ON THE EVALUATION OF GENERATIVE MODELS IN HEP

Raghav Kansal, Javier Duarte (UCSD)
Nadya Chernyavskaya, Maurizio Pierini (CERN)
Breno Orzari, Thiago Tomei (SPRACE)

ML4Jets
01/11/2022

- Lots of approaches in the last few years in ML for HEP simulations

- *"It is time to harvest"* - CMS ML Townhall 2022

- How do we choose and use these for HL-LHC?

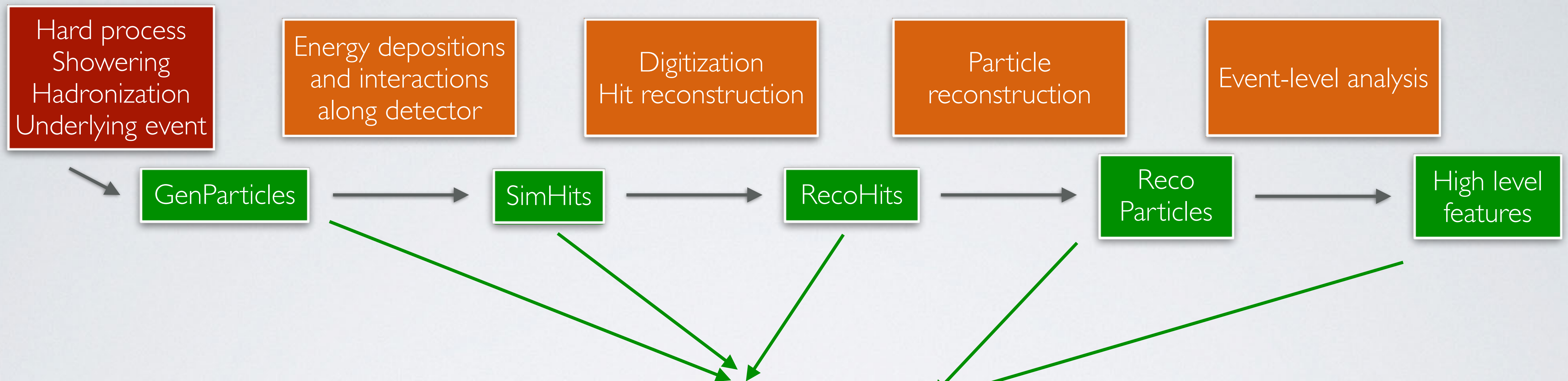
- How do we **trust** generated data?

- How do we compare generative models?

- How do we **trust** generated data? **Evaluation metrics**

- How do we compare generative models? **Evaluation metrics**

PROBLEM



- Want to quantify difference between $p_{\text{real}}(\mathbf{x})$ and $p_{\text{gen}}(\mathbf{x})$ distributions

⇒ Multivariate goodness-of-fit (g.o.f.) / two-sample test

- But no "best" g.o.f. test (Cousins 2016)
- Need to choose based on the relevant alternative hypotheses

TEST CRITERIA

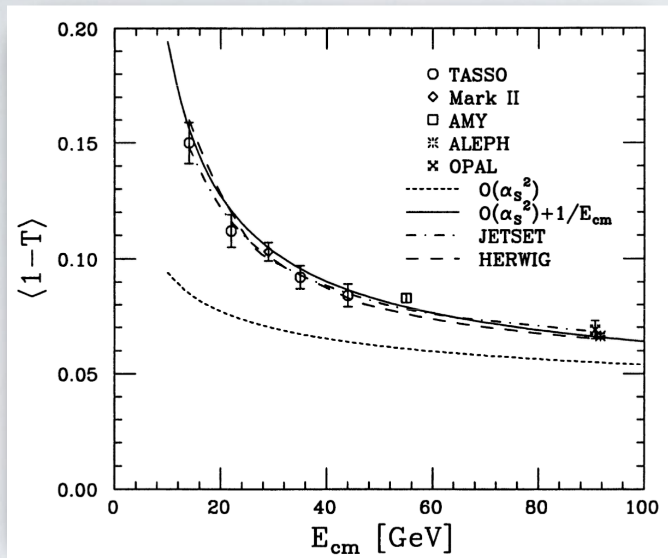
- To **trust** generated data, tests should be:
  - Sensitive to quality
  - Sensitive to diversity
  - Multivariate (for correlations & conditional generation)
  - Interpretable
- To **compare** generative models, tests should be:
  - Standardised across collaboration
  - Reproducible
  - ~Efficient

METHODS

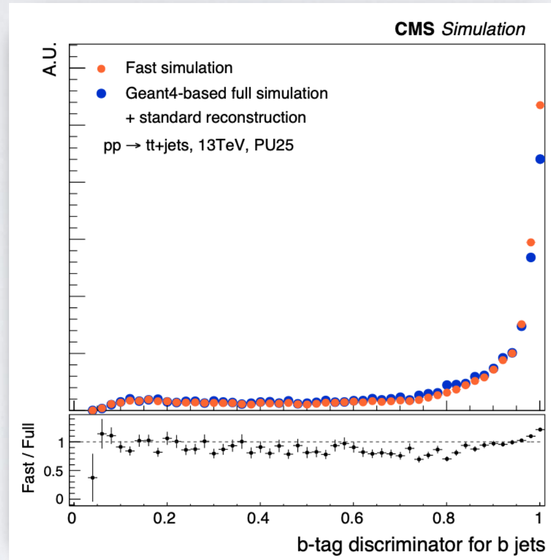
HISTOGRAMS

- Traditional method for evaluating physics simulations is to compare physical distributions

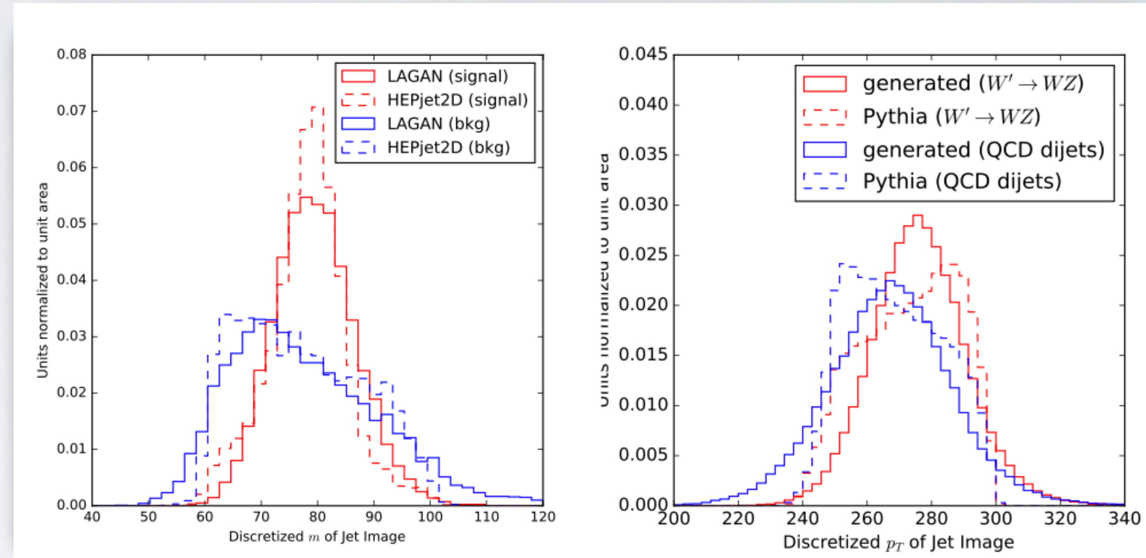
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)



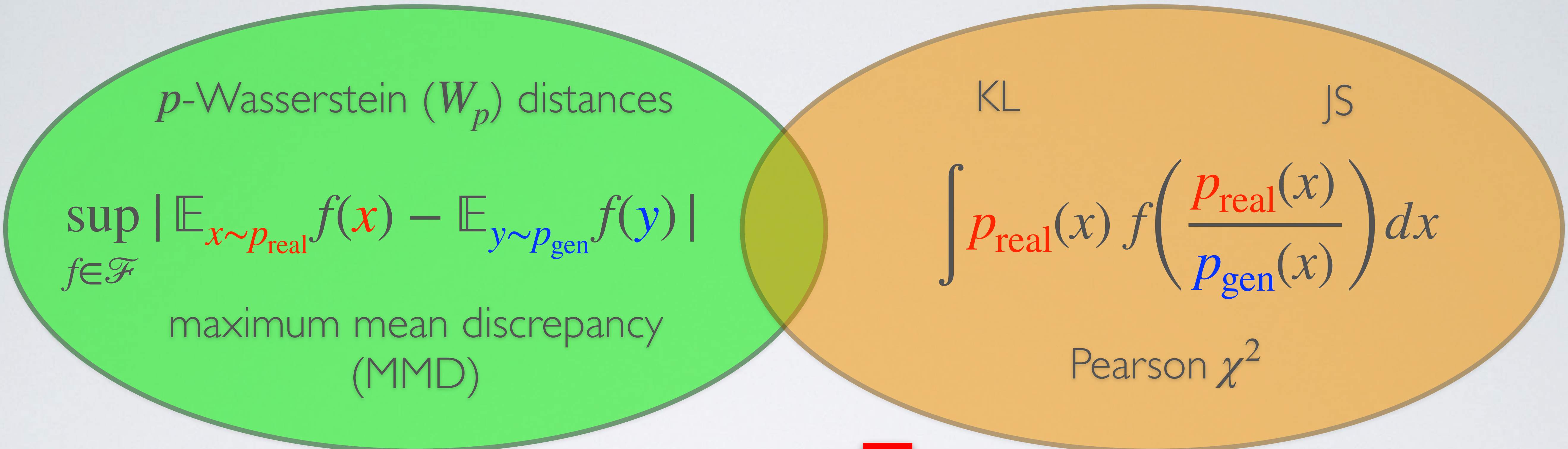
- Valuable insight into physics performance
- Should be quantified
- Cons:
    - Only 1D (curse of dimensionality for multivariate histograms)
    - Binning dependent
    - No well-defined way to aggregate scores across multiple distributions

$p_{\text{real}}(\mathbf{x})$ vs $p_{\text{gen}}(\mathbf{x})$

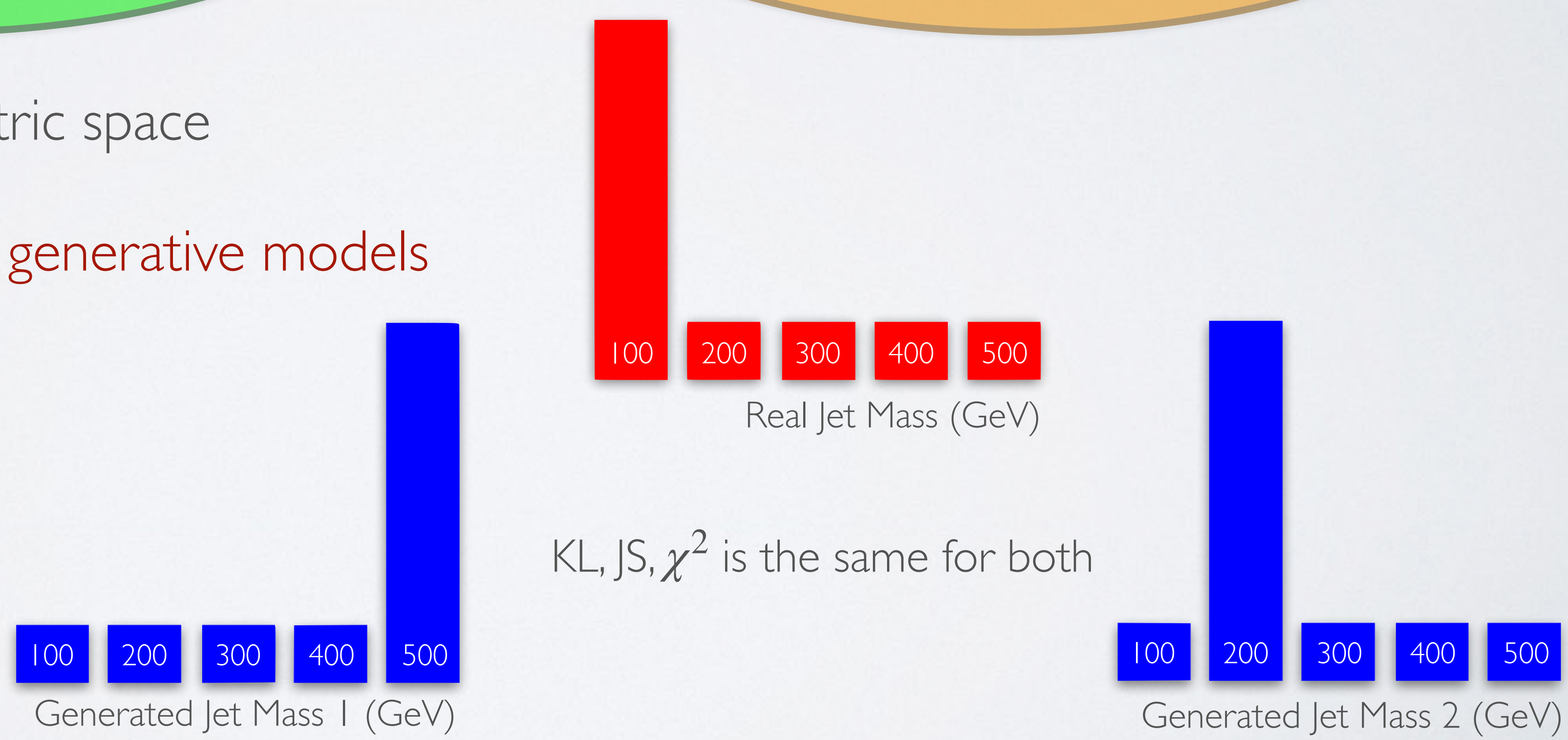
Sources 1, 2

Integral Probability Metrics $D_{\mathcal{F}}(p_{\text{real}}, p_{\text{gen}})$

f-Divergences $D_f(p_{\text{real}}, p_{\text{gen}})$



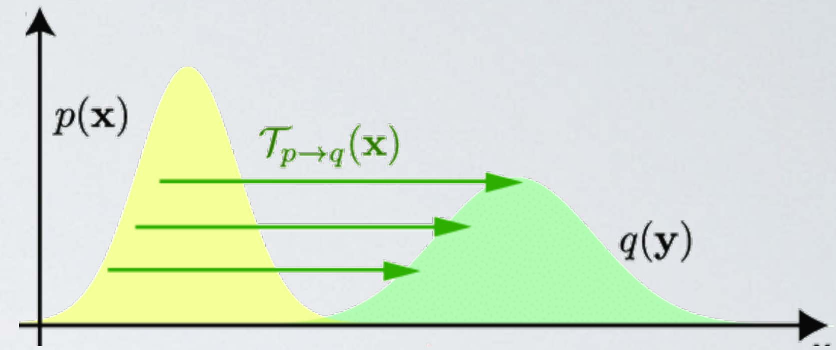
- IPMs take into account metric space
- More useful for comparing generative models



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 distance (W_1)
  ($\mathcal{F}$ is all K-Lipschitz functions)
  - Sensitive to quality, diversity; but biased and slow convergence



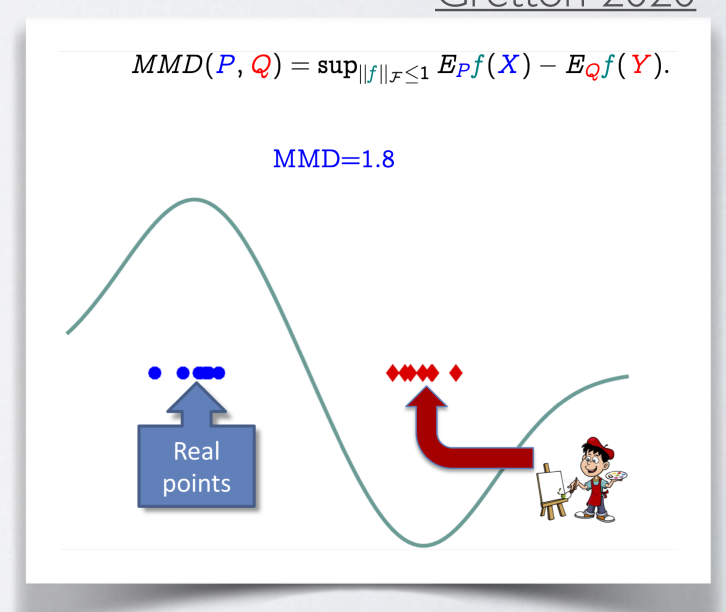
- Fréchet Gaussian distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID), efficient, sensitive to quality and diversity; but Gaussian assumption

$$\text{FGD} = \text{Frechet}(\mathcal{N}(\mu_r, \Sigma_r), \mathcal{N}(\mu_g, \Sigma_g))$$

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

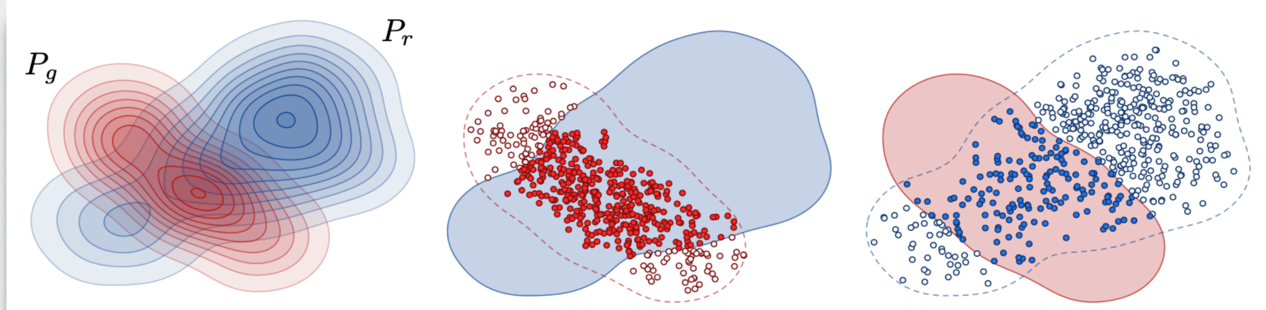
Gretton 2020

- Maximum Mean Discrepancy (MMD)
  ($\mathcal{F}$ is unit ball in reproducing Kernel Hilbert space (RKHS) for a chosen kernel $k(x, y)$)
  - Distance between embeddings of p_{real} and p_{gen} in RKHS
  - Fast, unbiased estimators, but depends on kernel



MORE METRICS

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

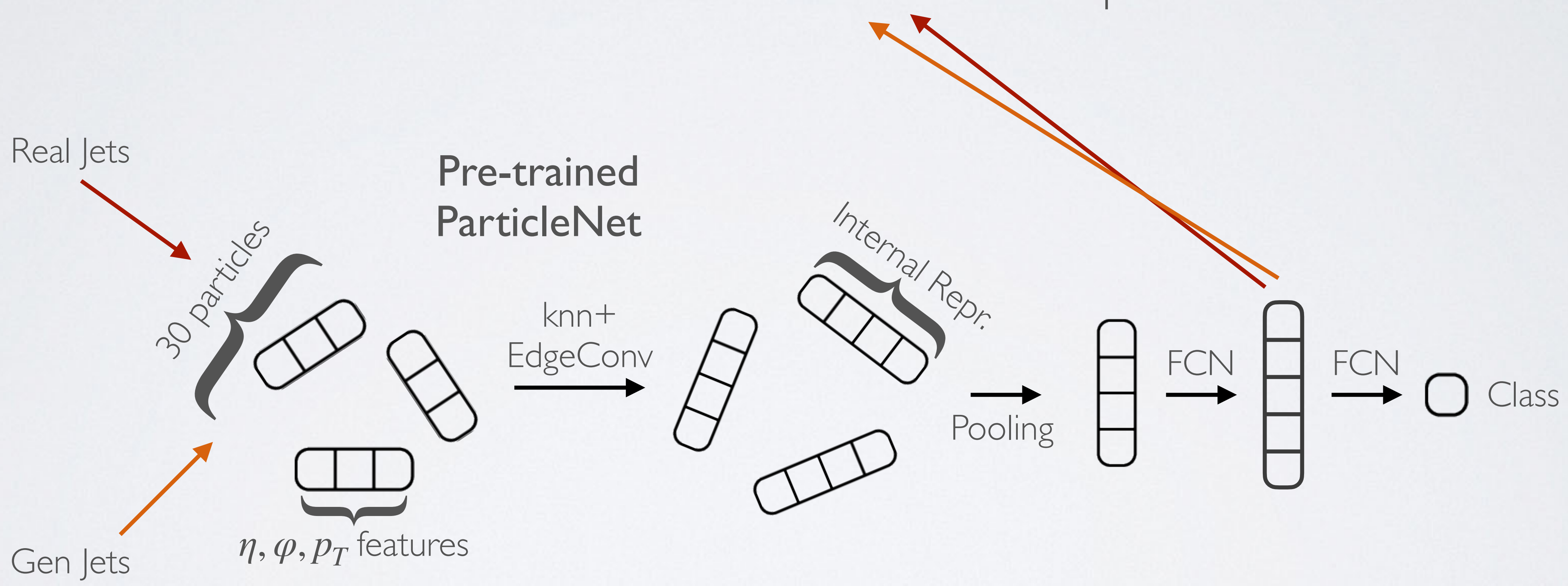


(a) Example distributions (b) Precision (c) Recall

- 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

FEATURE SELECTION

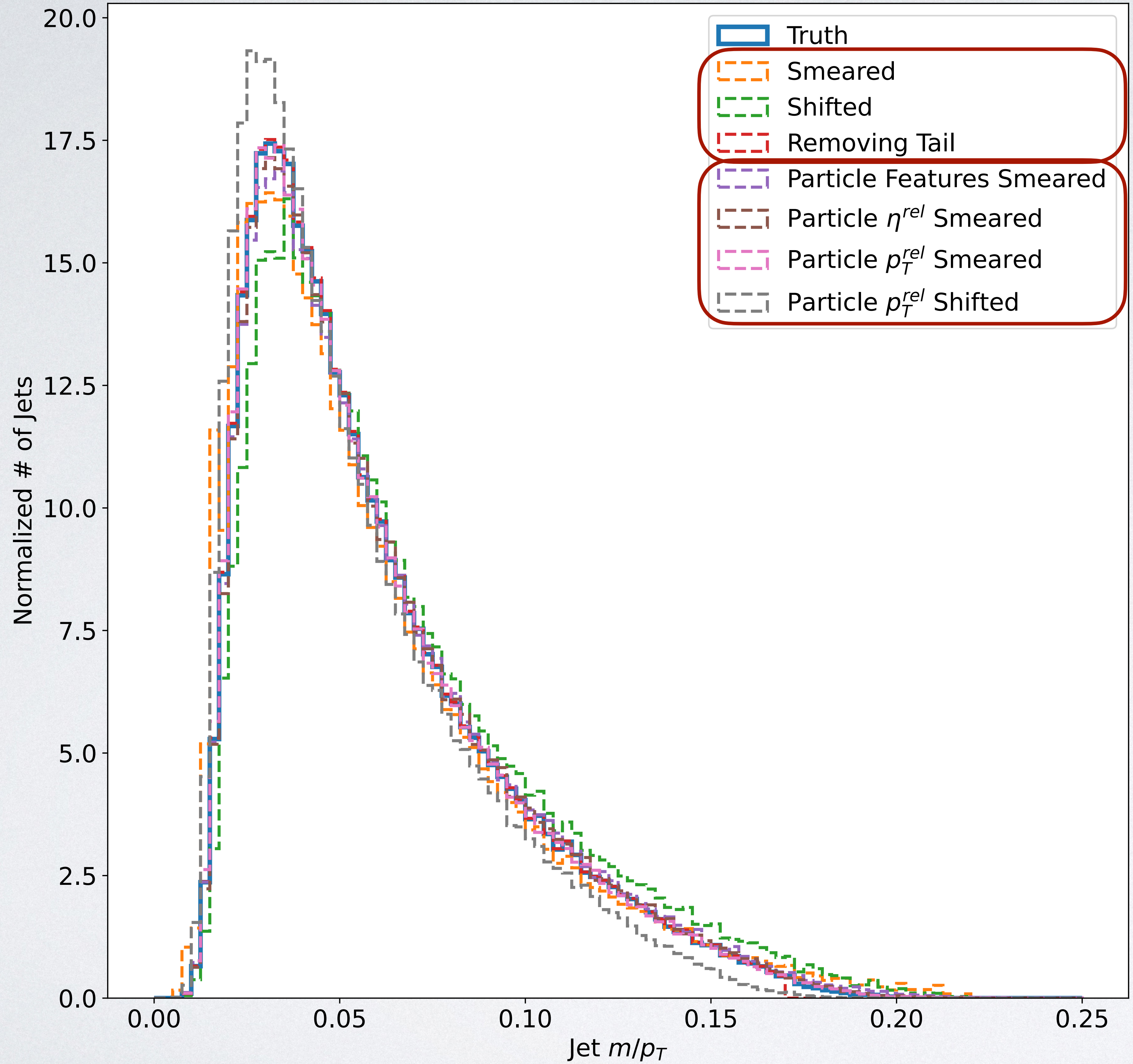
- Typically raw data (particle / hit features) is very high dimensional
- Not necessarily what we care about
- ML solution: derive lower dimensional salient features from a pre-trained classifier



- Alternative? Use physicists' hand-engineered features: jet observables, shower-shape variables

TESTS

JET DISTRIBUTIONS



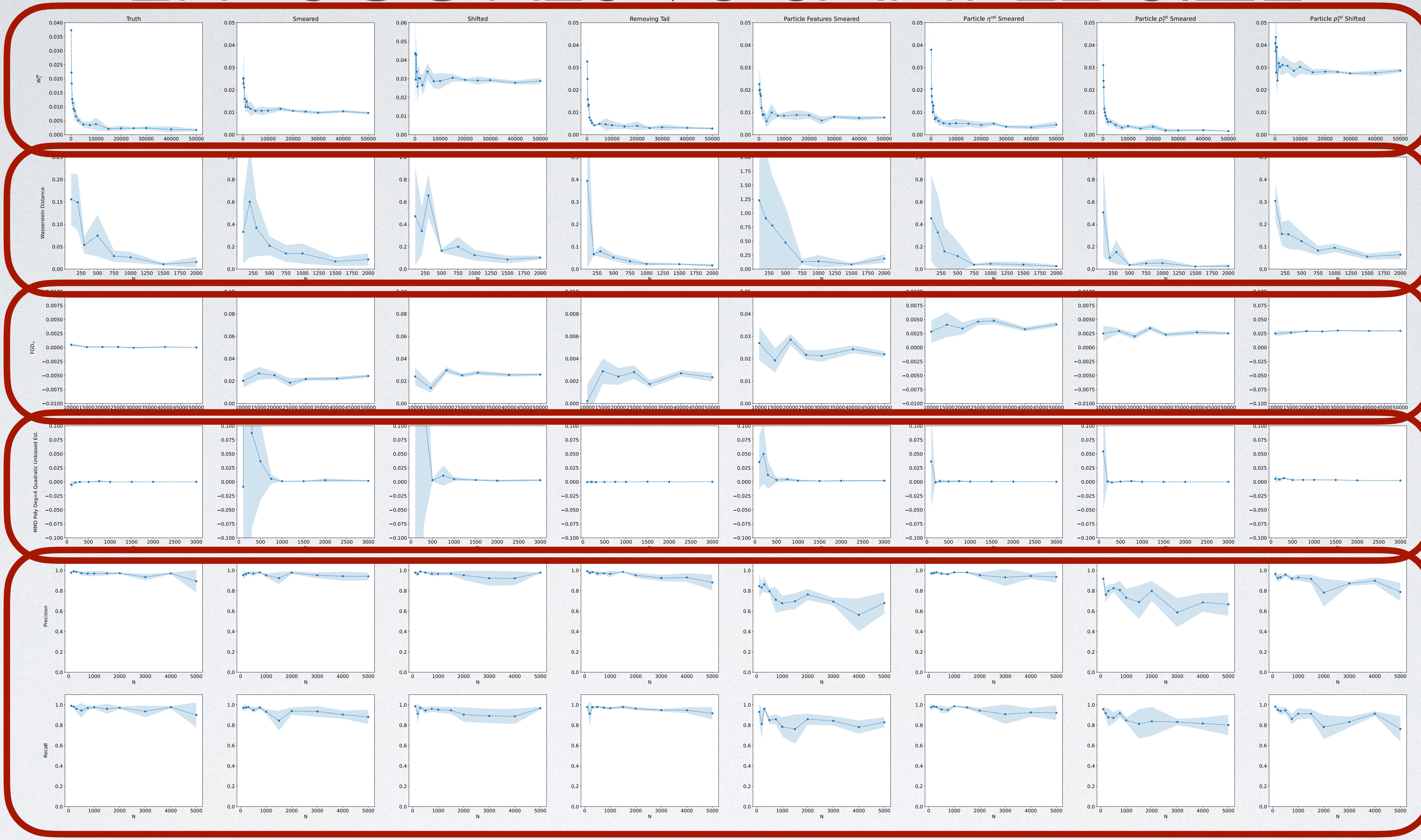
- Sample of gluon jets to test sensitivity of metrics
- We distort true distribution by:

1. Re-weighting in mass
2. Smearing/shifting particle features

- We look at sensitivity of metrics to distortions, using:

1. Energy Flow Polynomials (EFPs) (d ≤ 4)
2. ParticleNet activations

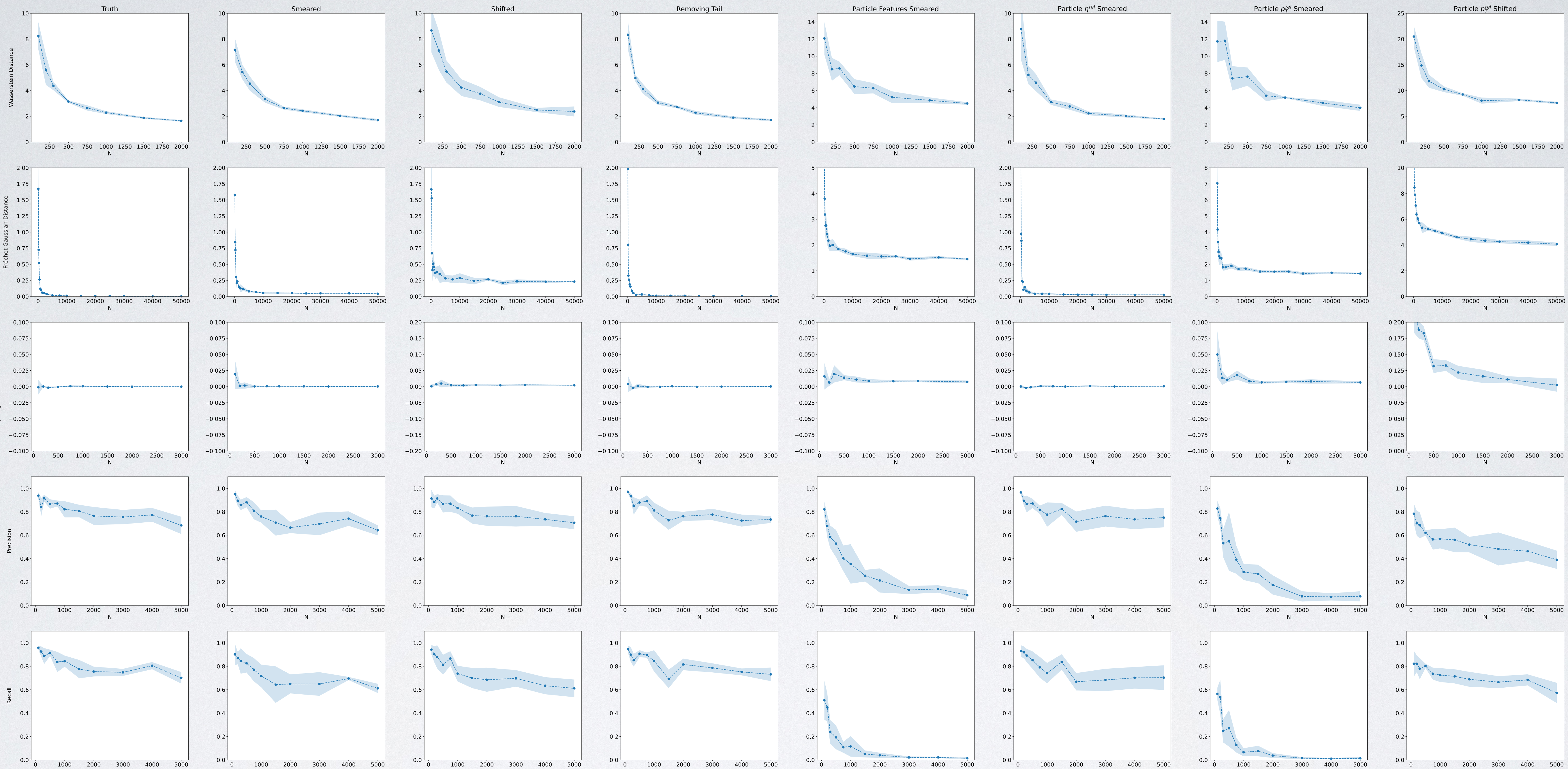
EFP SCORES VS SAMPLE SIZE



- W_1^M (looking at 1D mass distribution only) works somewhat, but not as sensitive
- Wasserstein distance is biased and slow to converge
- MMD fails completely (for all kernels tested)
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing

FGD is the most sensitive

PARTICLENET ACTIVATION SCORES



- Same conclusions overall as for EFPs
- FGD the best, MMD is not very sensitive, P&R are OK for diagnosing failure modes

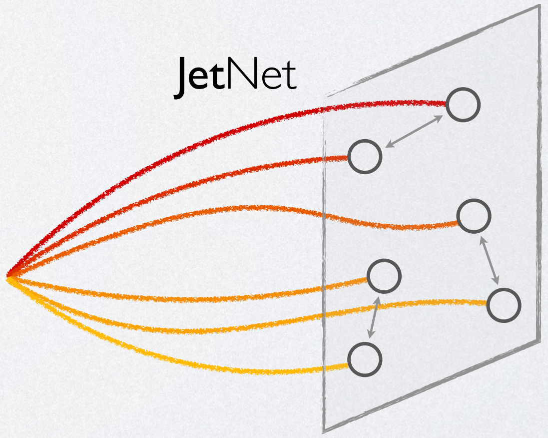
FINAL SCORES

Metric	Truth	Smeared	Shifted	Removing Tail	Particle Features Smeared	Particle η^{rel} Smeared	Particle p_T^{rel} Smeared	Particle p_T^{rel} Shifted
W_1^M	0.002 ± 0.000	0.010 ± 0.001	0.029 ± 0.002	0.003 ± 0.001	0.008 ± 0.001	0.004 ± 0.001	0.002 ± 0.000	0.029 ± 0.001
Wasserstein Distance EFP	0.016 ± 0.012	0.086 ± 0.054	0.102 ± 0.018	0.016 ± 0.007	0.186 ± 0.079	0.026 ± 0.011	0.027 ± 0.016	0.064 ± 0.020
FGD_∞ EFP	0.000 ± 0.000	0.025 ± 0.001	0.026 ± 0.001	0.002 ± 0.000	0.022 ± 0.001	0.004 ± 0.000	0.003 ± 0.000	0.030 ± 0.001
MMD Poly Deg=4 Quadratic Unbiased Est. EFP	-0.000 ± 0.000	0.002 ± 0.001	0.003 ± 0.002	0.000 ± 0.000	0.002 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.000
Precision EFP	0.894 ± 0.111	0.941 ± 0.039	0.978 ± 0.005	0.882 ± 0.077	0.680 ± 0.104	0.936 ± 0.057	0.667 ± 0.114	0.789 ± 0.092
Recall EFP	0.900 ± 0.123	0.881 ± 0.072	0.967 ± 0.015	0.916 ± 0.062	0.828 ± 0.050	0.921 ± 0.072	0.802 ± 0.101	0.763 ± 0.125
Wasserstein Distance PNet Activations	1.646 ± 0.063	1.699 ± 0.096	2.372 ± 0.388	1.708 ± 0.082	4.492 ± 0.145	1.789 ± 0.050	3.986 ± 0.362	7.595 ± 0.219
FGD_∞ PNet Activations	0.002 ± 0.001	0.042 ± 0.003	0.208 ± 0.013	0.006 ± 0.001	1.256 ± 0.028	0.019 ± 0.002	1.222 ± 0.017	3.635 ± 0.019
MMD Poly Deg=4 Quadratic Unbiased Est. PNet Activations	-0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.001	0.000 ± 0.001	0.007 ± 0.002	0.001 ± 0.000	0.006 ± 0.002	0.102 ± 0.010
Precision PNet Activations	0.684 ± 0.074	0.642 ± 0.043	0.706 ± 0.056	0.734 ± 0.029	0.088 ± 0.044	0.751 ± 0.083	0.078 ± 0.043	0.390 ± 0.078
Recall PNet Activations	0.701 ± 0.049	0.611 ± 0.039	0.612 ± 0.075	0.731 ± 0.058	0.014 ± 0.009	0.703 ± 0.105	0.014 ± 0.011	0.572 ± 0.087
Classifier AUC	0.50	0.52	0.54	0.50	0.97	0.81	0.93	0.99

- W_1^M is sensitive to some, but not all distortions
- Wasserstein distance is sensitive to most, but very slow to converge
- Despite Gaussian assumption, FGD is sensitive to all distortions
- Performance for EFPs and PNet activations is similar
- Classifier identifies particle feature distortions but misses distribution-level discrepancies

TAKEAWAYS

- Re-iterating Cousins 2016: no best g.o.f. test for all alternative hypotheses
    - His suggestion: use multiple, covering the relevant alternatives
- FGD proves to be the most sensitive for typical distortions we expect
    - Hand-engineered features and ParticleNet activations are similarly sensitive
    - Hand engineered are more interpretable, standardisable, and efficient
    - ⇒ **Recommend Fréchet Jet and Calo Distances**, using EFPs and shower-shape variables, for overall model evaluation and comparison
- But FGD can miss shape discrepancies, so continue with 1D histograms (W_1) as well
- Next steps:
    - Discuss with the ML4Sim community
    - Report on arXiv later this month
    - Implement in JetNet for easy, standard use
    - Pull request to Calo Challenge?



BACKUP

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)

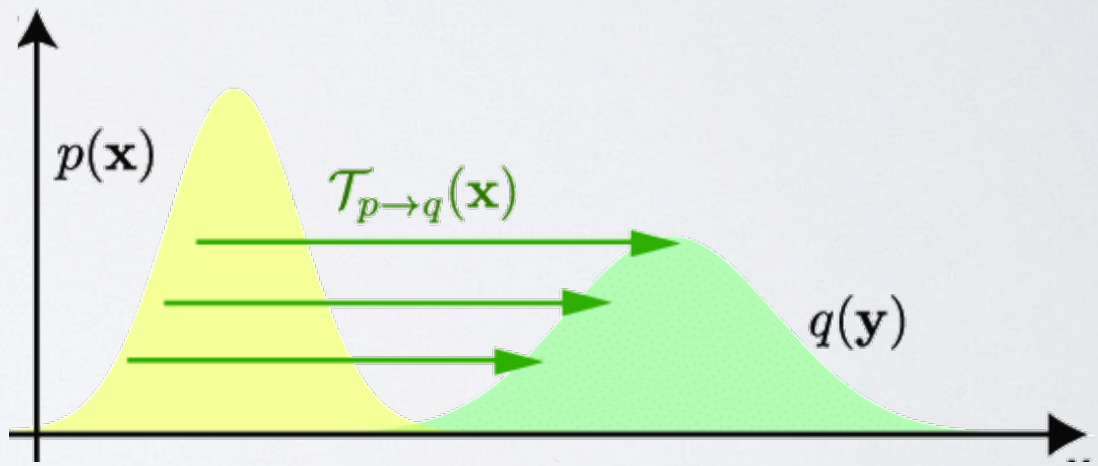
$$\text{FGD} = \text{Frechet}(\mathcal{N}(\mu_\text{r}, \boldsymbol{\Sigma}_\text{r}), \mathcal{N}(\mu_\text{g}, \boldsymbol{\Sigma}_\text{g}))$$

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

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

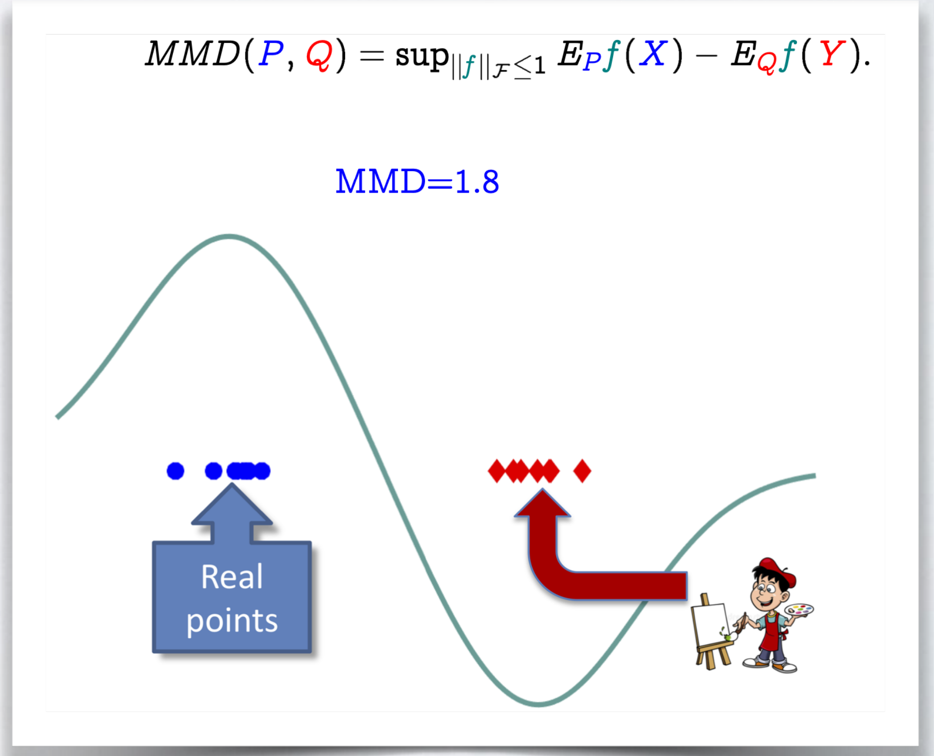


MORE ON IPMS

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

- 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



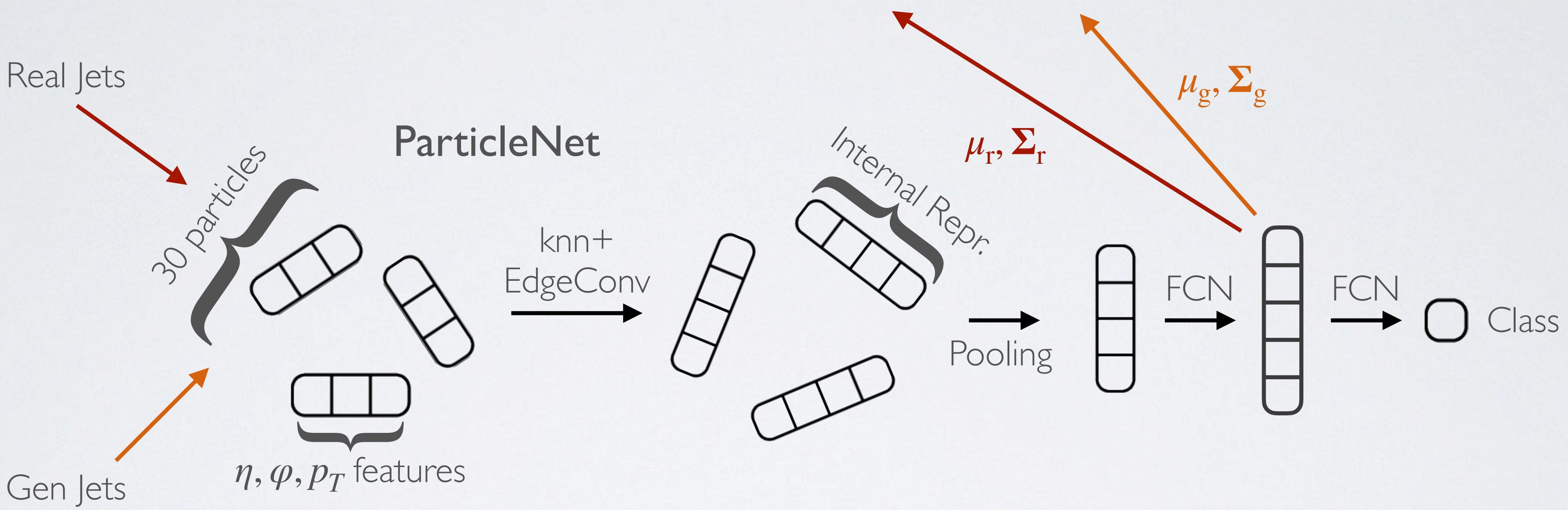
FRÉCHET <CLASSIFIER> DISTANCES

- 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:

$$\mathrm{FGD} = \mathrm{Frechet}(\mathcal{N}(\mu_r, \boldsymbol{\Sigma}_r), \mathcal{N}(\mu_g, \boldsymbol{\Sigma}_g)) = ||\mu_r - \mu_g||^2 + \mathrm{Tr}\left[\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\right]$$



- 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

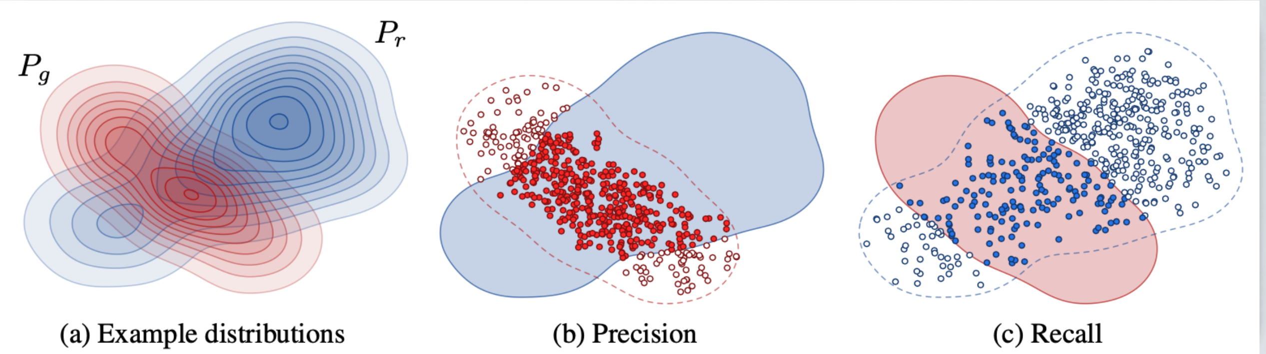
$$\sup_{f \in \mathscr{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)|$$

- IPM where $\mathscr{F}$ is unit ball in the reproducing kernel Hilbert space (RKHS) for kernel $k(x, y)$
  - RKHS $\Leftrightarrow f(x) = \langle f, \varphi(x) \rangle_{\mathscr{F}}$, where $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathscr{F}}$
  - $\mathbb{E}_{x \sim p} f(x) = \langle f, \mathbb{E}_{x \sim p} \varphi(x) \rangle_{\mathscr{F}} = \langle f, \mu_p \rangle_{\mathscr{F}}$
  - μ_p is the embedding of distribution p in $\mathscr{F}$
  - if k is 'characteristic', e.g. Gaussian, $p \to \mu_p$ is injective (μ_p captures everything)

$$\Rightarrow \sup_{f \in \mathscr{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)| = \sup_{f \in \mathscr{F}} |\langle f, \mu_{p_{\text{real}}} - \mu_{p_{\text{gen}}} \rangle_{\mathscr{F}}| = || \mu_{p_{\text{real}}} - \mu_{p_{\text{gen}}} ||$$

- MMD: distance between means in embedding space
- Very powerful method for calculating distance between distributions

TESTS FOR QUALITY / DIVERSITY



- 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

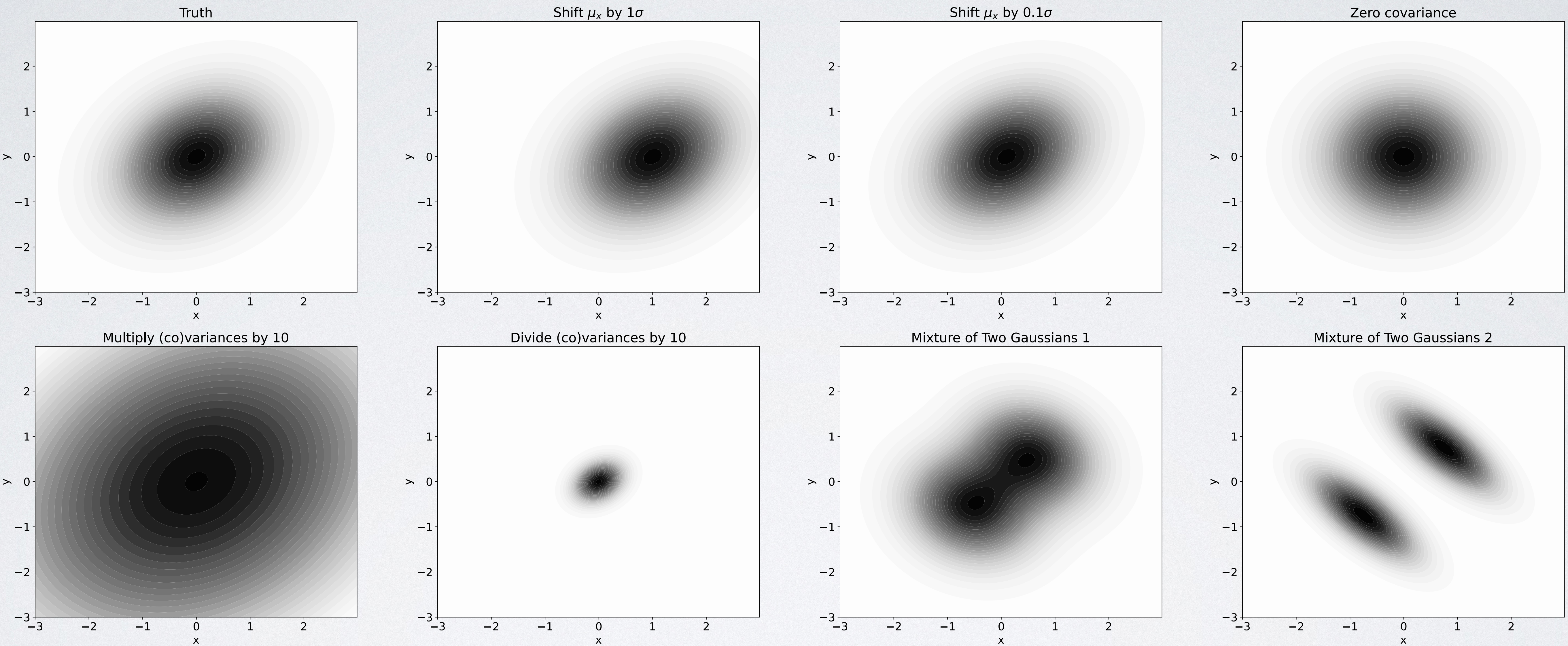
CLASSIFIER-BASED TESTS

- Train a classifier between real and generated data
- Friedman 2003, Paz and Oquab 2017 (C2ST), Liu et al. 2020 (Deep Kernel 2ST), Krause and Shih (2021)
- Can be powerful test of quality and diversity
- Not interpretable
- Hard to generalise to conditional evaluation
- Hard to standardise (need to choose an "optimal" classifier for relevant alternatives)
- Not generally reproducible (for non-convex, stochastic optimisation)
- Inefficient (Need to re-train for each dataset and algorithm)

TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

Tests if metrics are sensitive to correlations



Tests sensitivity to quality

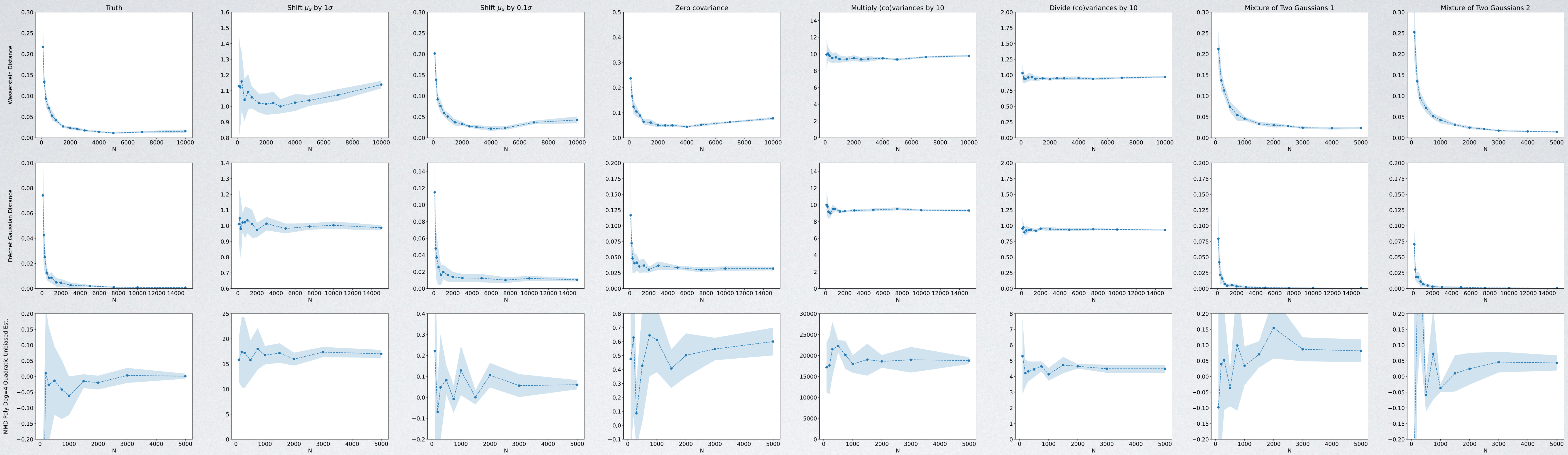
Tests sensitivity to diversity

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

Same statistics, but easier to distinguish (by eye)

RESULTS

- Scores vs. sample size (N)



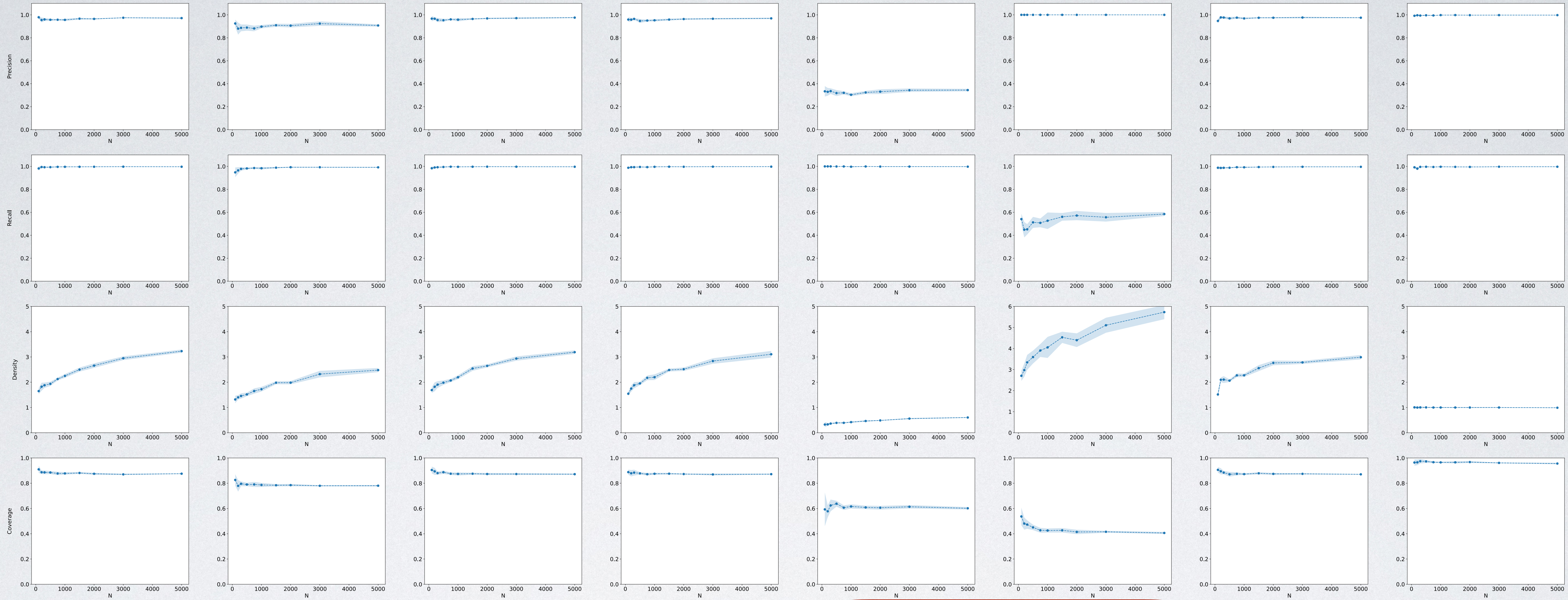
- Scores for largest N

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 Distance	0.016 ± 0.004	1.139 ± 0.024	0.043 ± 0.008	0.077 ± 0.006	9.792 ± 0.126	0.969 ± 0.013	0.023 ± 0.003	0.014 ± 0.002
Fréchet Gaussian Distance	0.001 ± 0.000	0.987 ± 0.016	0.010 ± 0.002	0.032 ± 0.003	9.320 ± 0.121	0.932 ± 0.010	0.001 ± 0.000	0.001 ± 0.000
MMD Poly Deg=4 Quadratic Unbiased Est.	-0.000 ± 0.005	16.576 ± 0.478	0.104 ± 0.031	0.550 ± 0.035	19395.900 ± 617.497	4.761 ± 0.048	0.073 ± 0.010	0.019 ± 0.011

- Wasserstein and FGD are biased (value depends on N) but work well overall
- Can't distinguish mixtures of Gaussians
- MMD estimator unbiased, converges ~quickly, can distinguish mixtures of Gaussians (after tuning kernel)

RESULTS

- P&R vs D&C



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
Precision	0.972 ± 0.005	0.907 ± 0.010	0.976 ± 0.004	0.969 ± 0.006	0.345 ± 0.011	1.000 ± 0.000	0.975 ± 0.003	0.998 ± 0.001
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.998 ± 0.001	0.998 ± 0.001	0.585 ± 0.018	0.996 ± 0.001	0.997 ± 0.001
Density	3.230 ± 0.063	2.480 ± 0.083	3.190 ± 0.071	3.107 ± 0.132	0.603 ± 0.015	5.731 ± 0.336	2.990 ± 0.087	0.989 ± 0.009
Coverage	0.876 ± 0.002	0.780 ± 0.006	0.872 ± 0.005	0.872 ± 0.004	0.602 ± 0.010	0.406 ± 0.008	0.871 ± 0.002	0.956 ± 0.006

- P&R match our intuition better
- Biased, but converge quickly