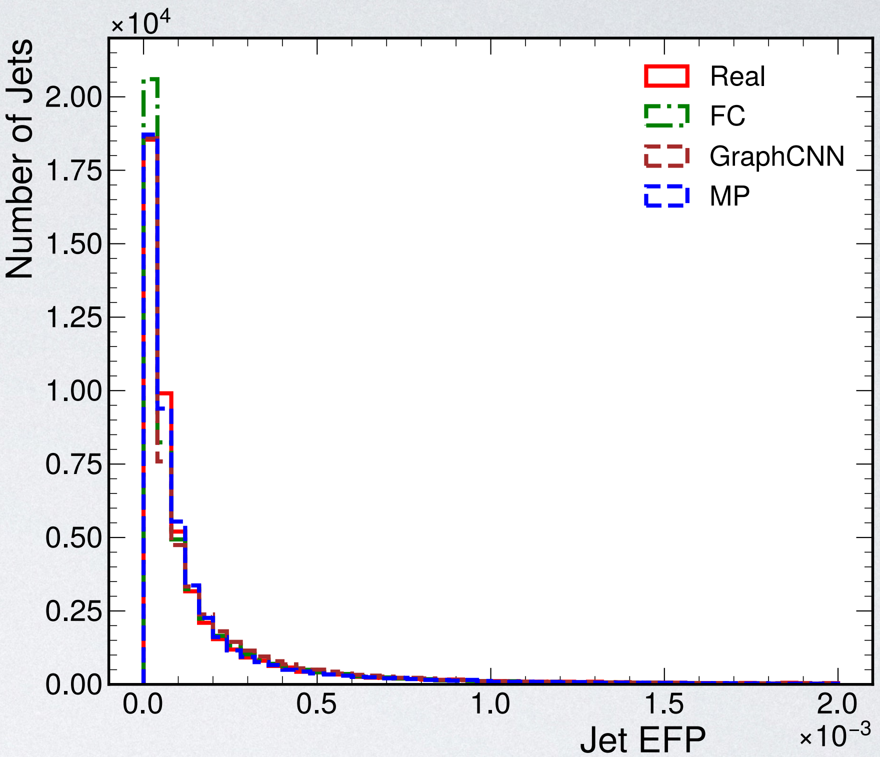
Sample feature distributions, with our MPGAN compared to FC and GraphCNN generators + PointNet discriminators



Real vs real
W1-P = (0.5 ± 0.1) × 10^{-3}



Real vs real
W1-M = (0.5 ± 0.1) × 10^{-3}



Real vs real
W1-EFP = (0.46 ± 0.04) × 10^{-5}

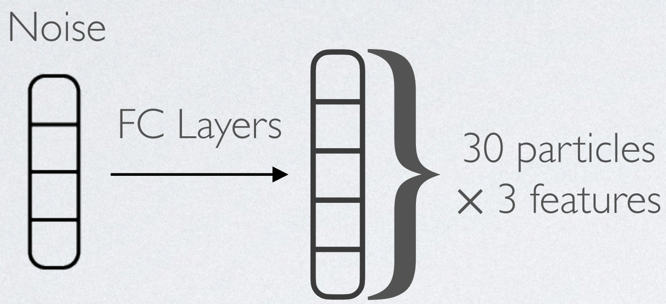
Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND	Coverage	MMD
FC	PointNet	1.5 ± 0.1	2.9 ± 0.2	2 ± 1	22k	0.36	0.026
GraphCNN	PointNet	3.9 ± 0.2	4.2 ± 1.6	20M ± 10M	19k	0.37	0.031
MP	MP	2.1 ± 0.1	0.6 ± 0.1	0.9 ± 0.4	2.4	0.54	0.026
MP	PointNet	22.0 ± 0.1	3.2 ± 0.2	5 ± 2	3.6k	0.22	0.035

- Masking strategy is successful
- MPGAN again best performing on every metric, apart from W1-P, significantly so on W1-M, W1-EFP, FPND
- Mass and ave. EFP scores all within error of the real vs real baseline

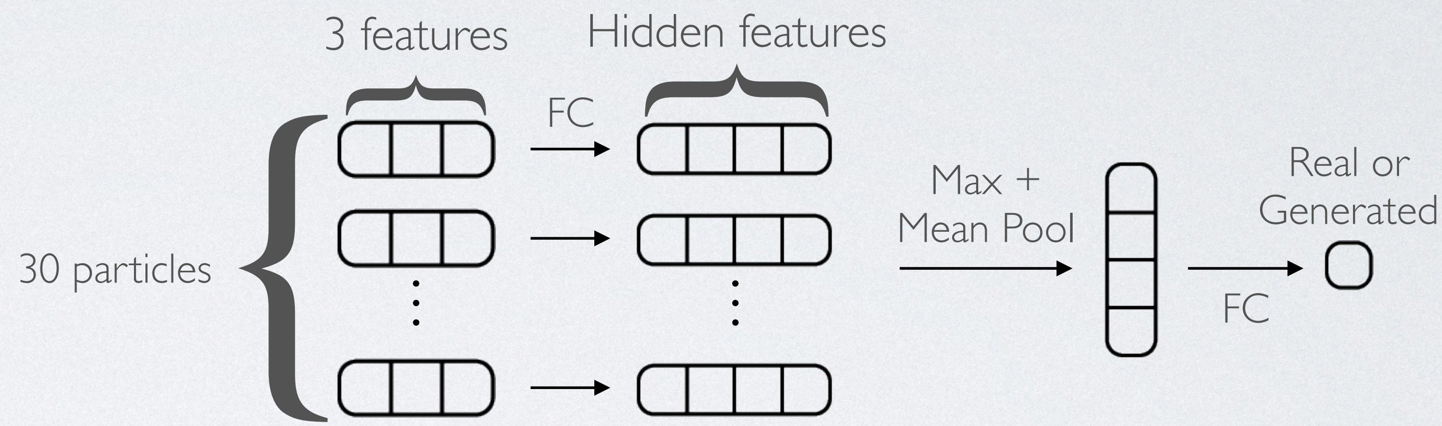
BASELINE POINT CLOUD GANS

We compare with existing point cloud GANs as baselines, two relevant architectures are:

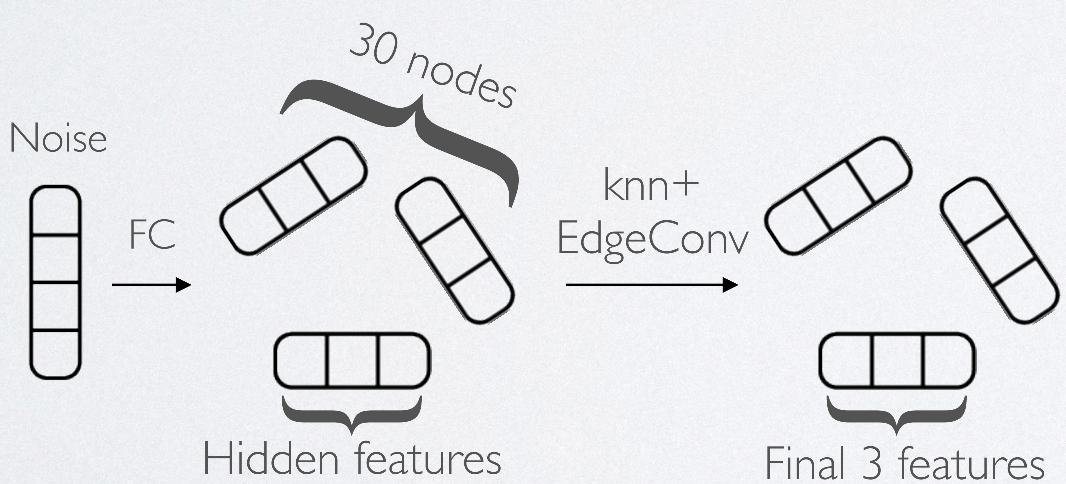
Fully Connected Generator (Achlioptas et al '18)



PointNet-Mix Discriminator is the most performant on ShapeNet, compared to FC and GraphCNN (Wang et al '21)



Graph Convolutional Generator (Valsesia et al '19)



These GANs work somewhat, but not well enough for our purpose (next slides)

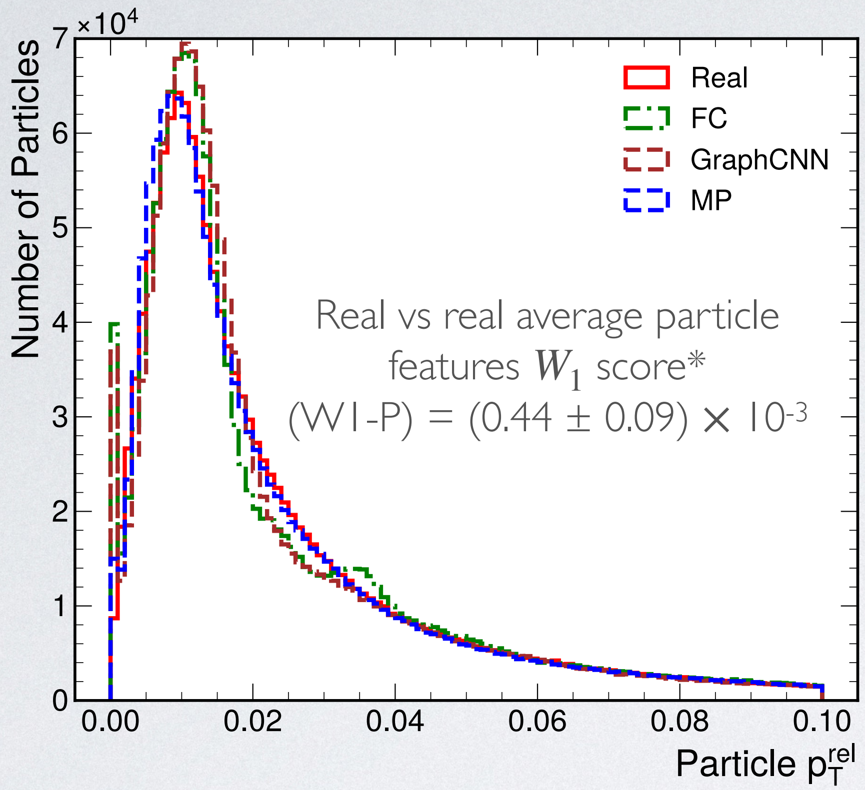
Interestingly, Wang et al find the FC generator works better than the GraphCNN (with a PointNet-Mix discriminator)

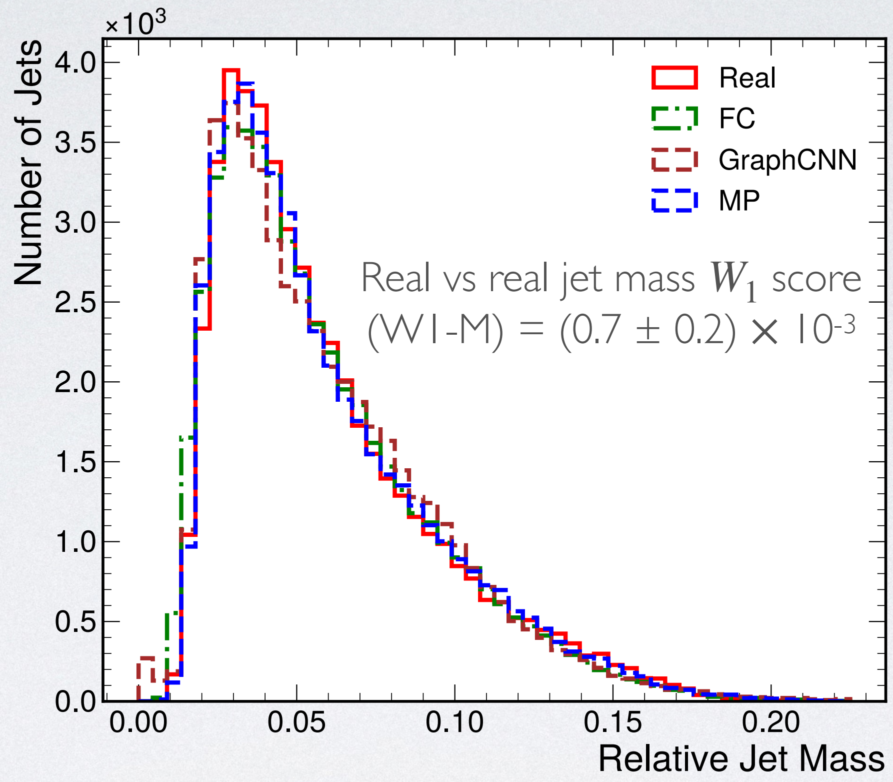
RESULTS: GLUON JETS

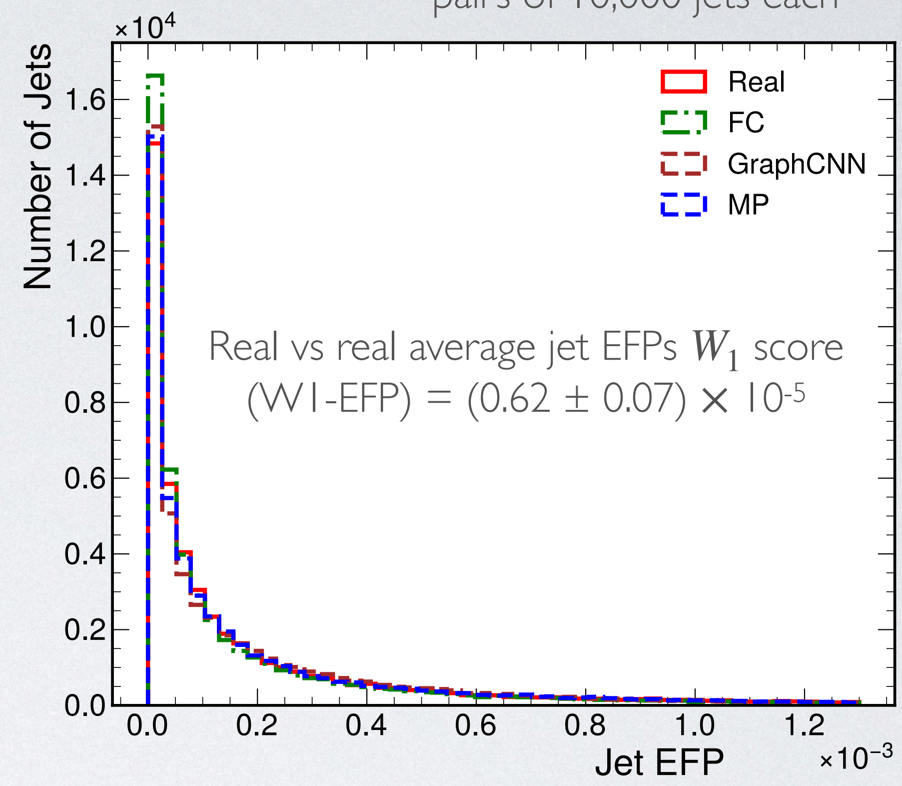
Kansal et al., ML4PS @ NeurIPS 2020
Kansal et al., NeurIPS 2021

Sample feature distributions, with MPGAN compared to baseline point cloud generators

*Mean and error over 5 sets of pairs of 10,000 jets each



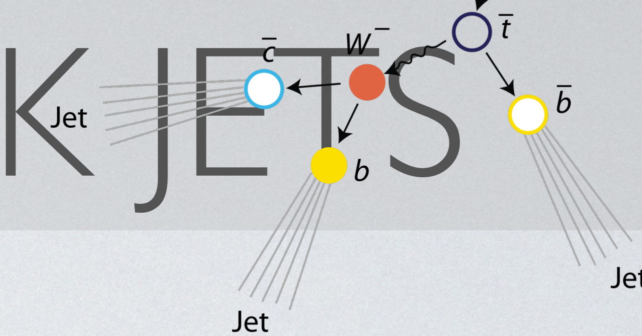


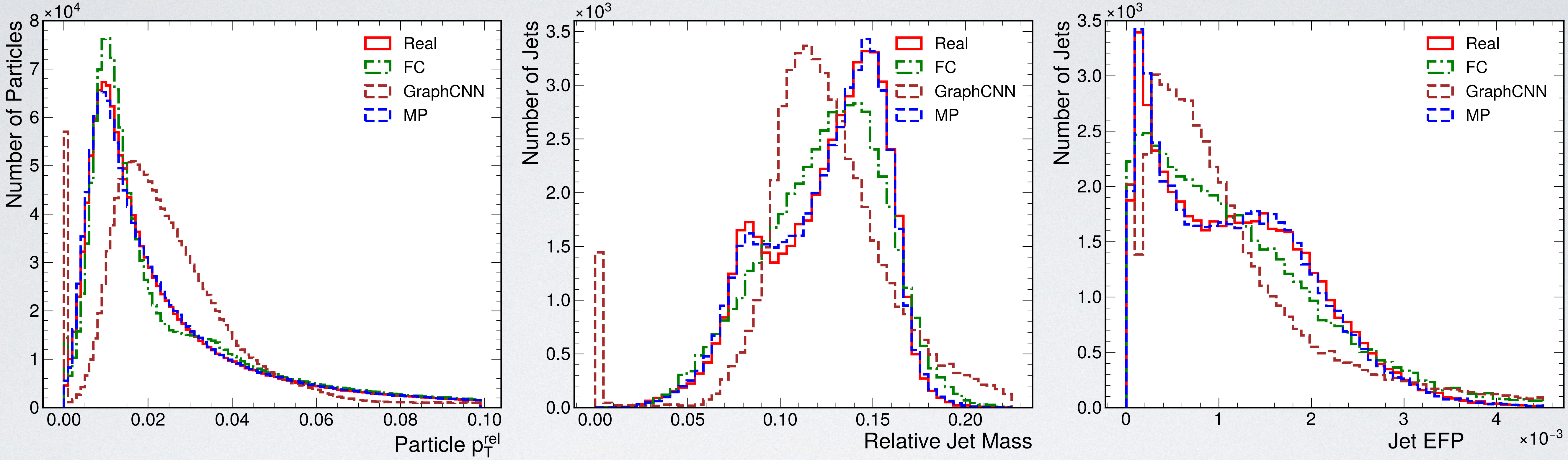


Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.3 ± 0.2	1.3 ± 0.4	1.5 ± 0.9	5.0
GraphCNN	PointNet	16 ± 6	1.9 ± 0.2	200 ± 1000	7k
MP	MP	0.9 ± 0.3	0.7 ± 0.2	0.7 ± 0.2	0.12
MP	PointNet	1.2 ± 0.4	1.3 ± 0.4	4 ± 2	18

- MPGAN generator is the best performing on every metric
- Significantly outperforms alternatives on high level feature metrics (W1-M, W1-EFP, FPND)
- Mass and ave. EFP scores are within error of the real vs real baseline ⇒ learning jet substructure correctly

RESULTS: TOP QUARK JETS





Real vs real
W1-P = (0.55 ± 0.07) × 10^{-3}

Real vs real
W1-M = (0.51 ± 0.07) × 10^{-3}

Real vs real
W1-EFP = (1.1 ± 0.1) × 10^{-5}

Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.6 ± 0.4	2.7 ± 0.1	7.7 ± 0.5	3.9
GraphCNN	PointNet	30 ± 20	11.3 ± 0.9	37 ± 2	30k
MP	MP	2.3 ± 0.3	0.6 ± 0.2	2 ± 1	0.37
MP	PointNet	1.6 ± 0.4	0.76 ± 0.08	4 ± 1	3.7

- MPGAN learns perfectly the complex bimodal jet feature distributions
- Mass and ave. EFP scores remain within error of real vs real baseline