GENERATIVE TRANSFORMERS AND HOW TO EVALUATE THEM

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

*Also Fermilab

IML Meeting
14/02/2023

ML4PS @ NeurIPS 2020
NeurIPS 2021
2211.10295 2022

GENERATIVE TRANSFORMERS AND HOW TO EVALUATE THEM

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

*Also Fermilab

IML Meeting
14/02/2023

LHC SIMULATIONS

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

LHC SIMULATIONS

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

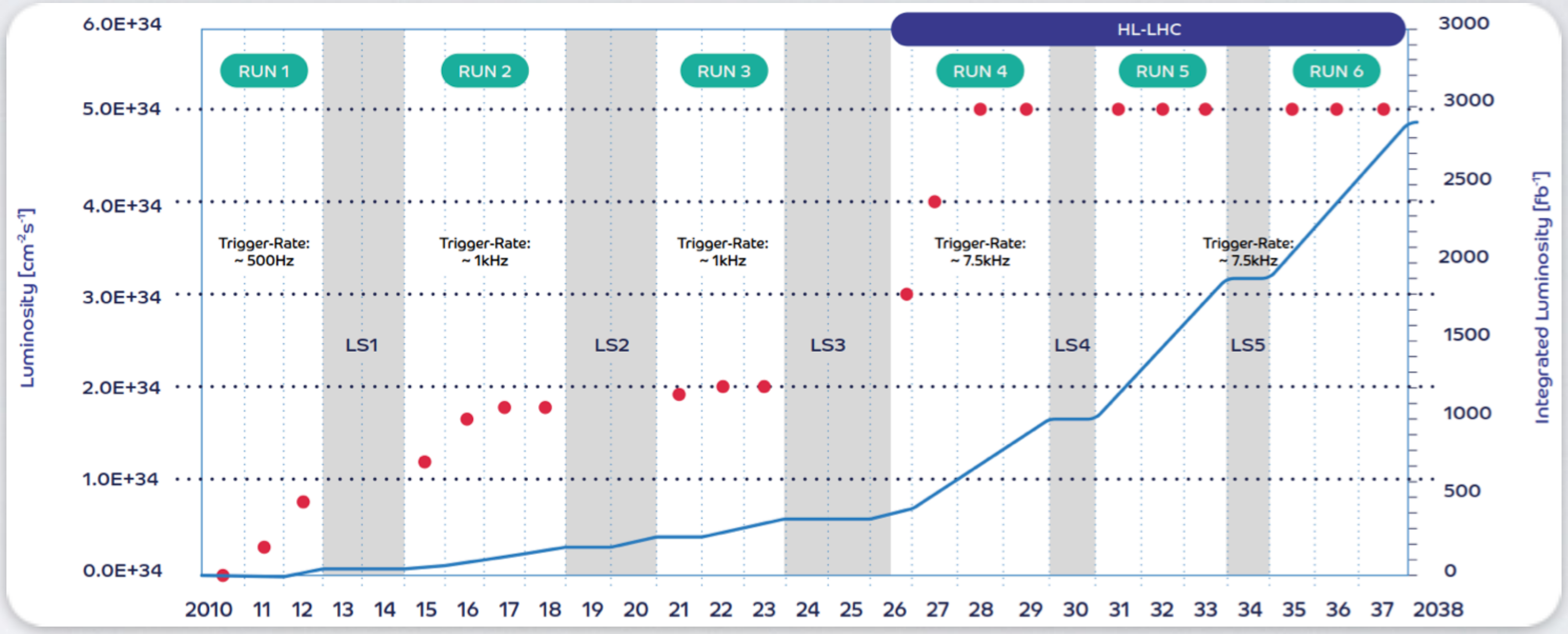
- Full detector simulation takes ~40% of grid CPU resources

LHC SIMULATIONS

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

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

*Delayed



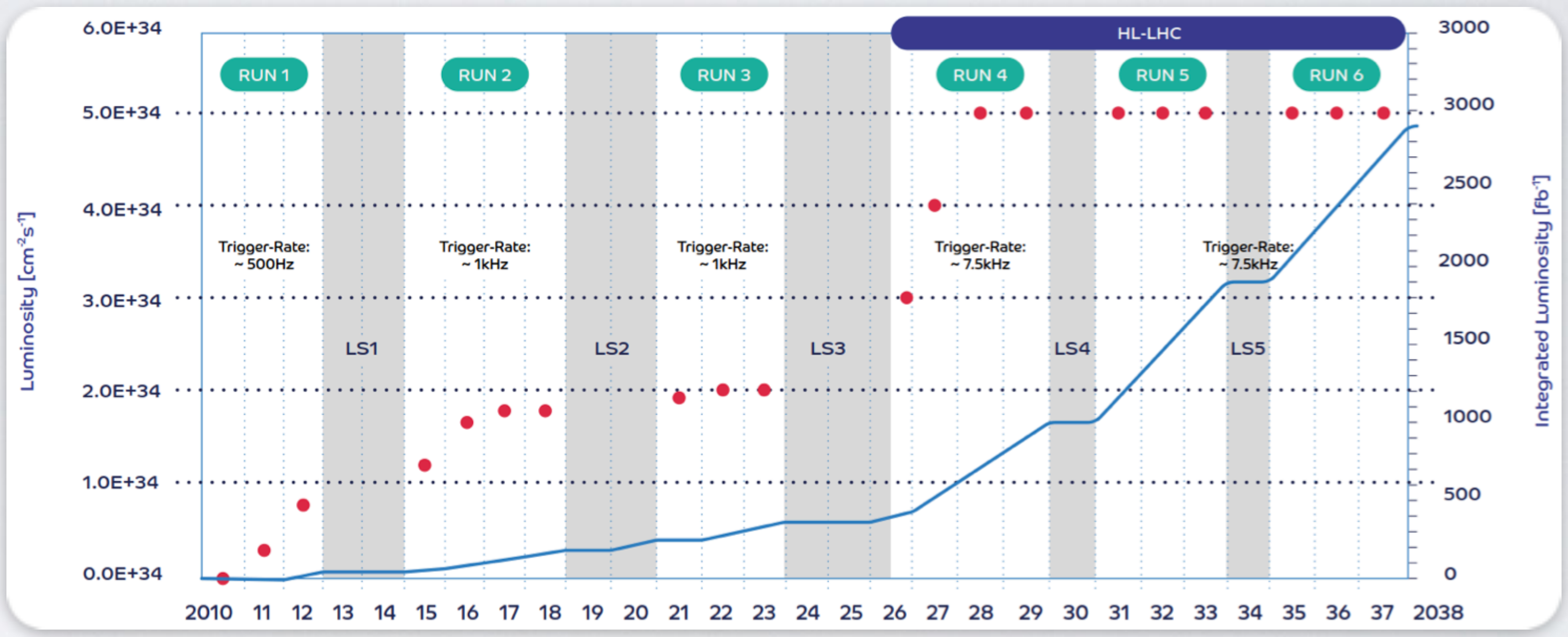
- HL-LHC looming

LHC SIMULATIONS

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

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

*Delayed



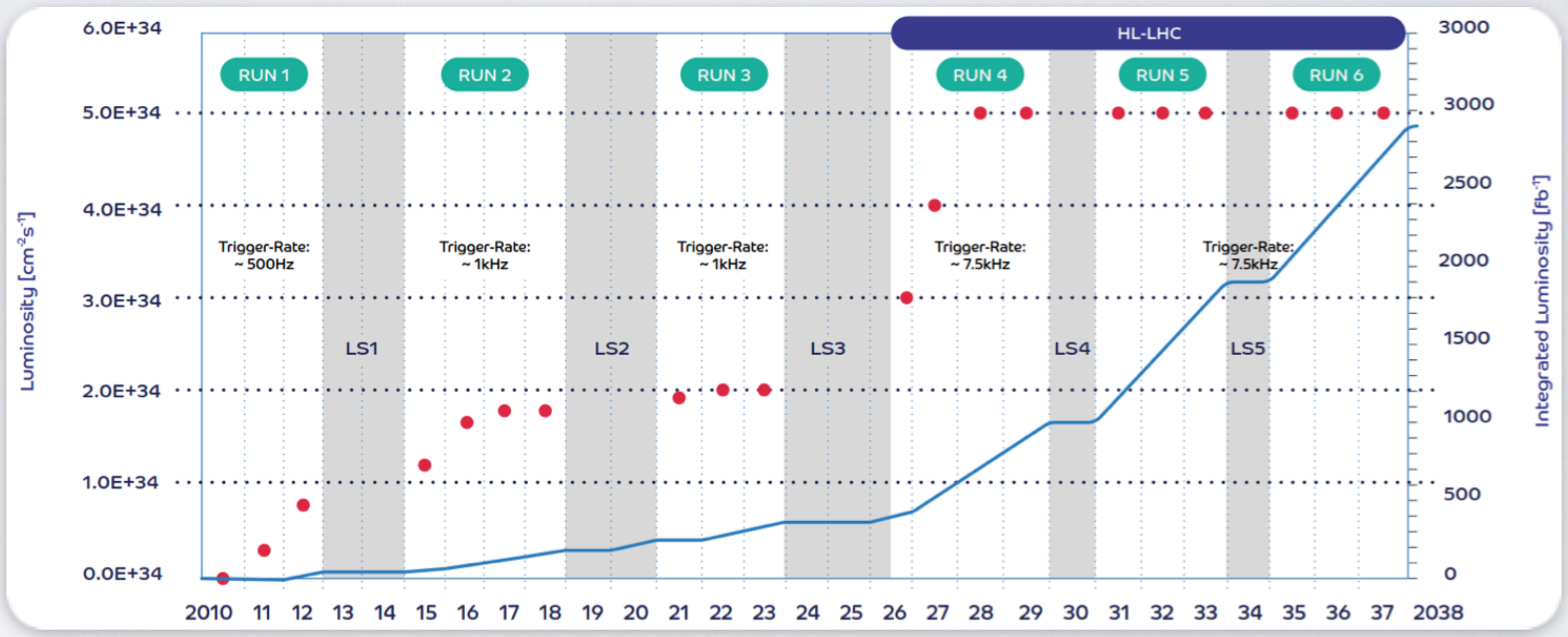
- HL-LHC looming
  - Order-of-magnitude more simulations needed

LHC SIMULATIONS

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

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

*Delayed



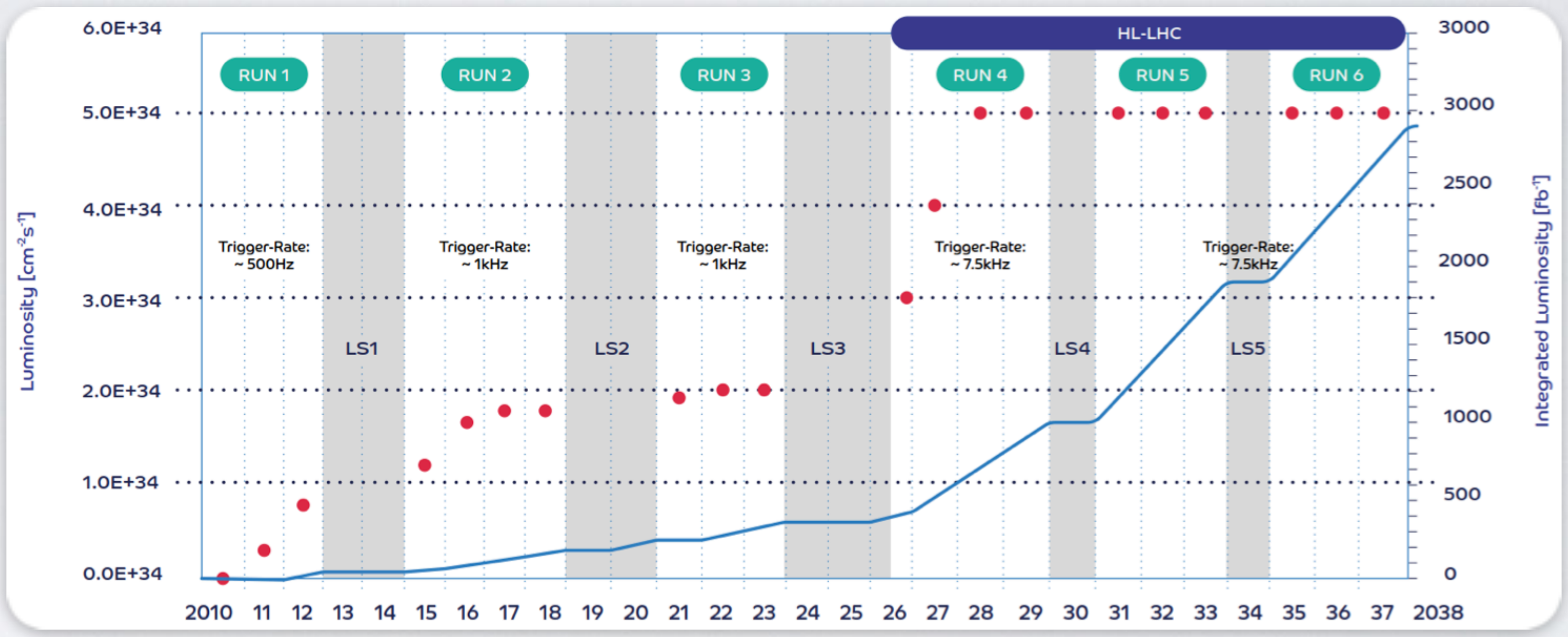
- HL-LHC looming
  - Order-of-magnitude more simulations needed
  - Improved detectors ⇒ higher granularity, increased complexity

LHC SIMULATIONS

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

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

*Delayed



- HL-LHC looming
  - Order-of-magnitude more simulations needed
  - Improved detectors ⇒ higher granularity, increased complexity
  - ML a possible solution?

LHC SIMULATIONS*

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

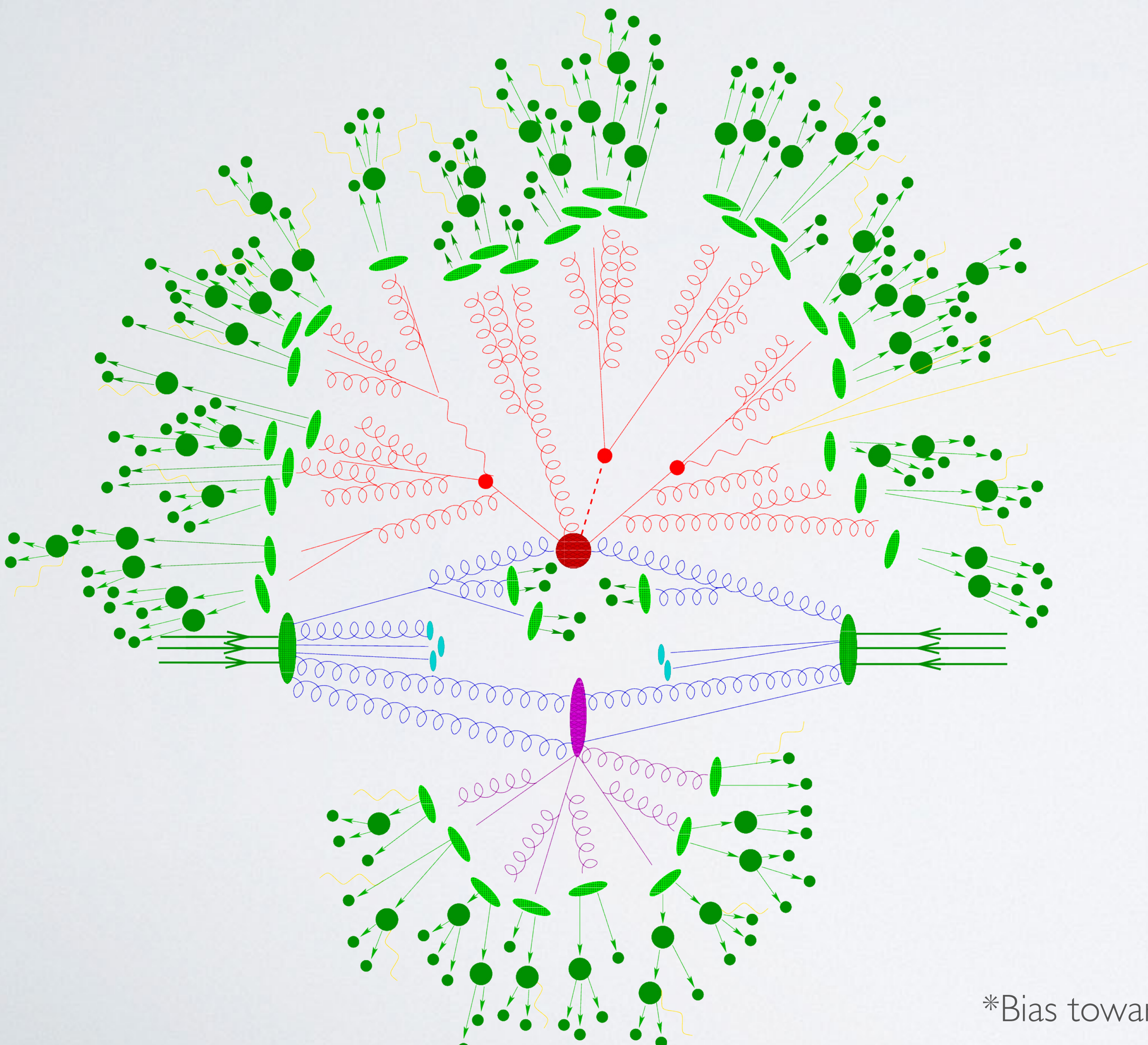
*Bias towards CMS

LHC SIMULATIONS*

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

Hard process
Showering
Hadronization
Underlying event

GenParticles



*Bias towards CMS

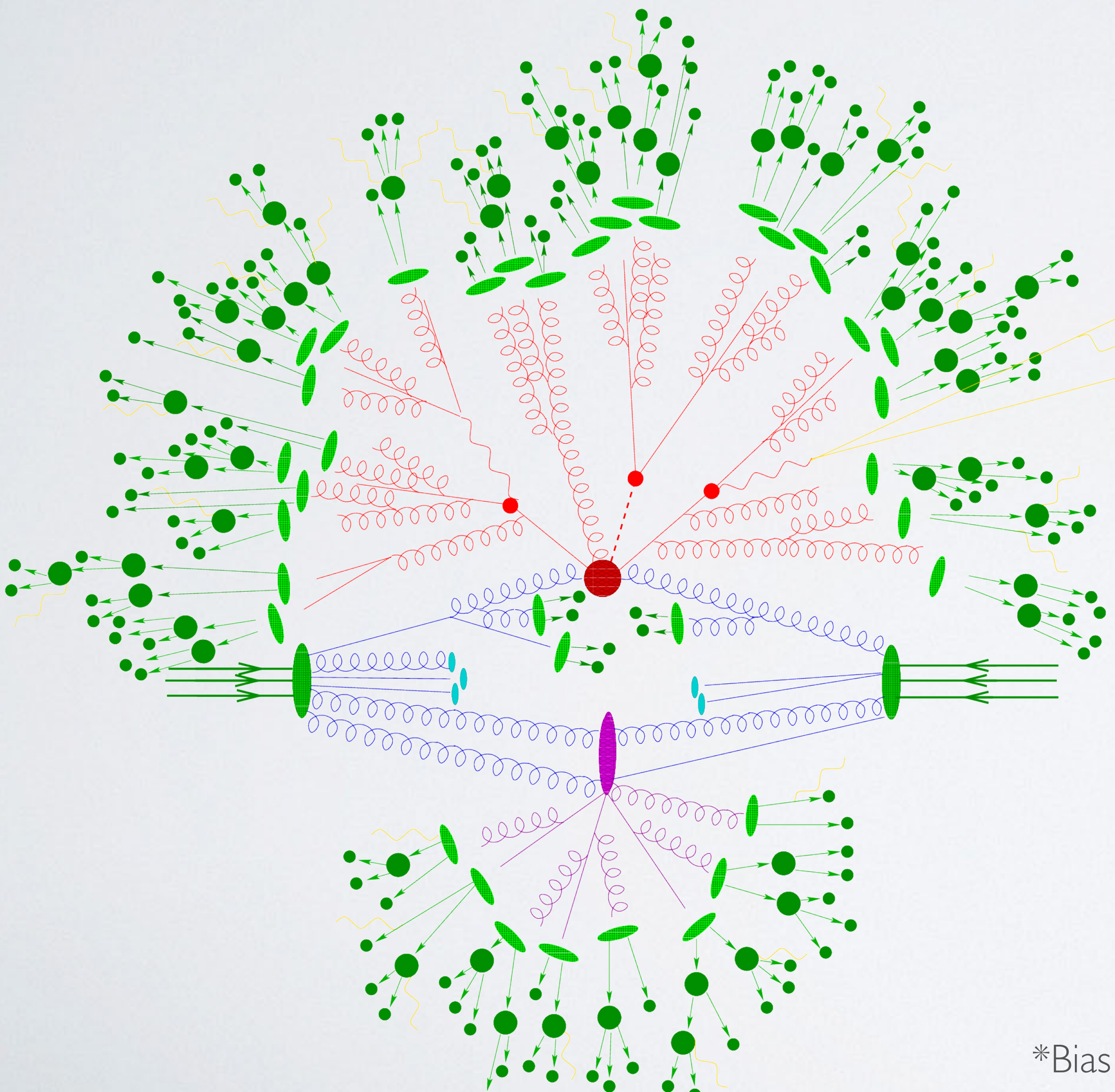
LHC SIMULATIONS*

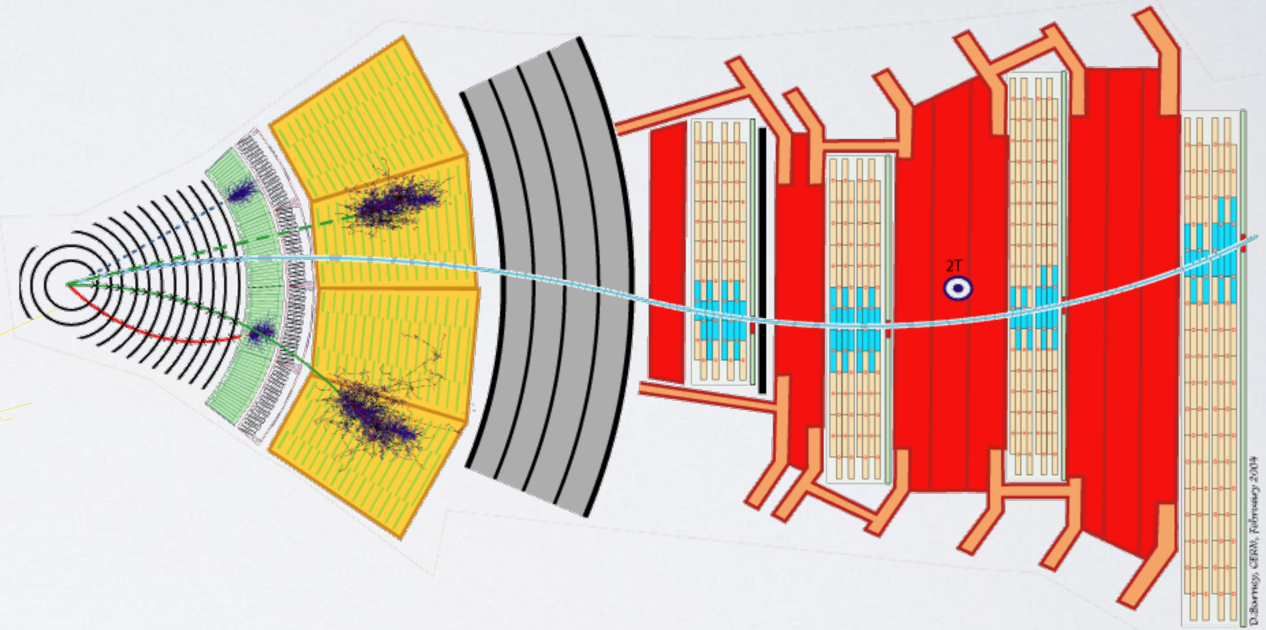
Sources
S. Sekmen, LPC 2017
F. Krauss, Kyoto 2011

Hard process
Showering
Hadronization
Underlying event

Energy depositions
and interactions
along detector

GenParticles → SimHits





*Bias towards CMS

LHC SIMULATIONS*

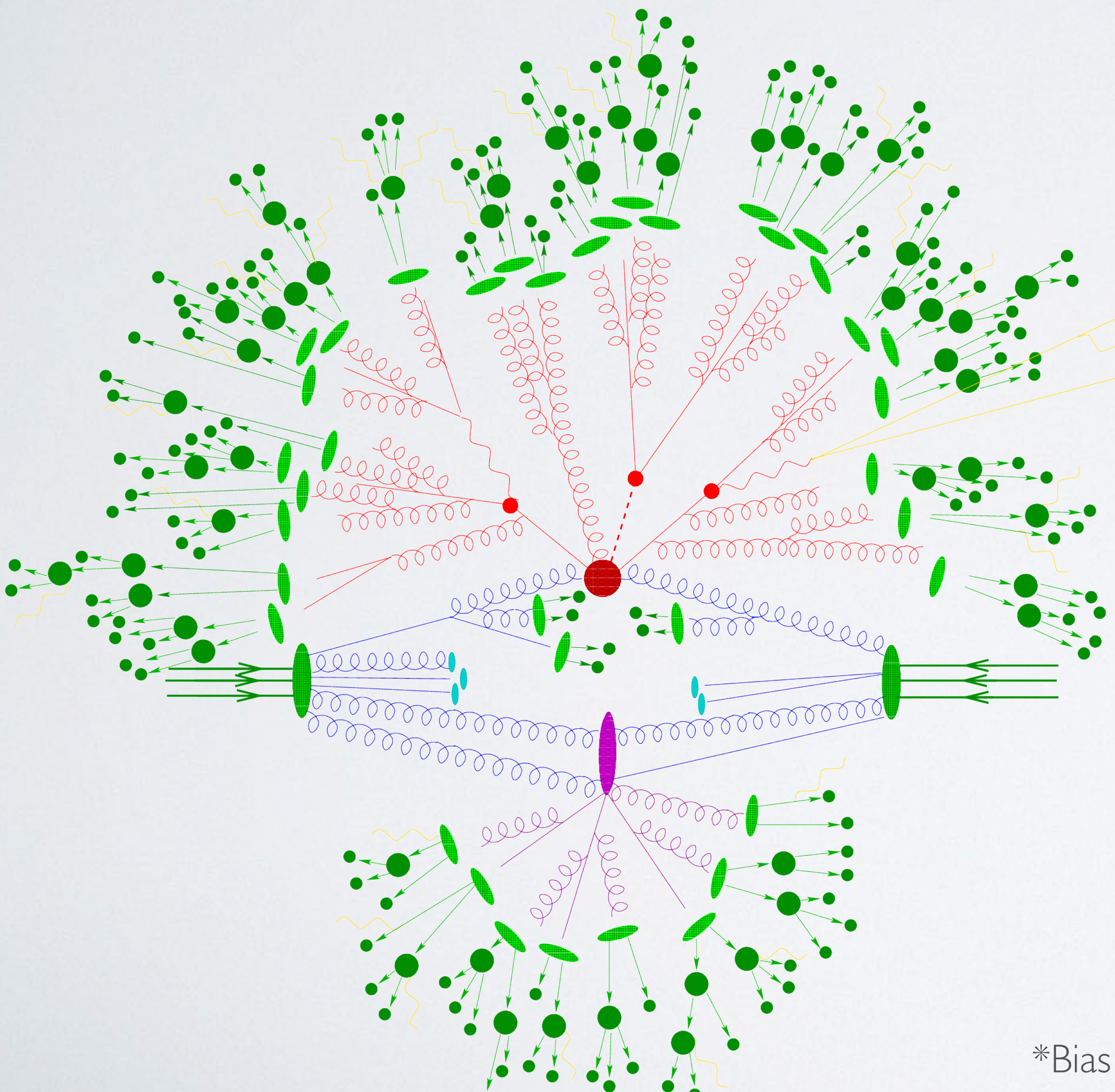
Sources
S. Sekmen, LPC 2017
F. Krauss, Kyoto 2011

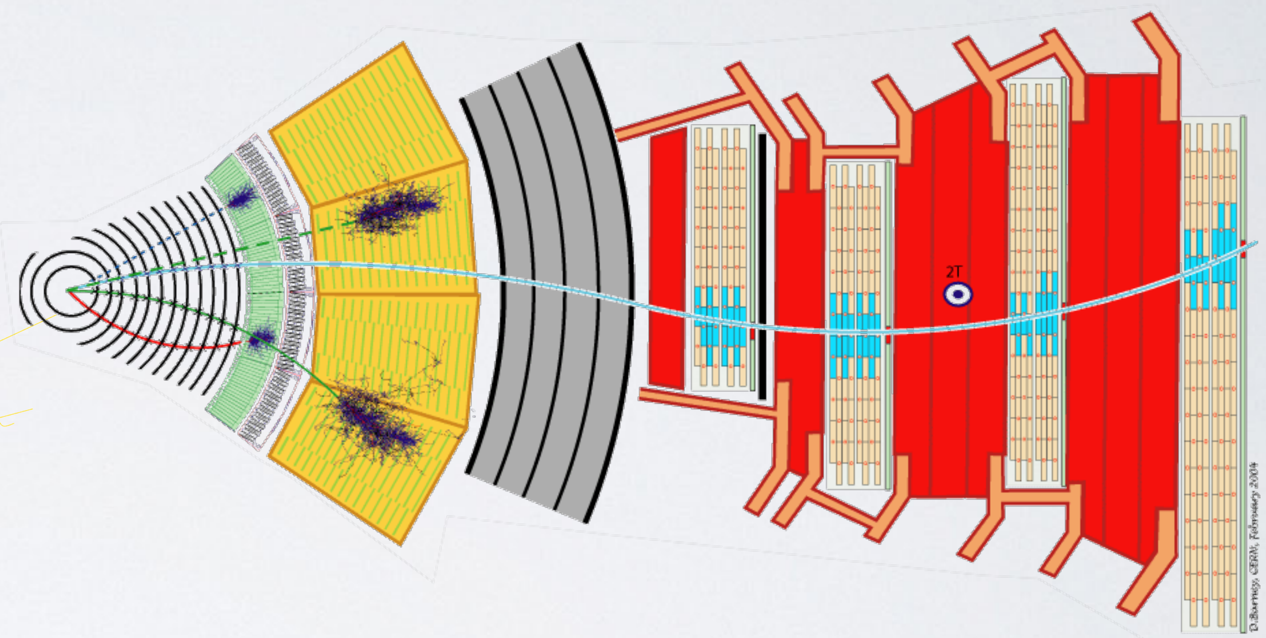
Hard process
Showering
Hadronization
Underlying event

Energy depositions and interactions along detector

Digitization
Hit reconstruction

GenParticles → SimHits → RecoHits





*Bias towards CMS

LHC SIMULATIONS*

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

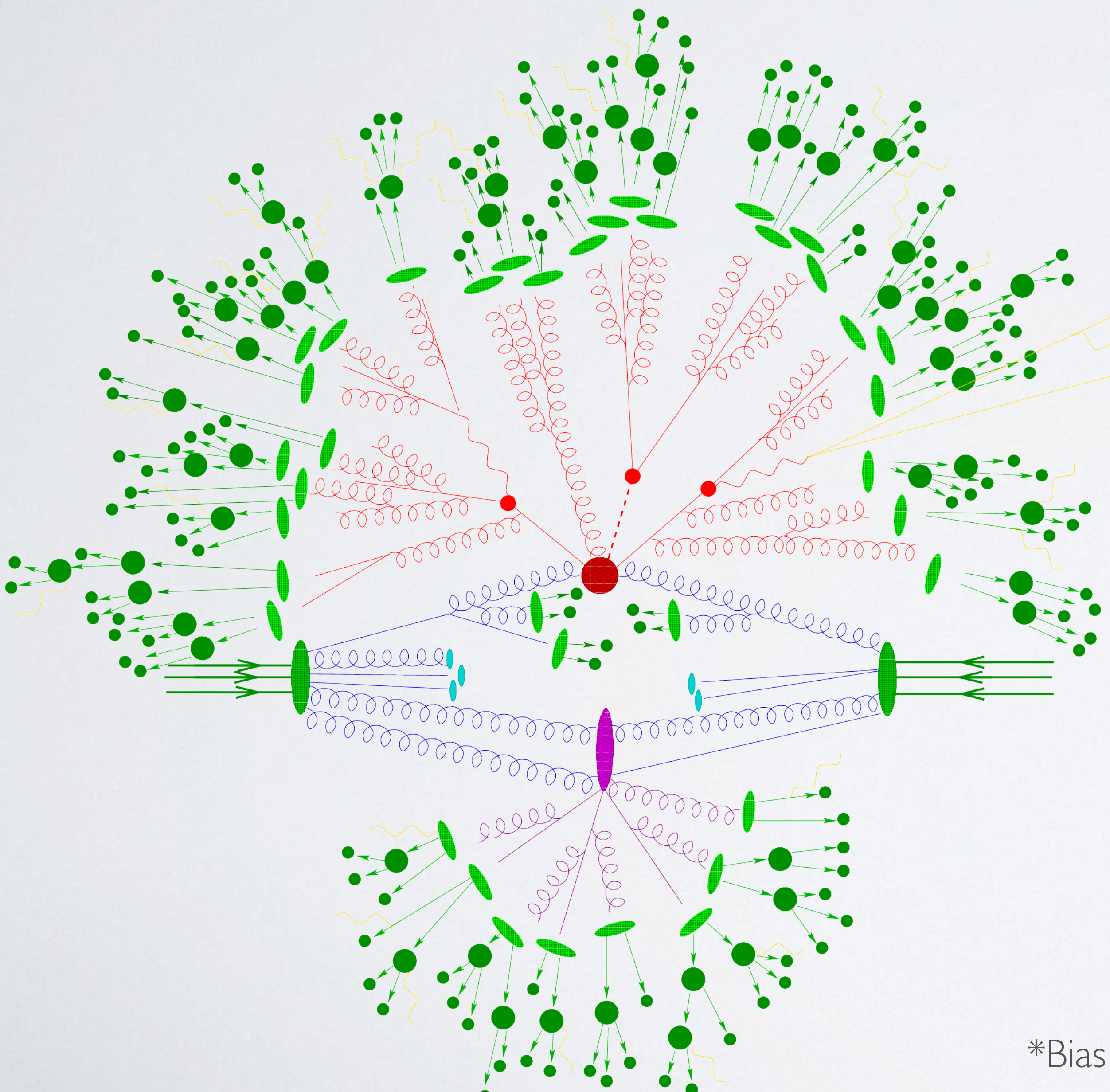
Hard process
Showering
Hadronization
Underlying event

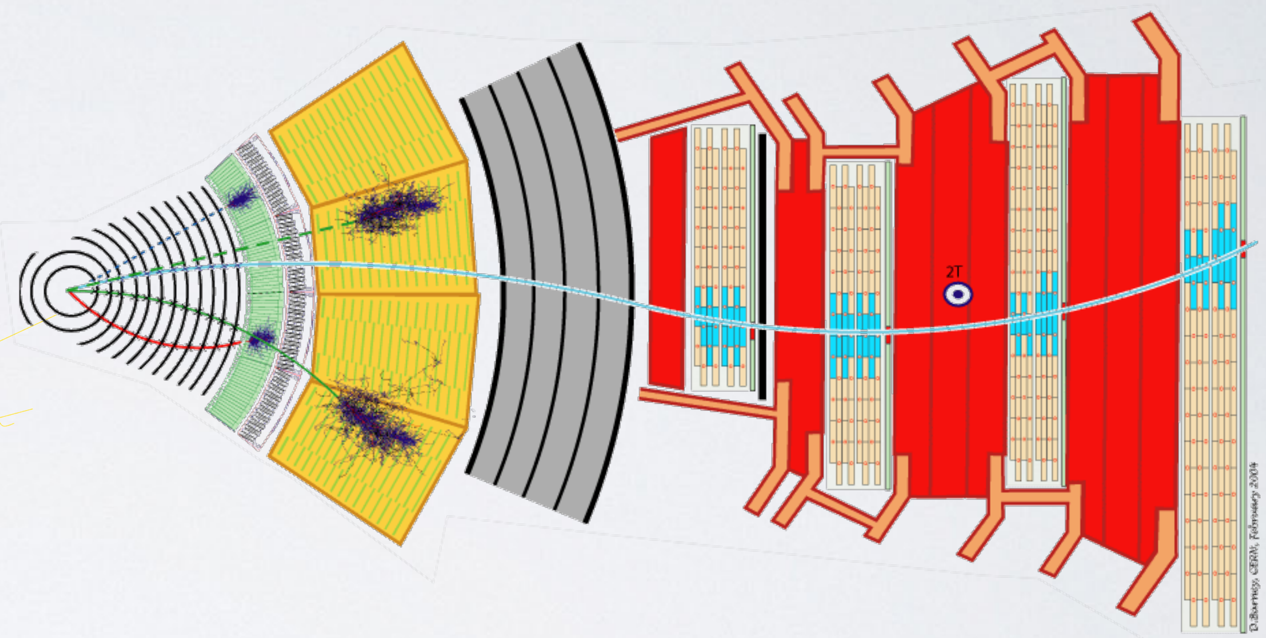
Energy depositions
and interactions
along detector

Digitization
Hit reconstruction

Particle
reconstruction

GenParticles → SimHits → RecoHits → Reco Particles





*Bias towards CMS

LHC SIMULATIONS*

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

Hard process
Showering
Hadronization
Underlying event

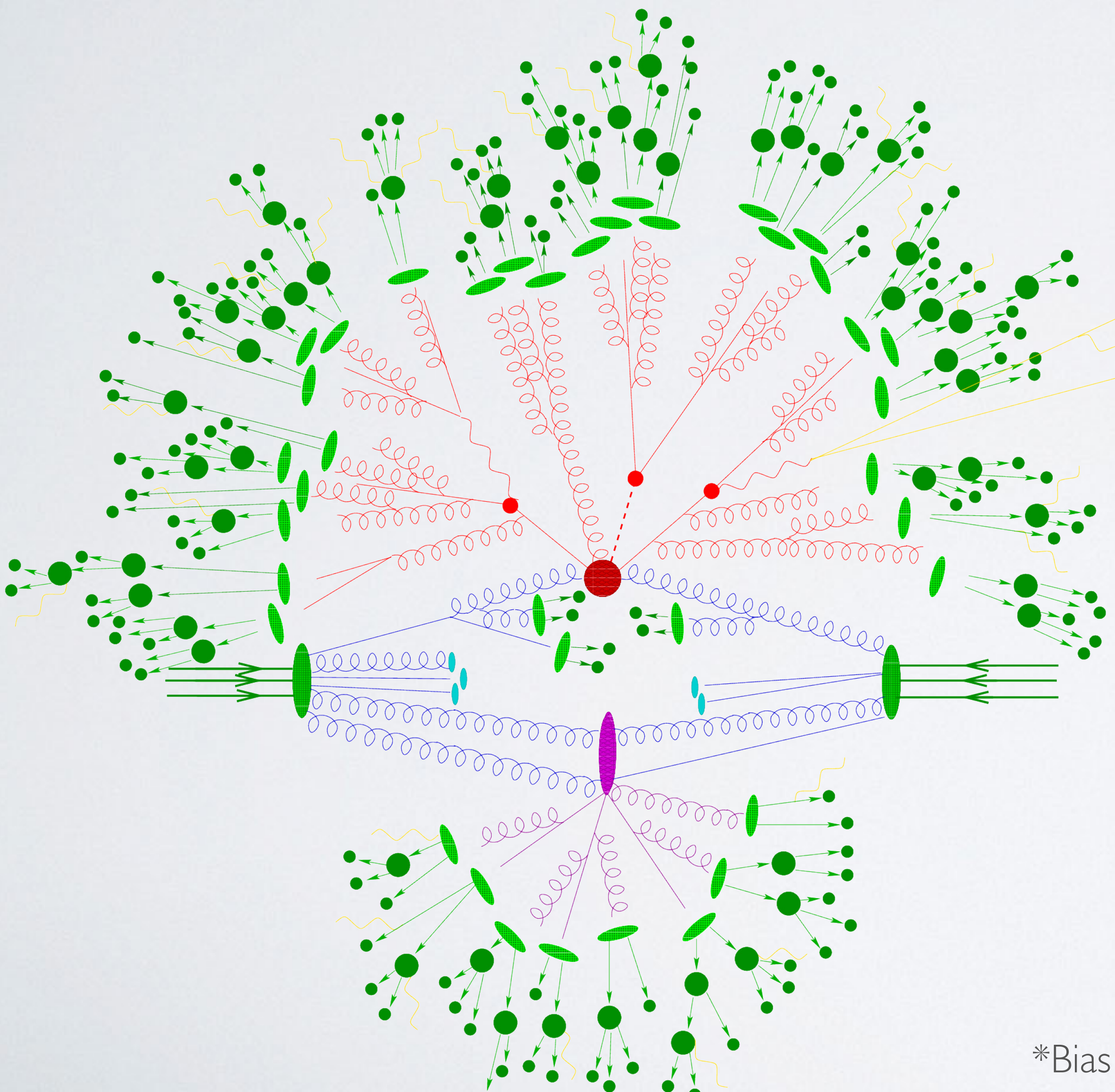
Energy depositions and interactions along detector

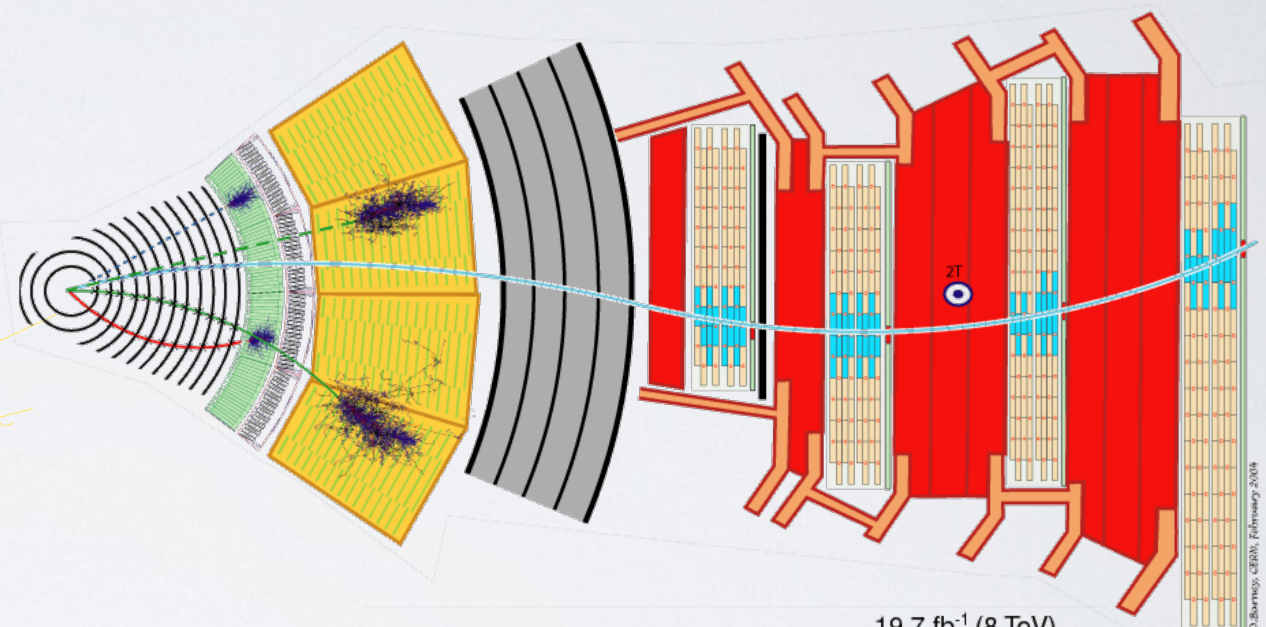
Digitization
Hit reconstruction

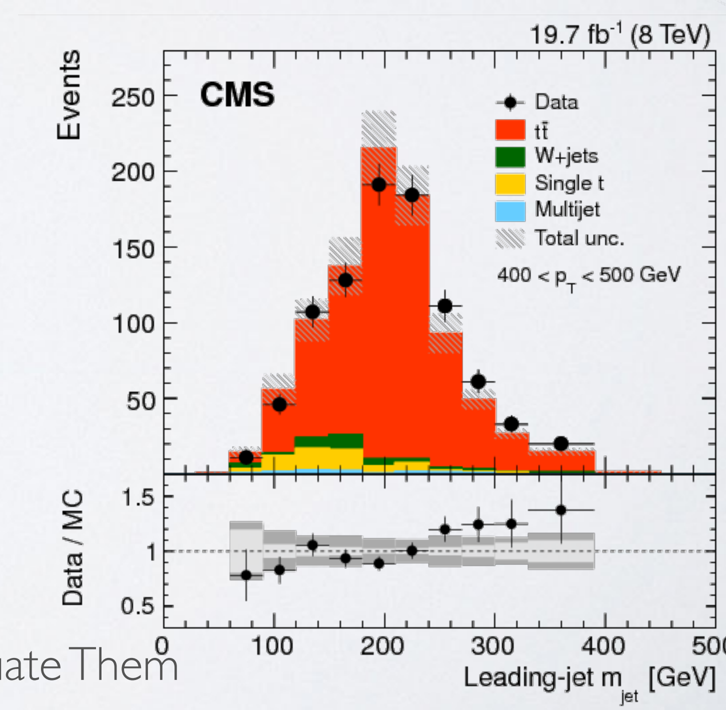
Particle reconstruction

Event-level analysis

GenParticles → SimHits → RecoHits → Reco Particles → High level features

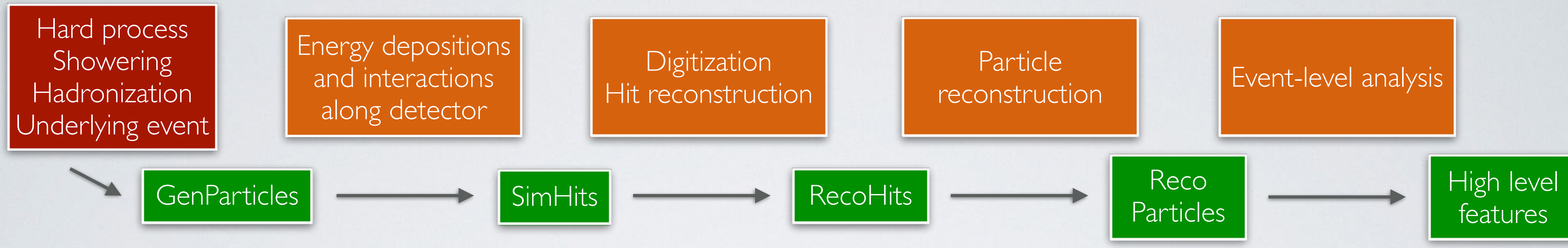




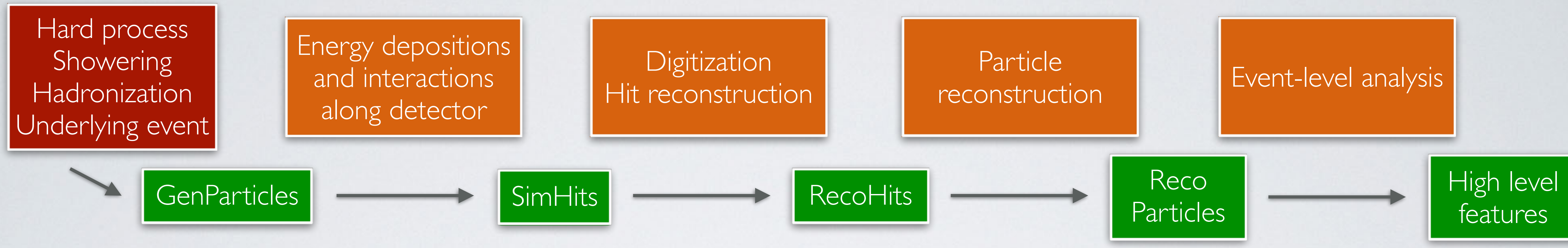


*Bias towards CMS

LHC SIMULATIONS

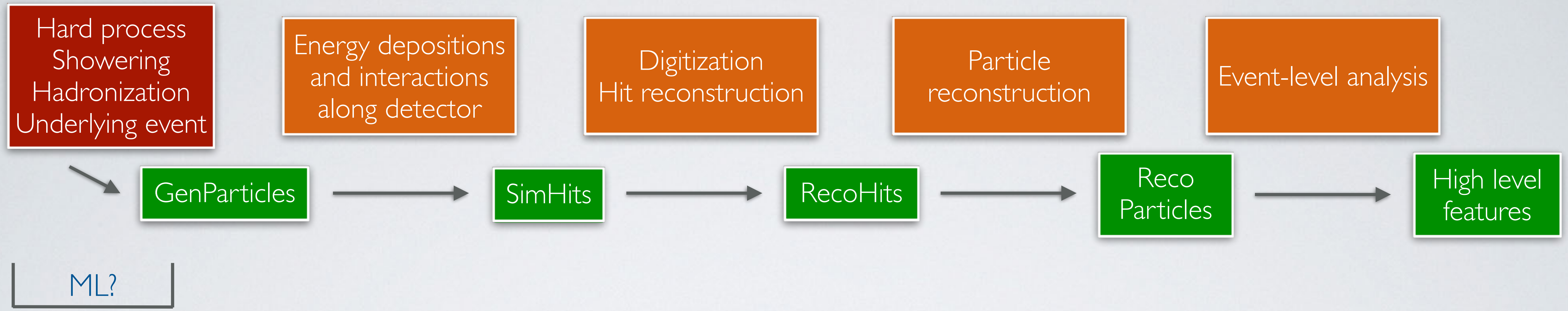


LHC SIMULATIONS



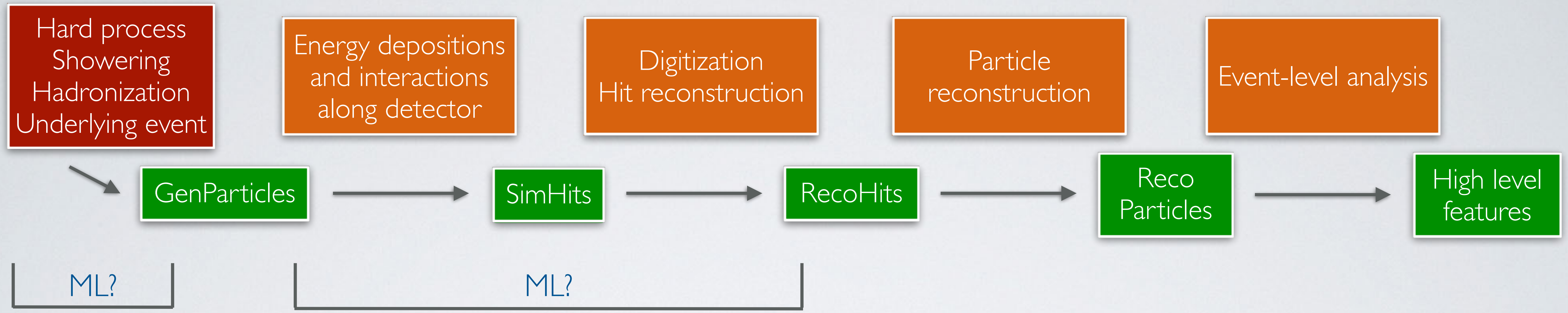
- Opportunity for ML alternatives in many steps

LHC SIMULATIONS



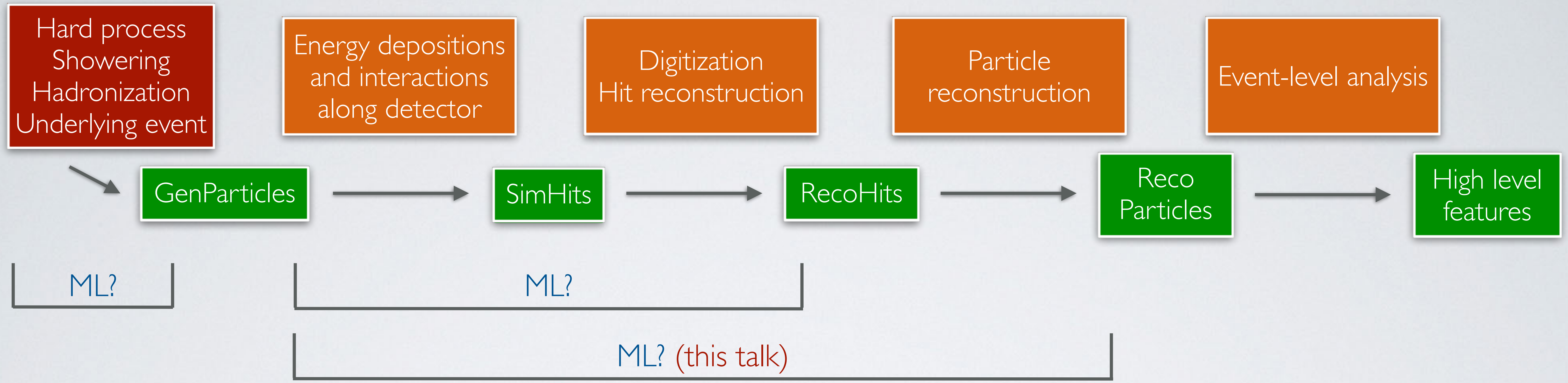
- Opportunity for ML alternatives in many steps

LHC SIMULATIONS



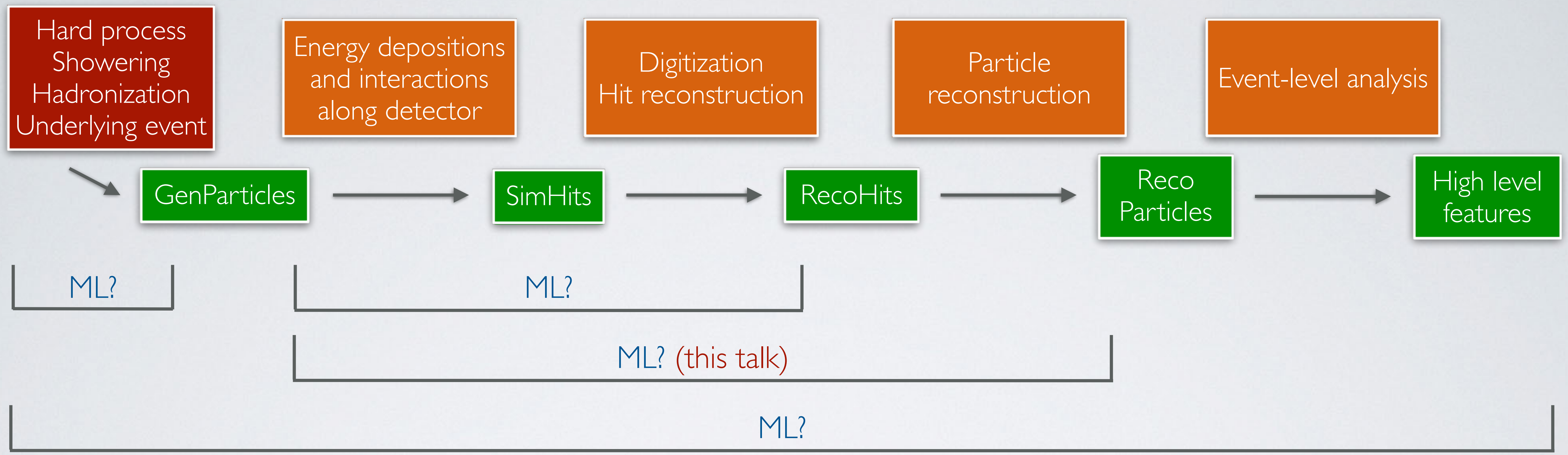
- Opportunity for ML alternatives in many steps

LHC SIMULATIONS



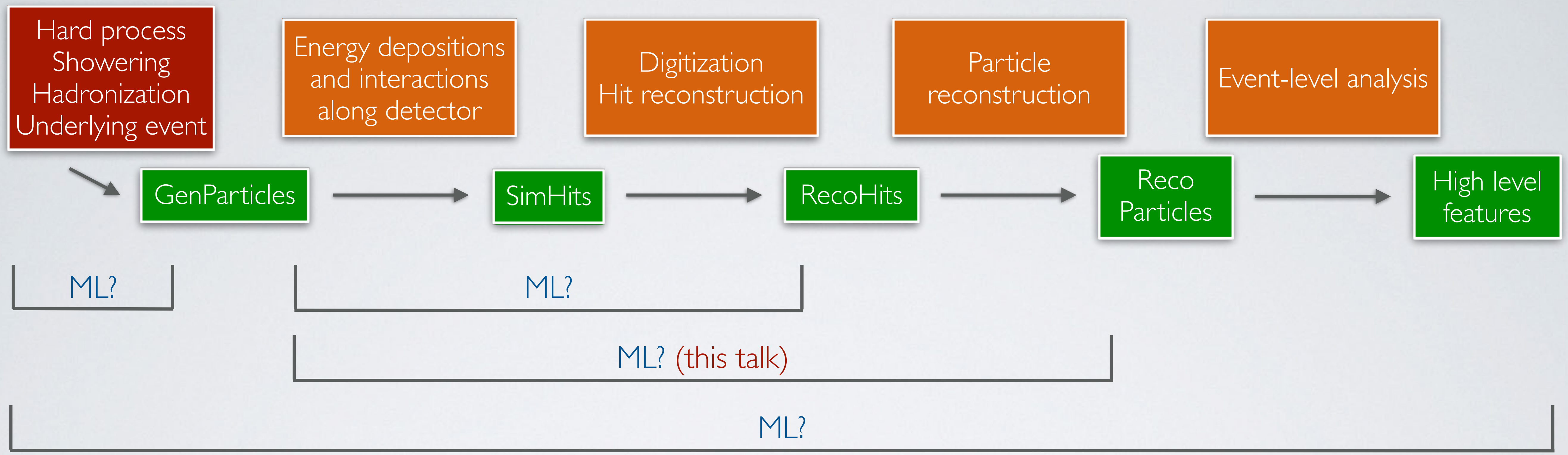
- Opportunity for ML alternatives in many steps

LHC SIMULATIONS



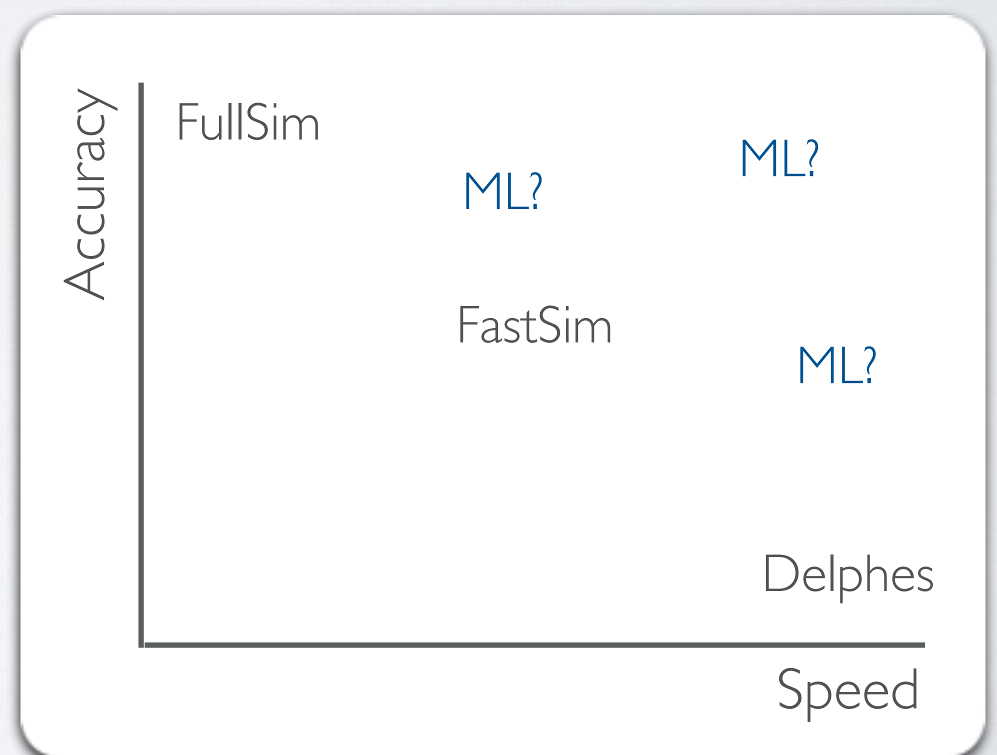
- Opportunity for ML alternatives in many steps

LHC SIMULATIONS



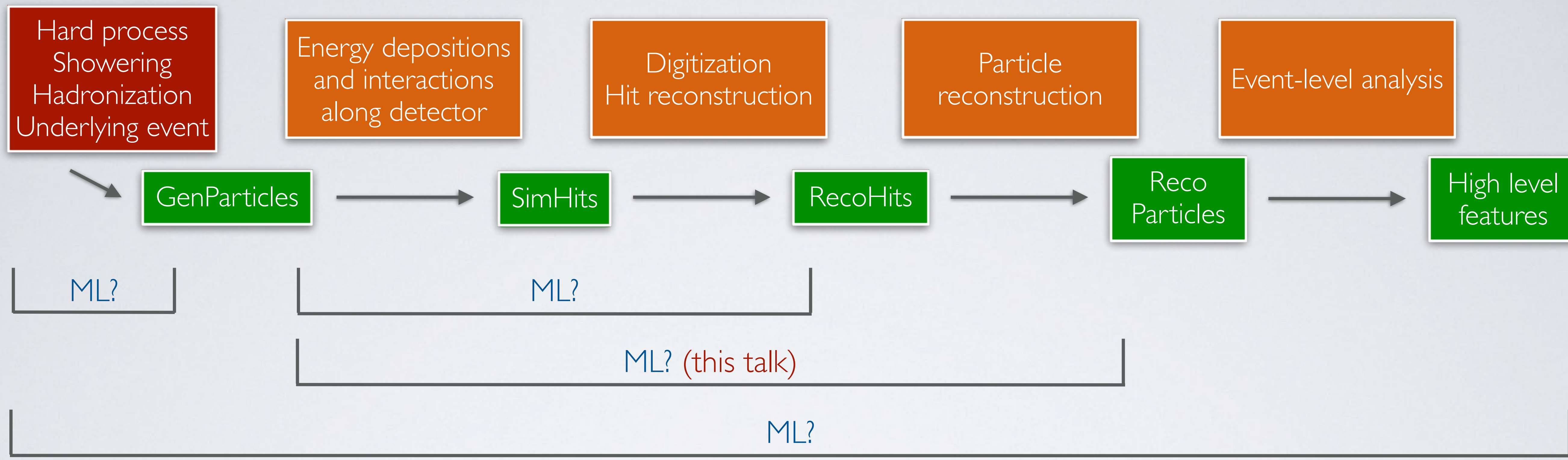
- Opportunity for ML alternatives in many steps

- Trading accuracy of "FullSim" (Geant) for speed

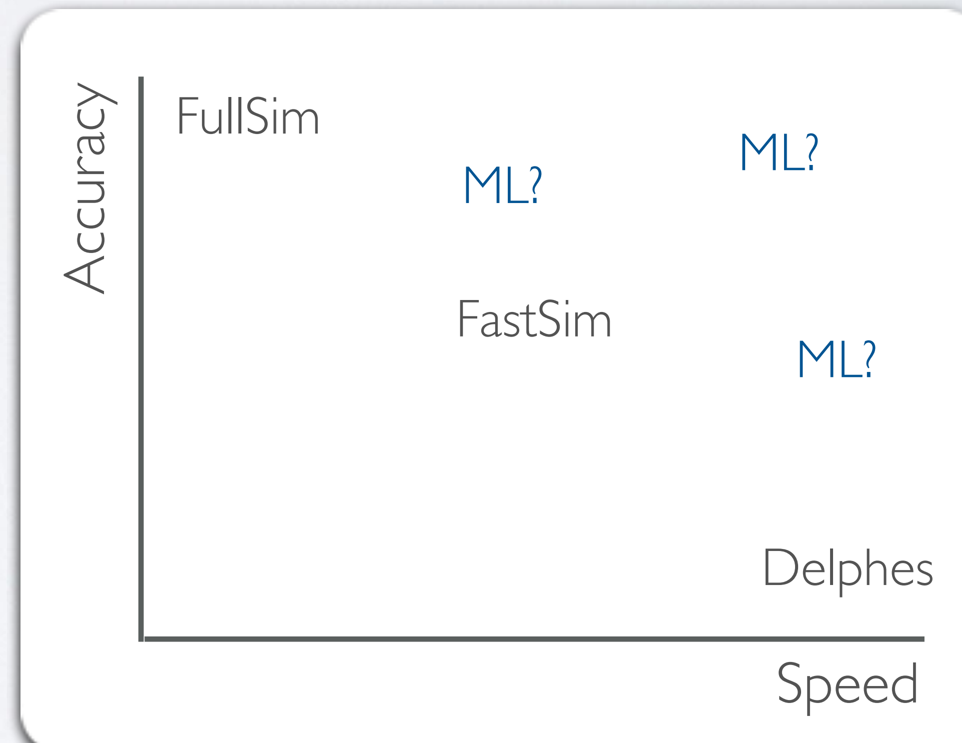


K. Pedro, HSF 2020

LHC SIMULATIONS



- Opportunity for ML alternatives in many steps
- Trading accuracy of "FullSim" (Geant) for speed
- Trading interpretability/trust for # of steps



K. Pedro, HSF 2020

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

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

- *"It is time to harvest"* - CMS ML Townhall 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 **trust** generated data?

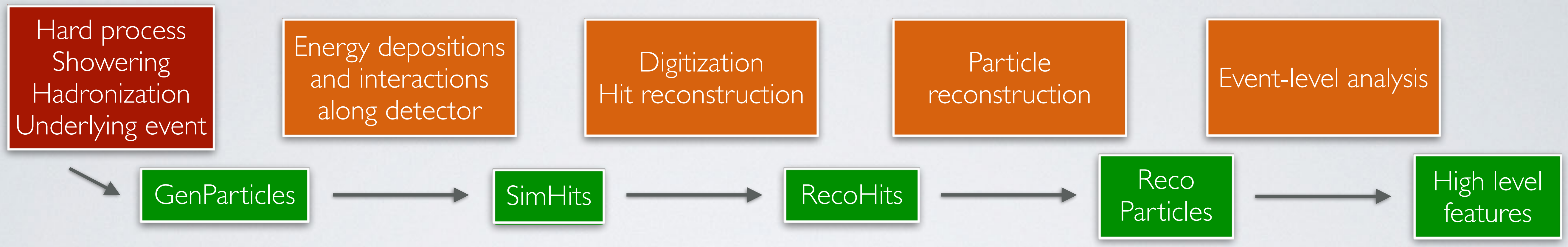
- How do we compare generative models?

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

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

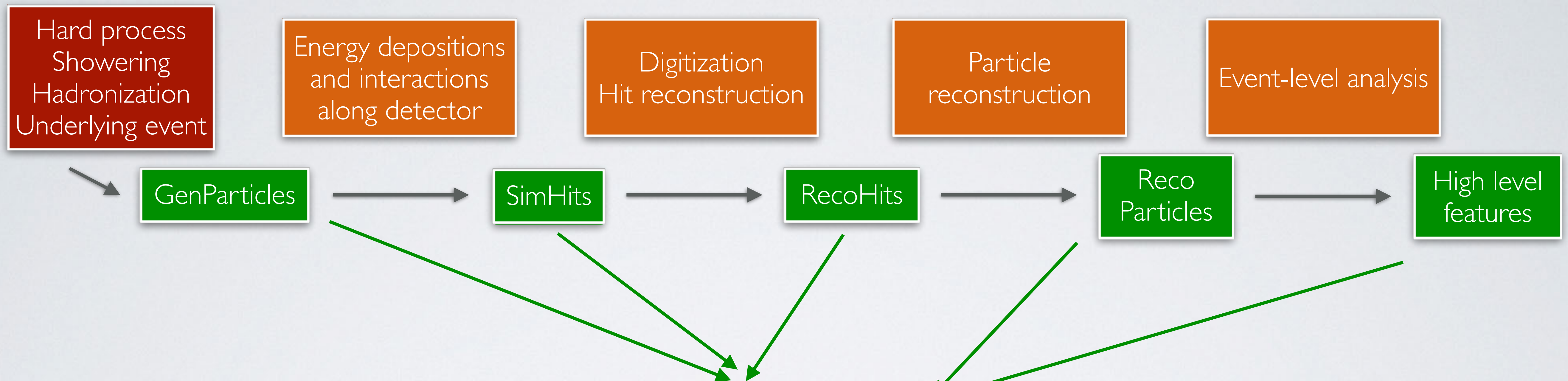
PROBLEM

PROBLEM



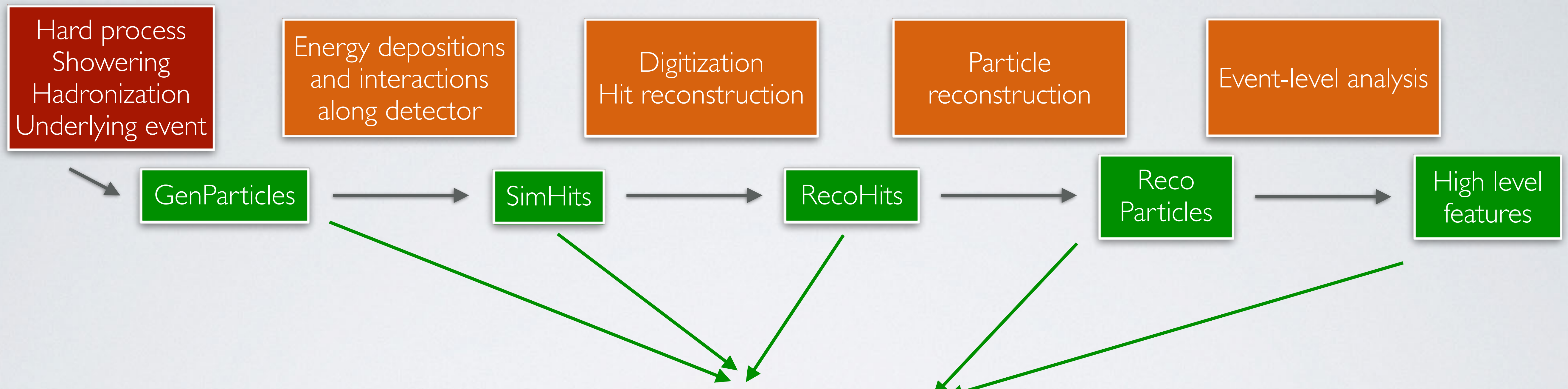
Hard process
Showering
Hadronization
Underlying event
Energy depositions and interactions along detector
Digitization
Hit reconstruction
Particle reconstruction
Event-level analysis
GenParticles
SimHits
RecoHits
Reco Particles
High level features

PROBLEM



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

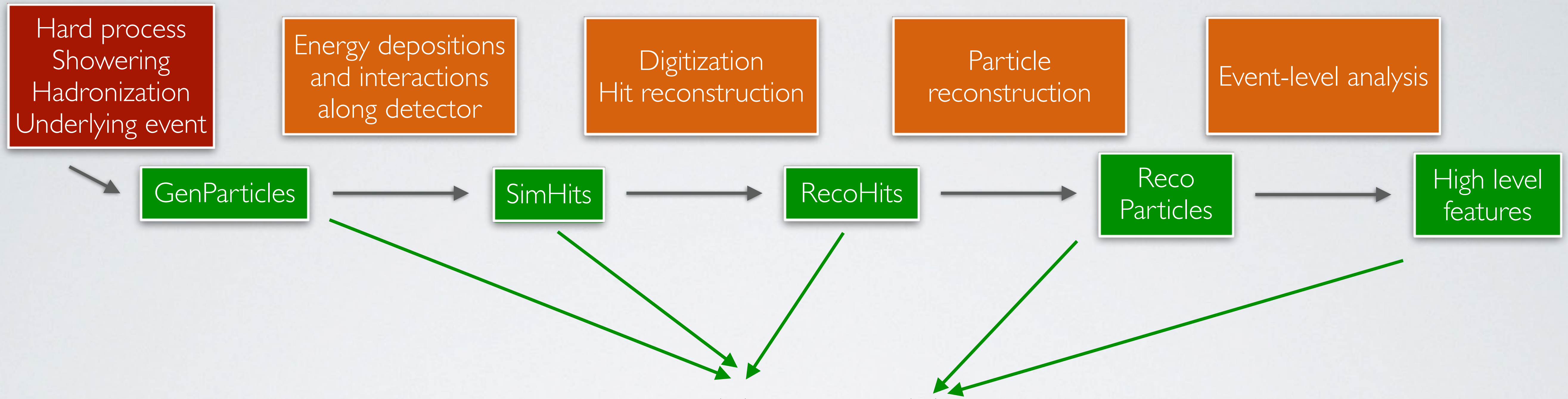
PROBLEM



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

⇒ Multivariate goodness-of-fit (GOF) / two-sample test

PROBLEM

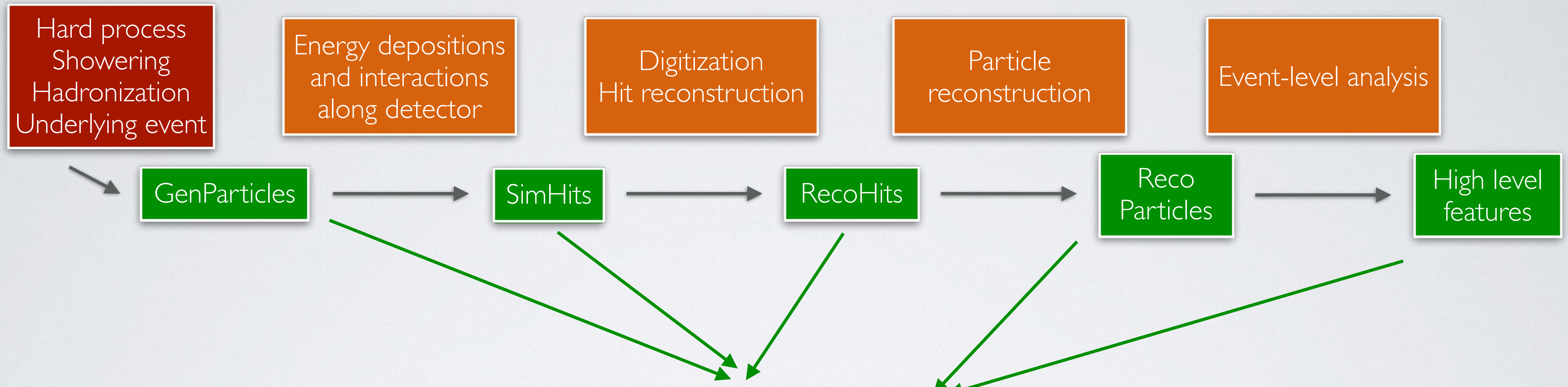


- Want to quantify difference between $p_{\mathrm{real}}(\mathbf{x})$ and $p_{\mathrm{gen}}(\mathbf{x})$ distributions

⇒ Multivariate goodness-of-fit (GOF) / two-sample test

- But no "best" GOF test (Cousins 2016)

PROBLEM



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

⇒ Multivariate goodness-of-fit (GOF) / two-sample test

- But no "best" GOF test (Cousins 2016)
- Need to choose based on the relevant alternative hypotheses

TEST CRITERIA

TEST CRITERIA

- To **trust** generated data, tests should be:

TEST CRITERIA

- To **trust** generated data, tests should be:
  - Sensitive to quality

TEST CRITERIA

- To **trust** generated data, tests should be:
  - Sensitive to quality
  - Sensitive to diversity

TEST CRITERIA

- To **trust** generated data, tests should be:
  - Sensitive to quality
  - Sensitive to diversity
  - Multivariate (for correlations & conditional generation)

TEST CRITERIA

- To **trust** generated data, tests should be:
  - Sensitive to quality
  - Sensitive to diversity
  - Multivariate (for correlations & conditional generation)
  - Interpretable

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:

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

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

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

METHODS

HISTOGRAMS

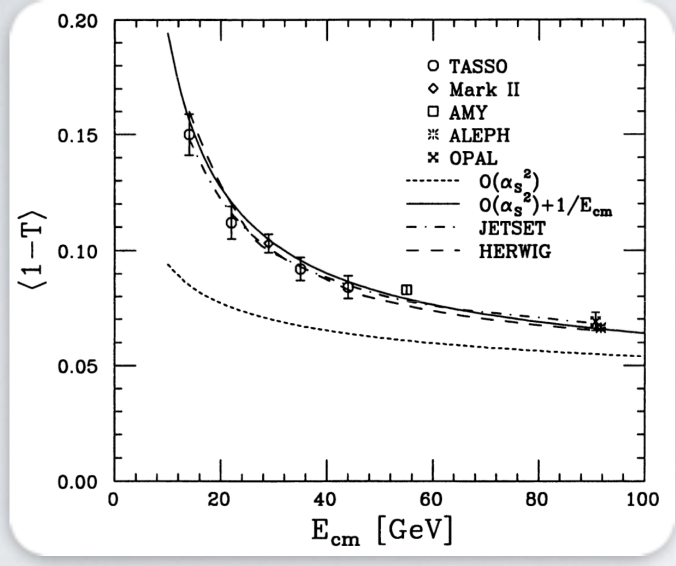
HISTOGRAMS

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

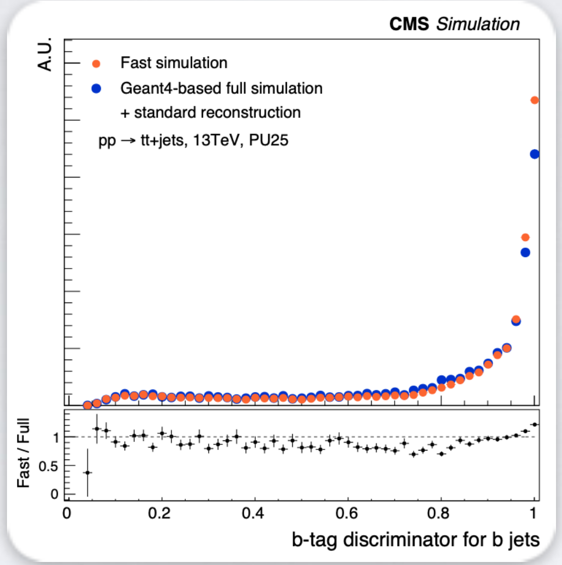
HISTOGRAMS

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

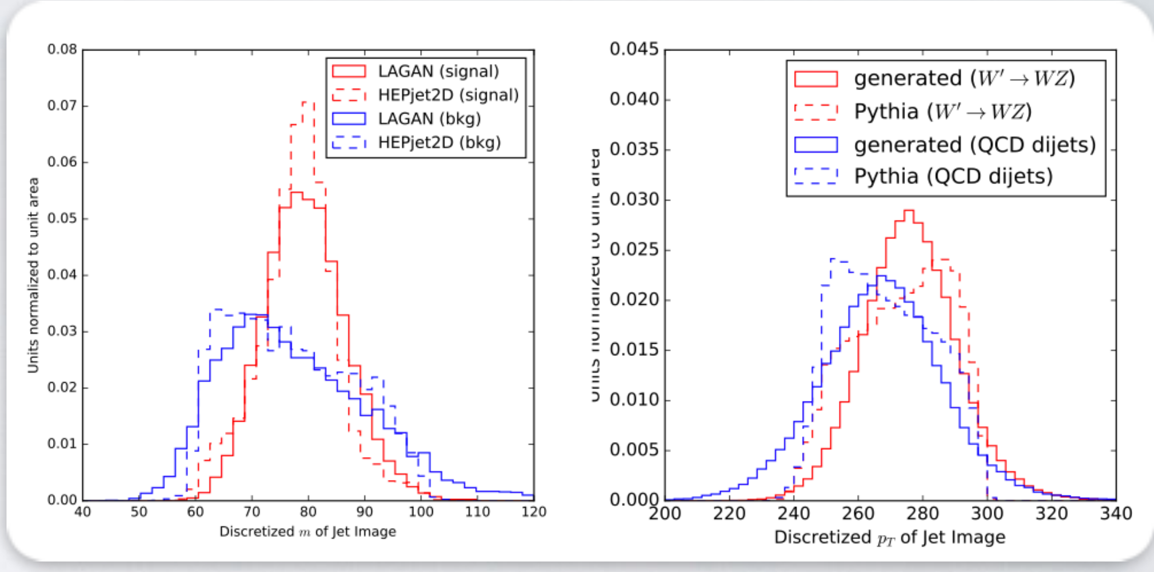
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



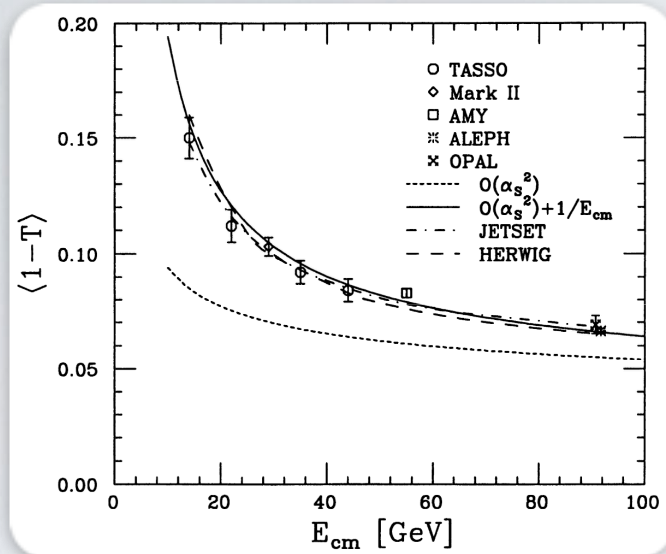
LAGAN (de Oliveira et al '17)



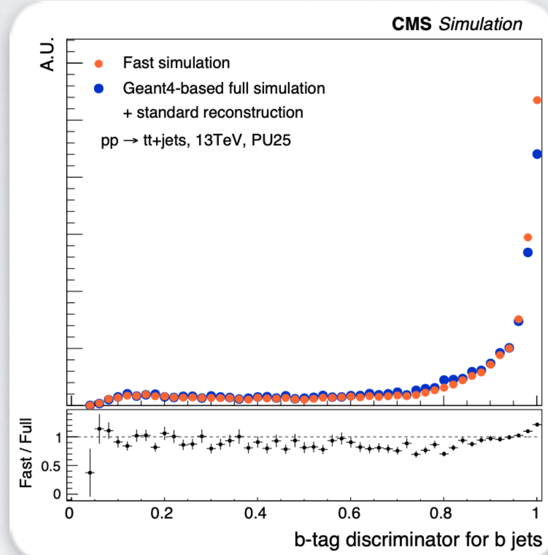
HISTOGRAMS

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

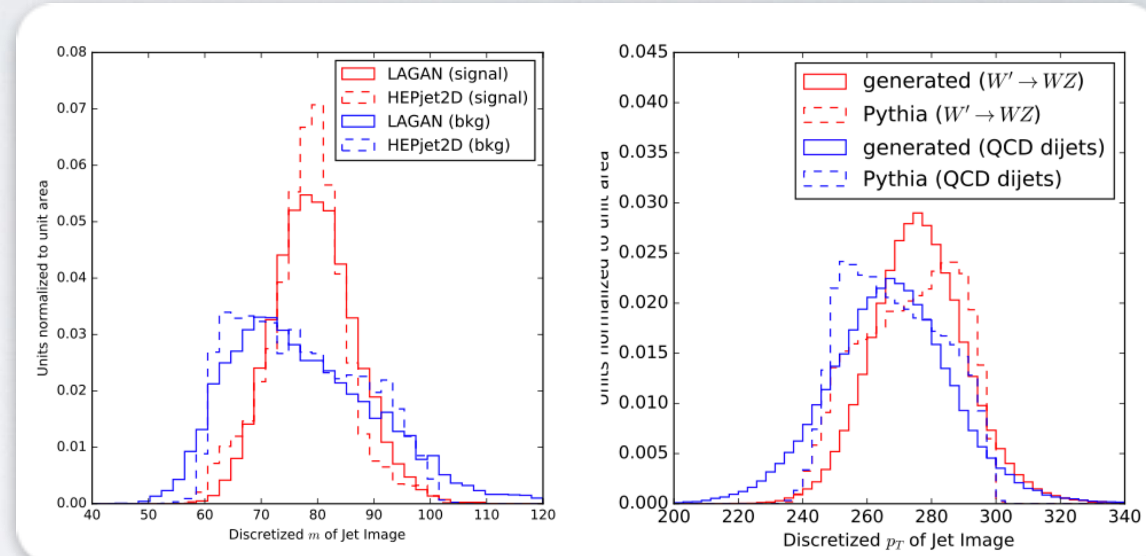
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)

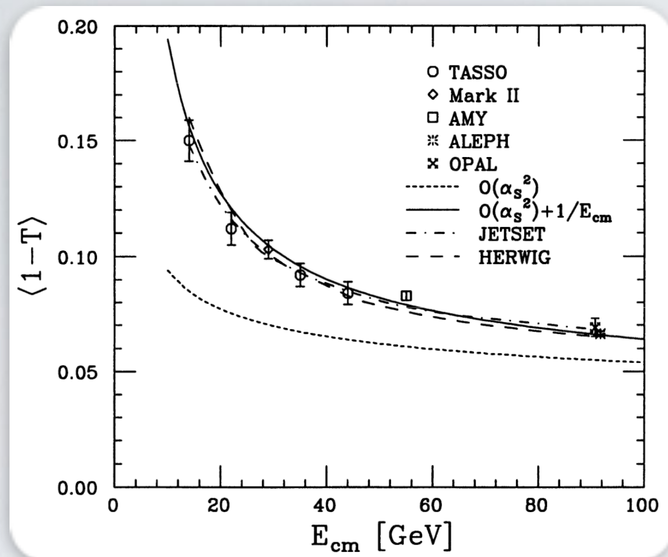


- Valuable insight into physics performance

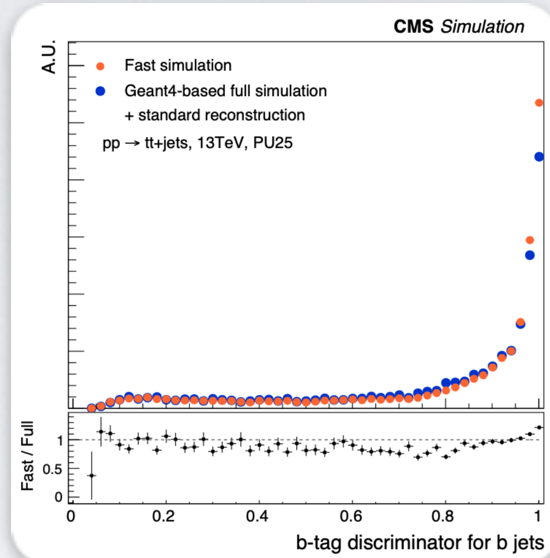
HISTOGRAMS

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

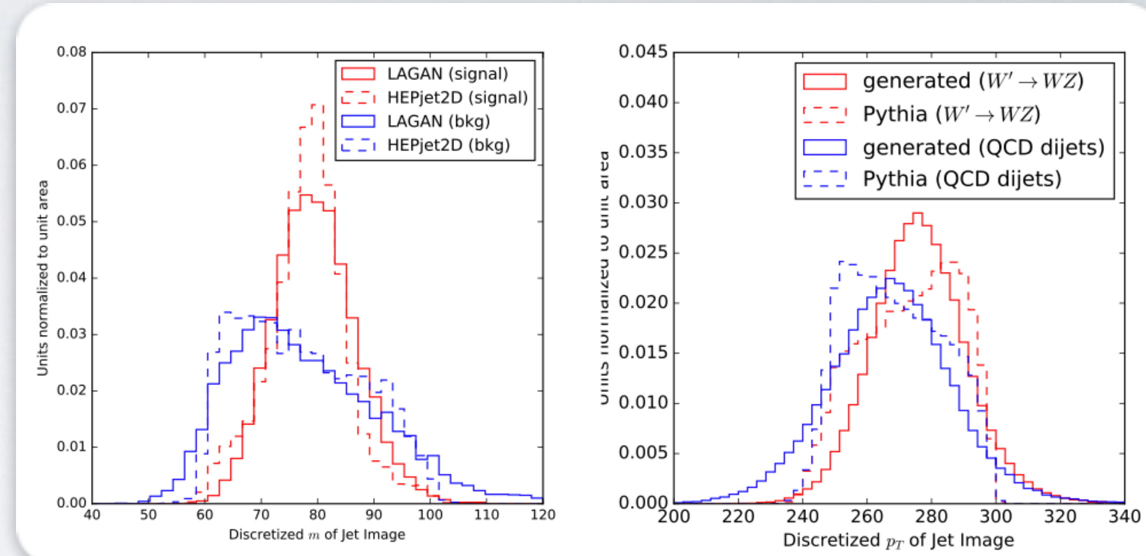
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)



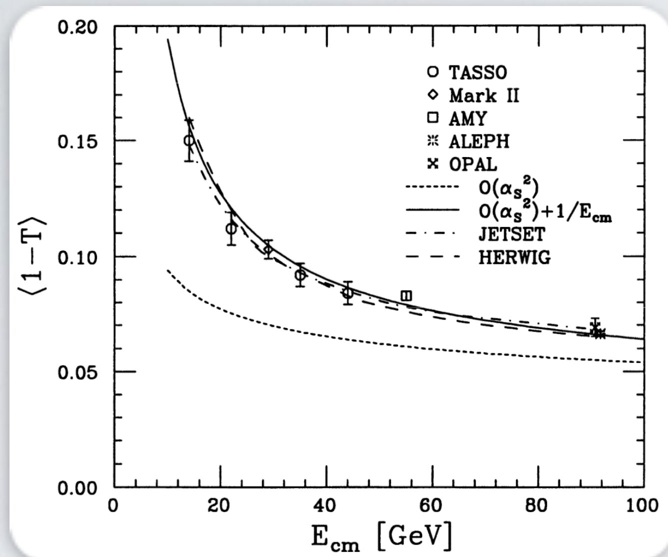
- Valuable insight into physics performance

- Should be quantified

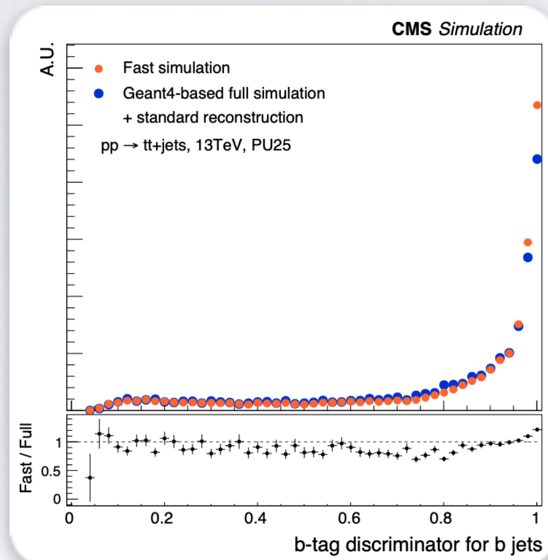
HISTOGRAMS

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

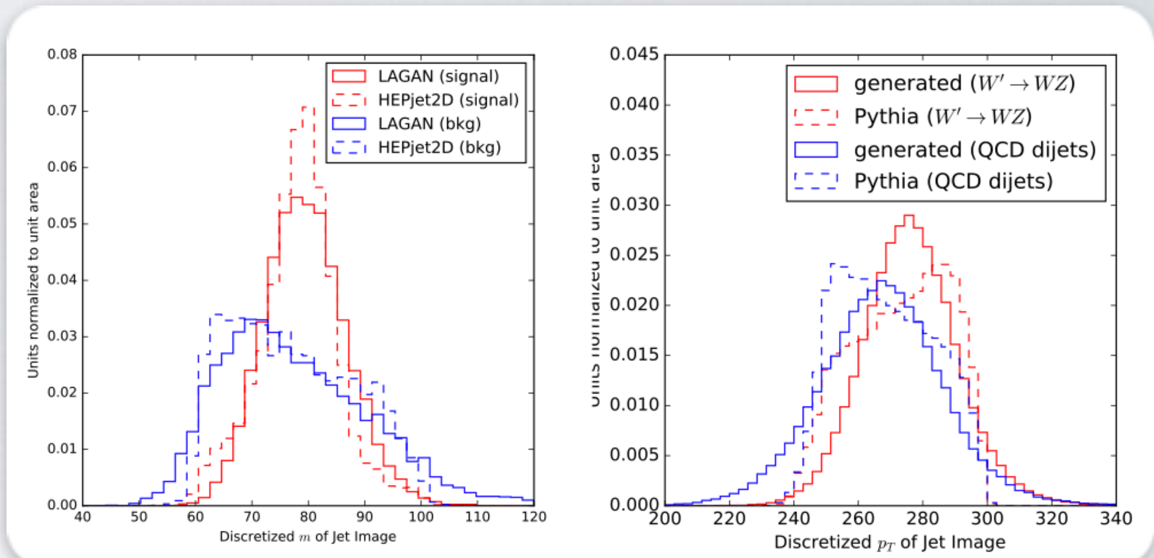
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)

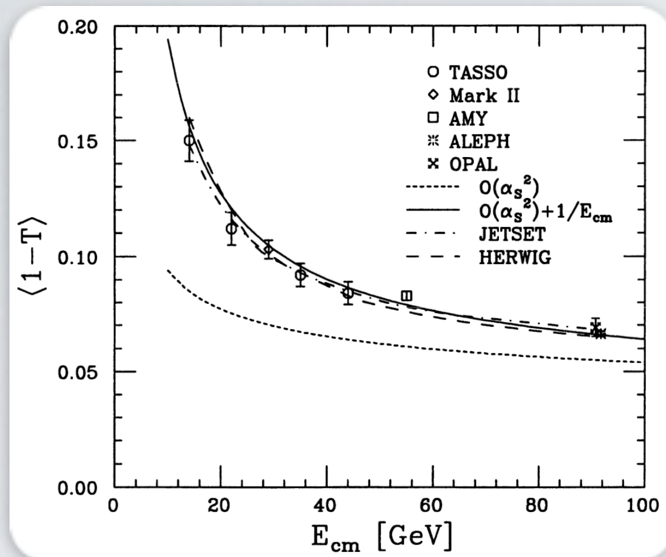


- Valuable insight into physics performance
- Should be quantified
- Cons:

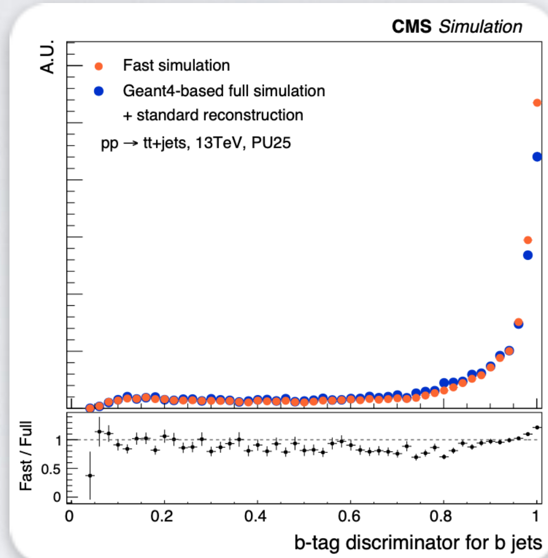
HISTOGRAMS

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

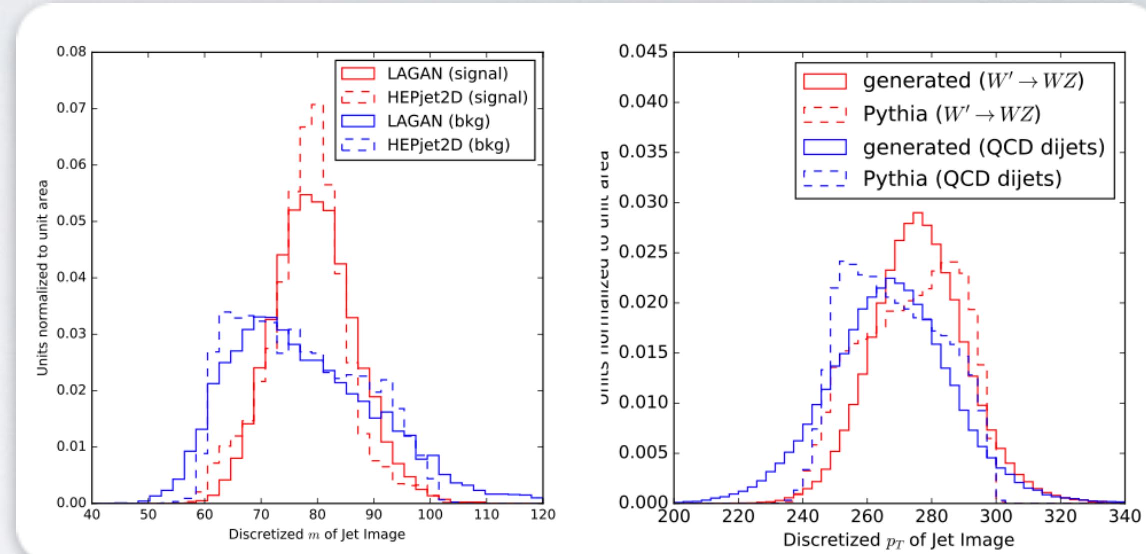
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)

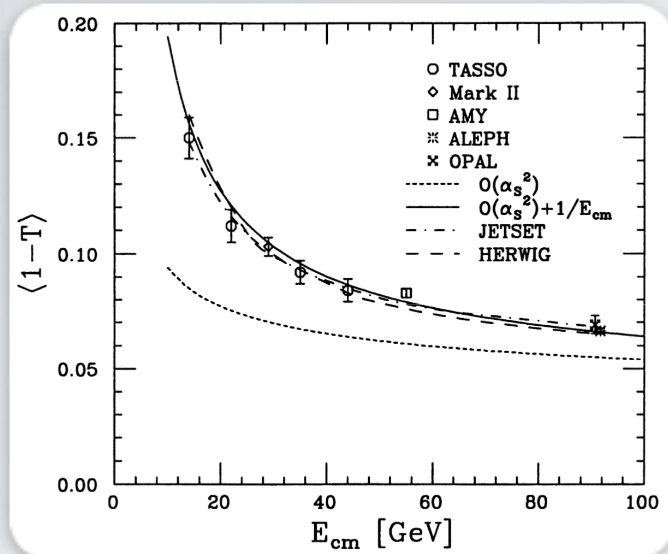


- Valuable insight into physics performance
- Should be quantified
- Cons:
  - Only 1D (curse of dimensionality for multivariate histograms)

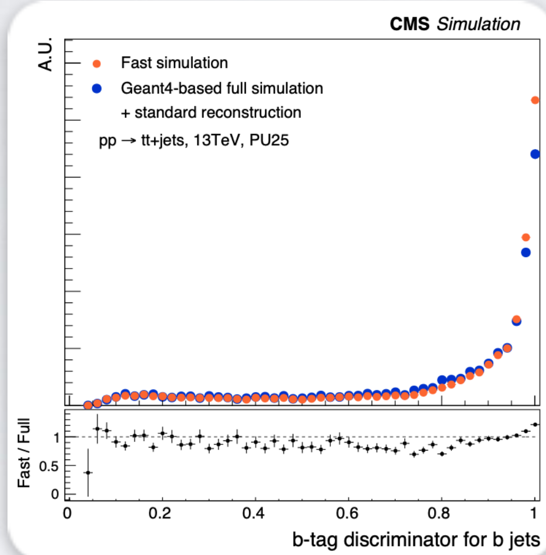
HISTOGRAMS

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

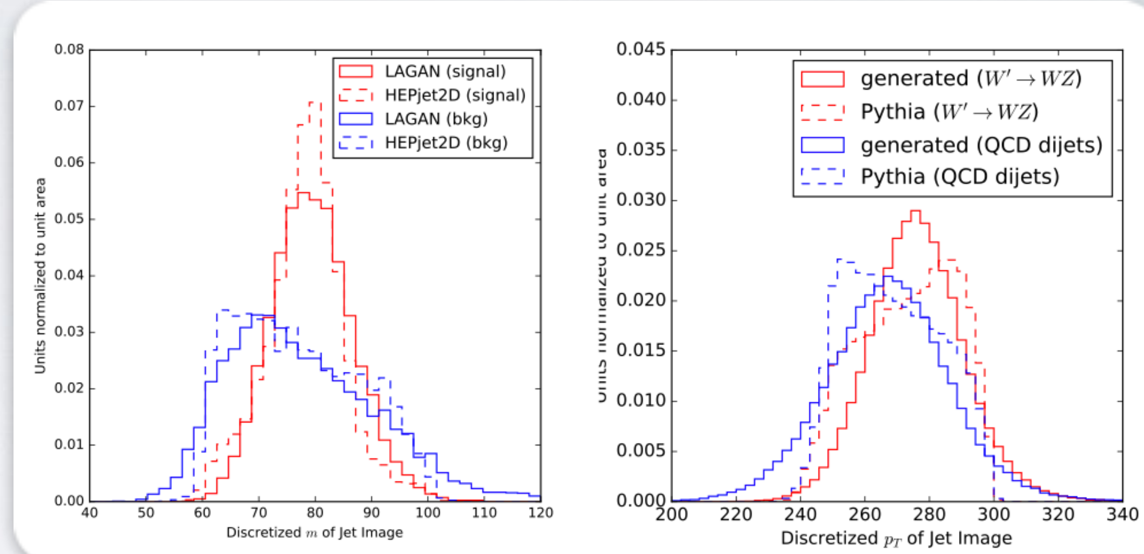
MC generator evaluation (Ellis et al '96)



FastSim (Sekmen '17)



LAGAN (de Oliveira et al '17)

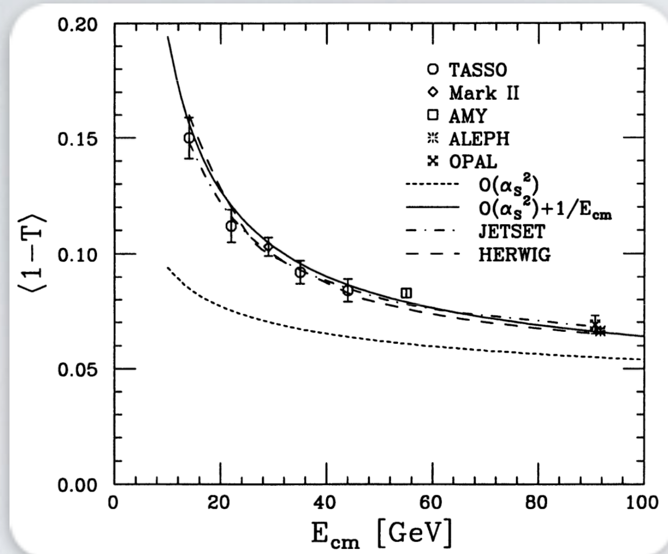


- Valuable insight into physics performance
- Should be quantified
- Cons:
  - Only 1D (curse of dimensionality for multivariate histograms)
  - Binning dependent

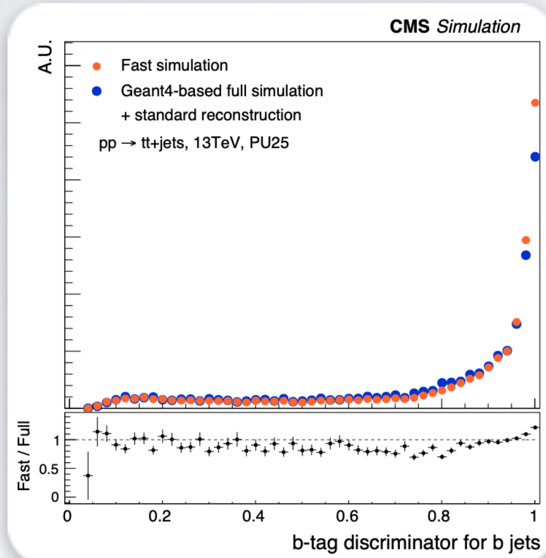
HISTOGRAMS

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

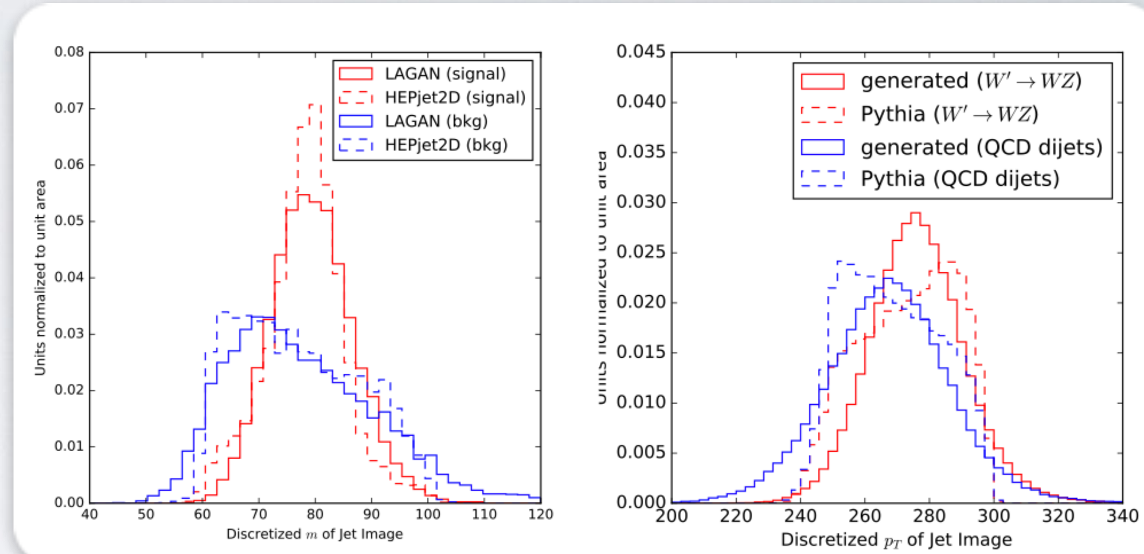
MC generator evaluation (Ellis et al '96)



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

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

Sources 1, 2

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

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

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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

$$\int p_{\text{real}}(x)\, f\left(\frac{p_{\text{real}}(x)}{p_{\text{gen}}(x)} \right) dx$$

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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

KL

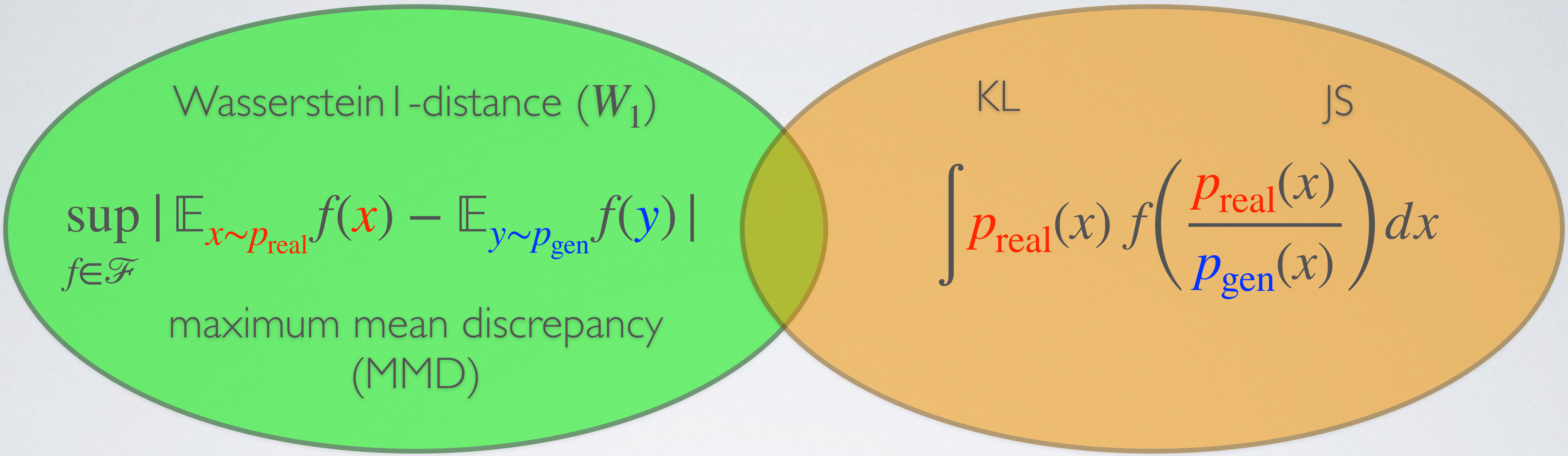
$$\int p_{\text{real}}(x)\, f\left(\frac{p_{\text{real}}(x)}{p_{\text{gen}}(x)}\right) dx$$

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

Sources 1, 2

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

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

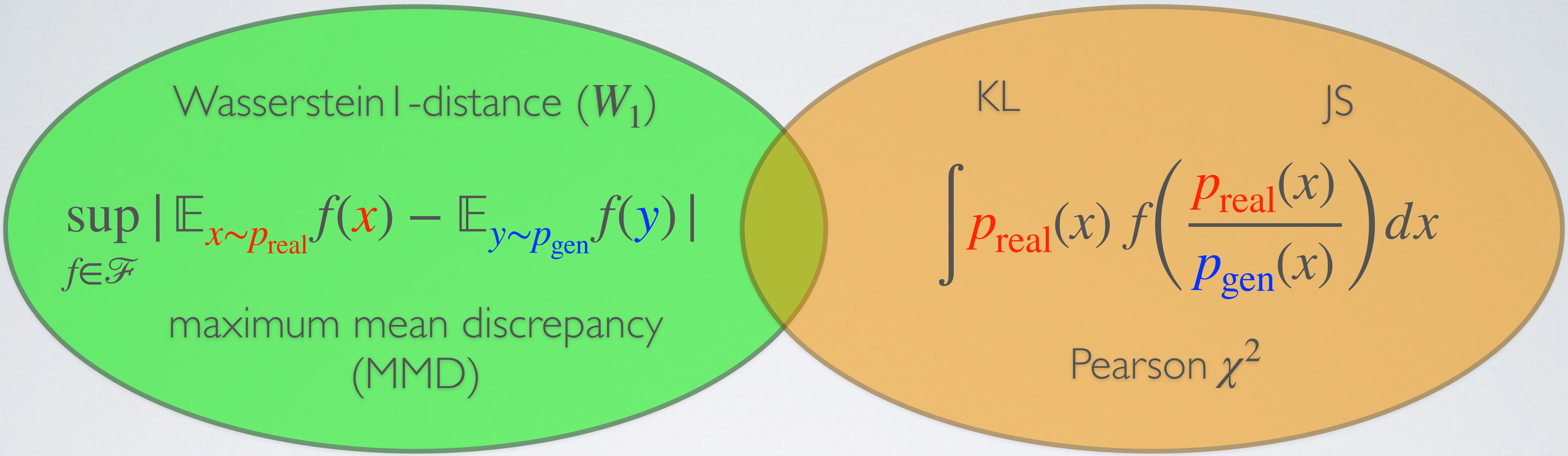


$p_{\text{real}}(\mathbf{x})$ vs $p_{\text{gen}}(\mathbf{x})$

Sources 1, 2

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

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

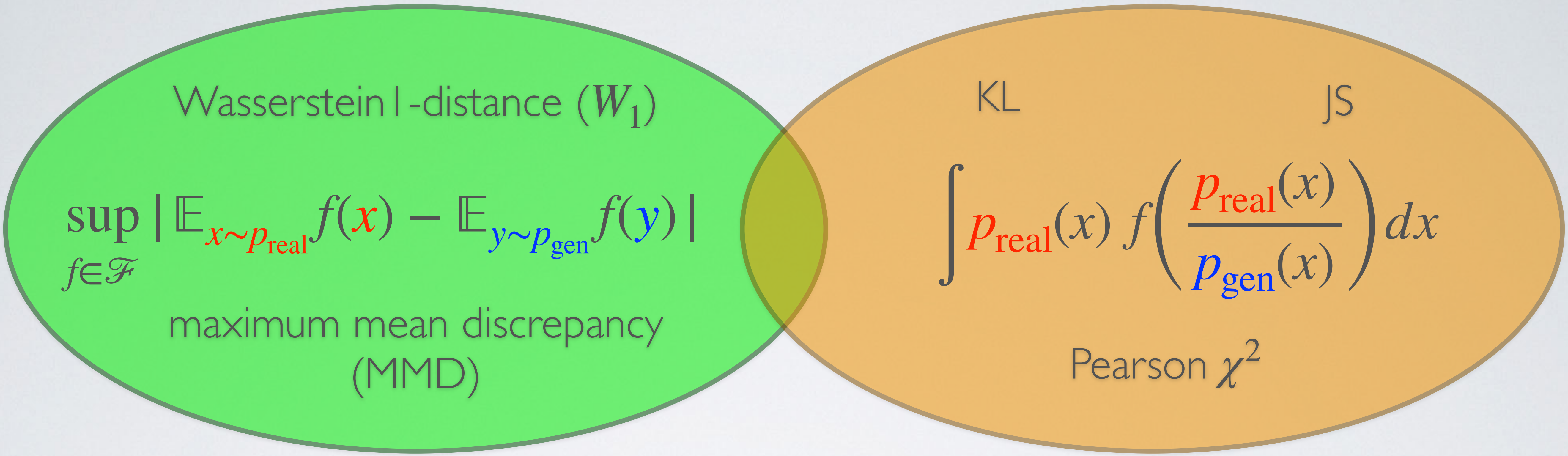


$p_{\mathrm{real}}(\mathbf{x})$ vs $p_{\mathrm{gen}}(\mathbf{x})$

Sources 1, 2

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

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



- IPMs take into account metric space

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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

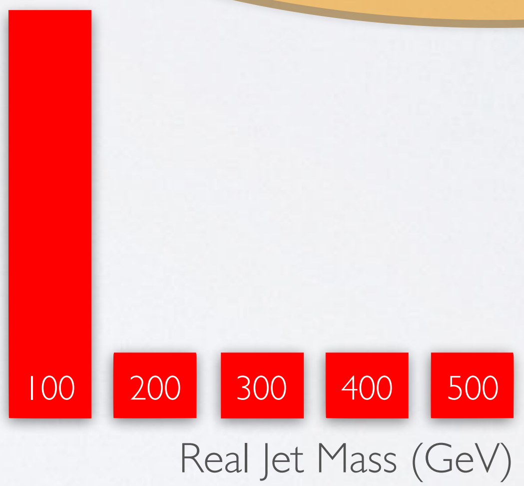
KL

JS

$$\int p_{\text{real}}(x)\, f\left(\frac{p_{\text{real}}(x)}{p_{\text{gen}}(x)}\right) dx$$

Pearson χ^2

- IPMs take into account metric space



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

Sources 1, 2

Integral Probability Metrics $D_{\mathscr{F}}(p_{\mathrm{real}}, p_{\mathrm{gen}})$

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

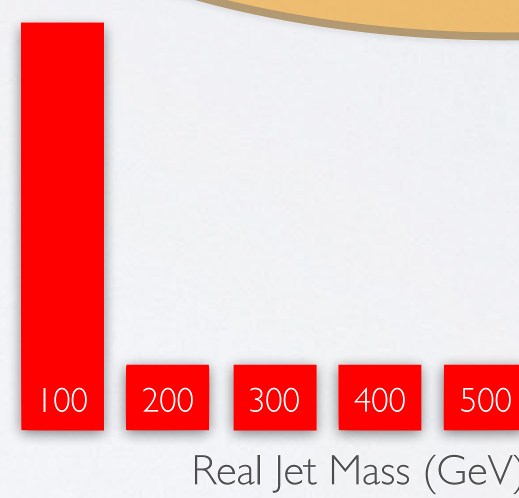
KL

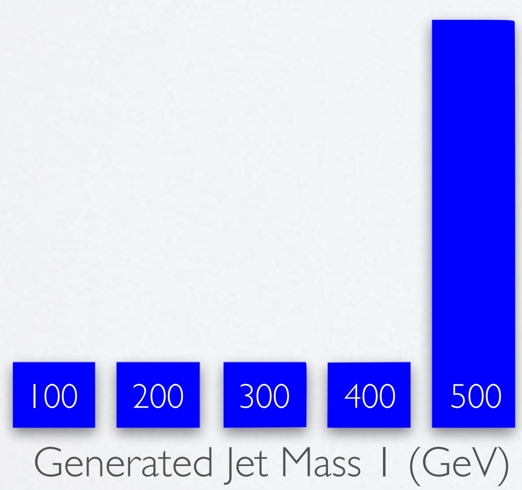
JS

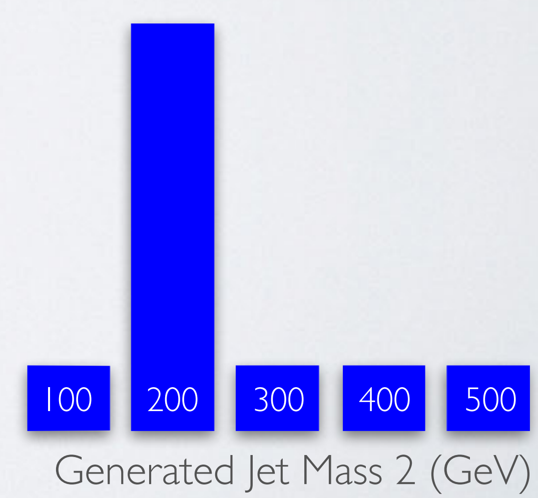
$$\int p_{\mathrm{real}}(x)\, f\left(\frac{p_{\mathrm{real}}(x)}{p_{\mathrm{gen}}(x)}\right) dx$$

Pearson χ^2

- IPMs take into account metric space







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

Sources 1, 2

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

Wasserstein 1-distance (W_1)

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

maximum mean discrepancy (MMD)

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

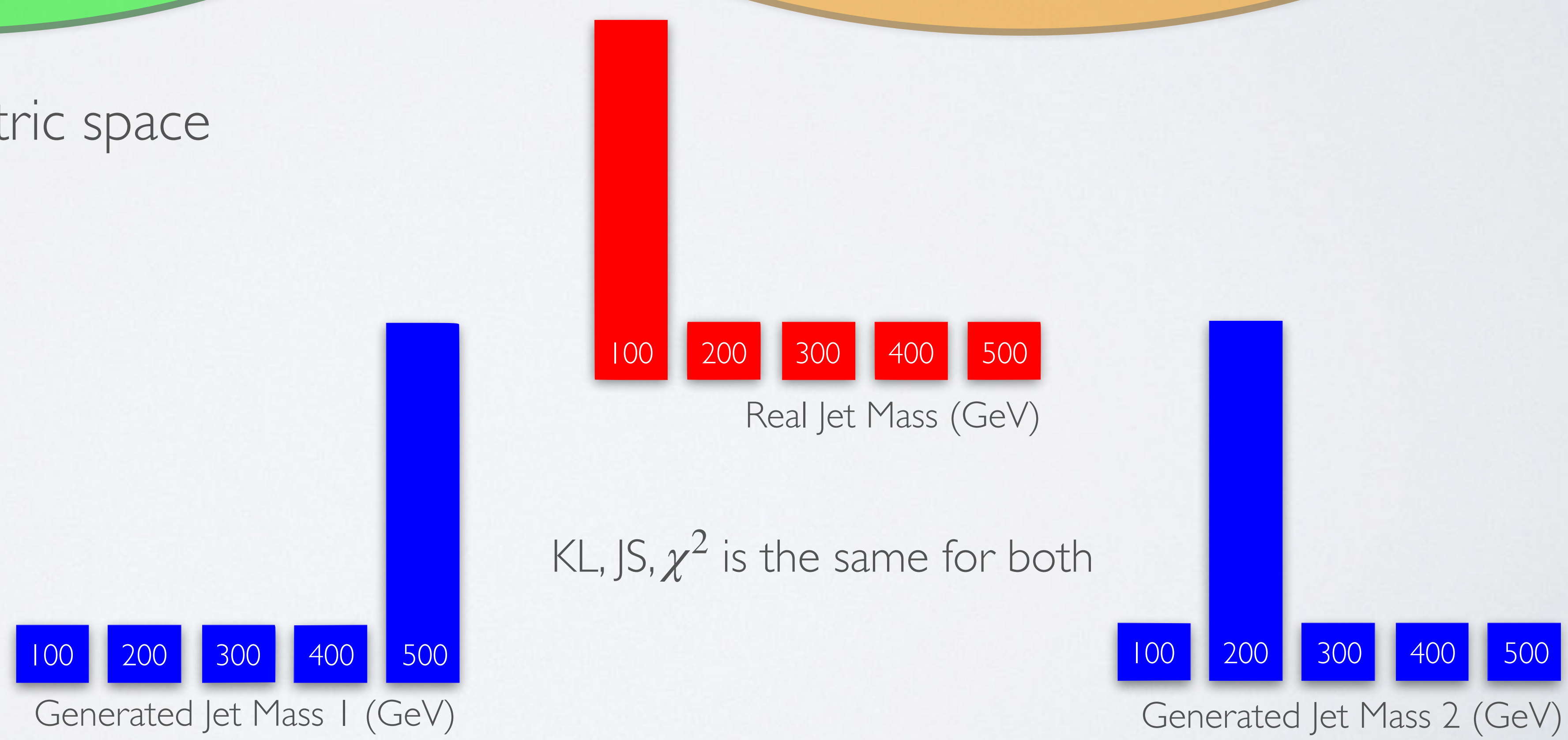
KL

JS

$$\int p_{\text{real}}(x)\, f\left(\frac{p_{\text{real}}(x)}{p_{\text{gen}}(x)}\right) dx$$

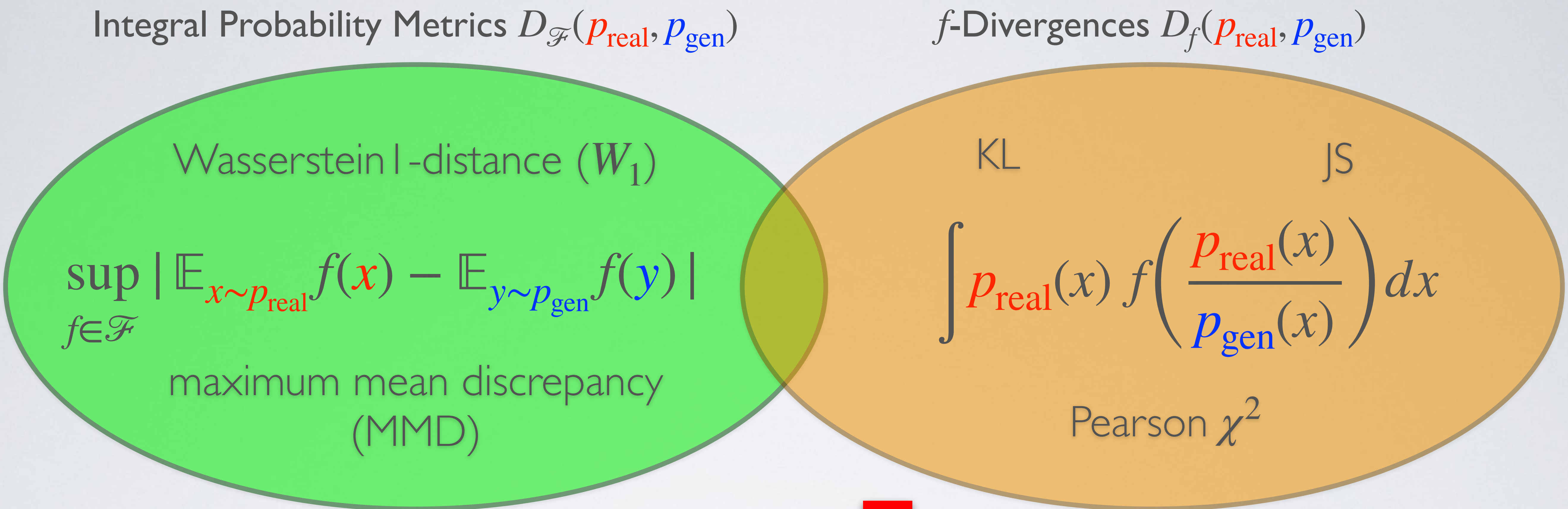
Pearson χ^2

- IPMs take into account metric space

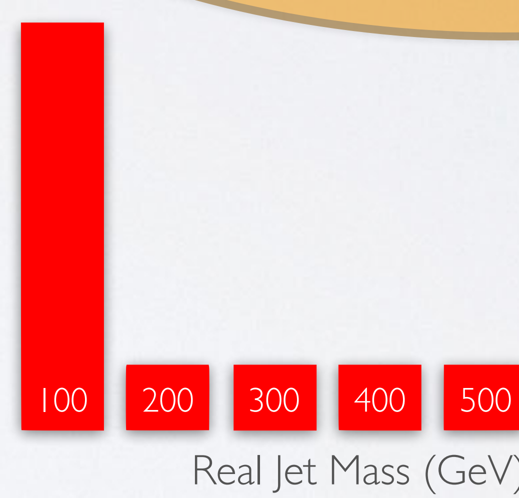


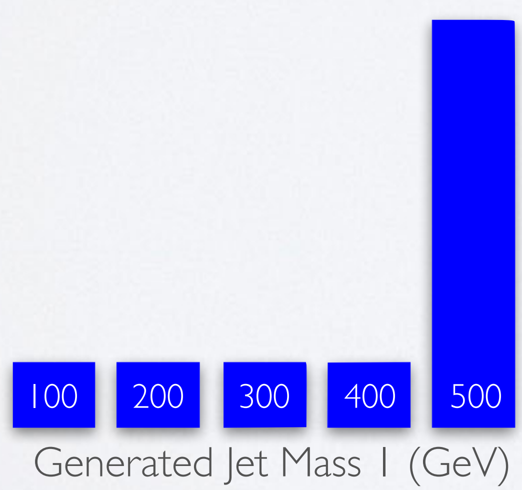
$p_{\text{real}}(\mathbf{x})$ vs $p_{\text{gen}}(\mathbf{x})$

Sources 1, 2

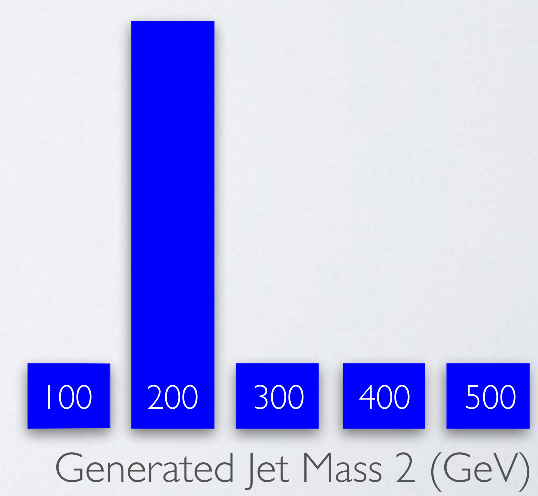


- IPMs take into account metric space
- More useful for comparing generative models





KL, JS, χ^2 is the same for both

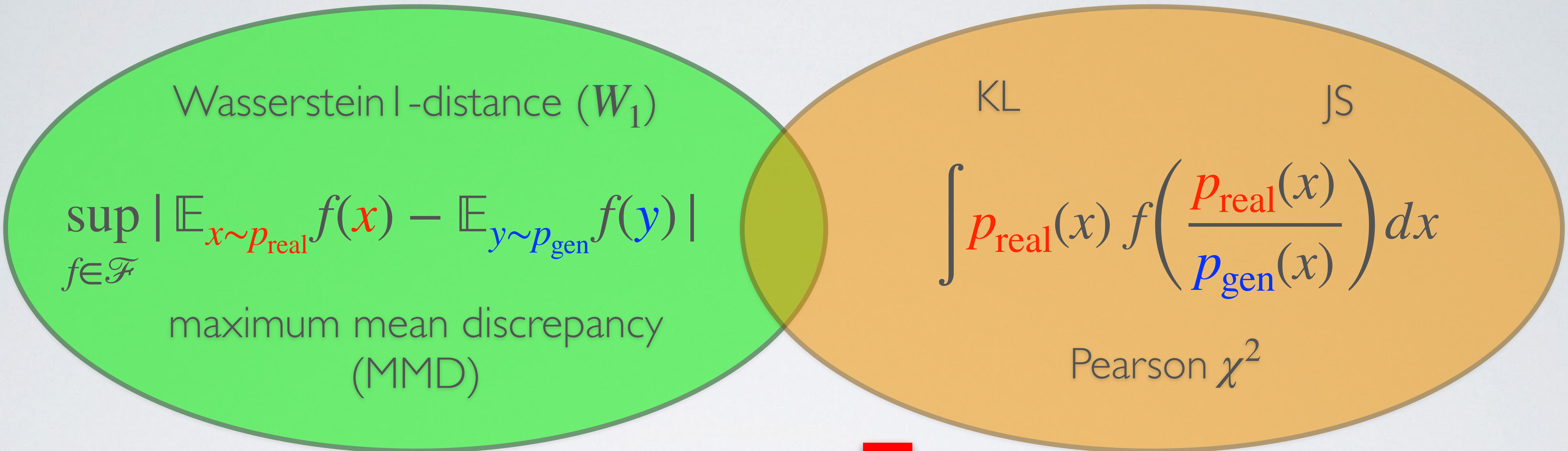


$p_{\mathrm{real}}(\mathbf{x})$ vs $p_{\mathrm{gen}}(\mathbf{x})$

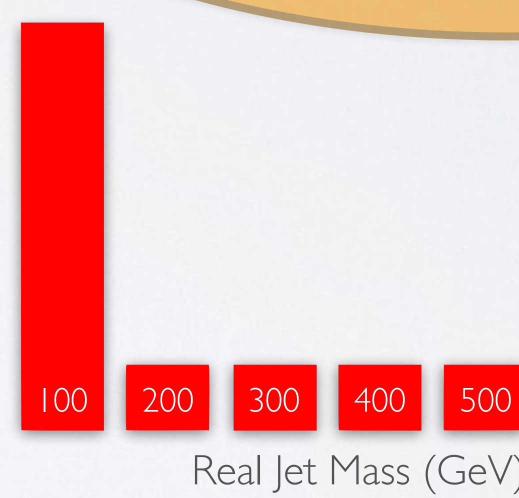
Sources 1, 2

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

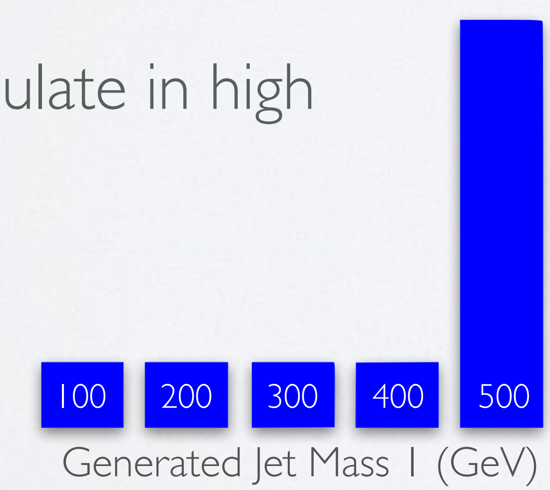
f-Divergences $D_f(p_{\mathrm{real}}, p_{\mathrm{gen}})$

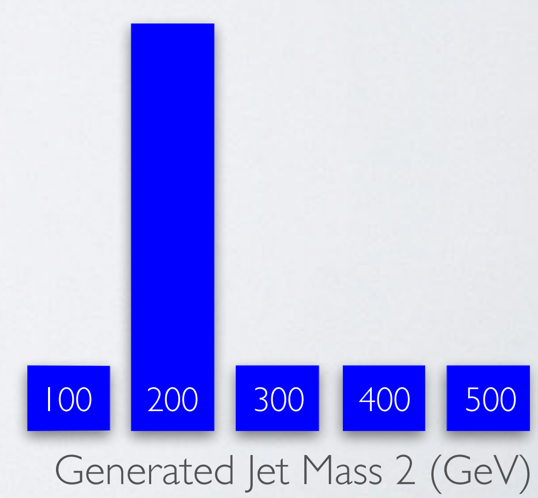


- IPMs take into account metric space
- More useful for comparing generative models
- And more efficient to calculate in high dimensions



KL, JS, χ^2 is the same for both





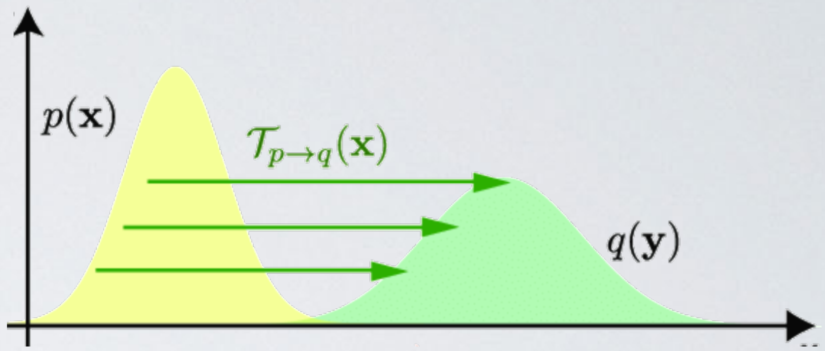
MORE ON IPMS

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

MORE ON IPMS

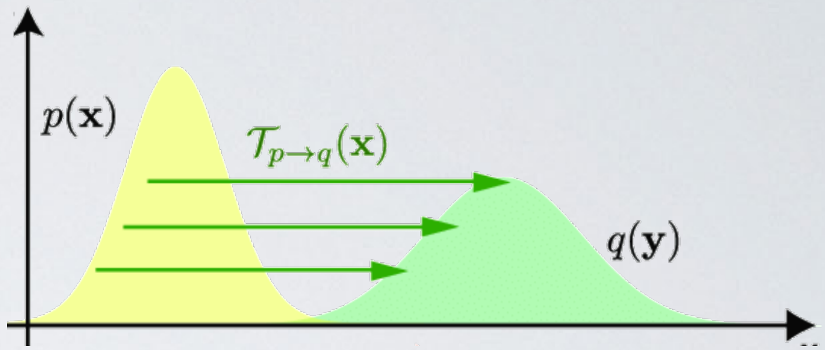
$$\sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)|$$

- Wasserstein distance (W_1)
  ($\mathcal{F}$ is all K-Lipschitz functions)



MORE ON IPMS

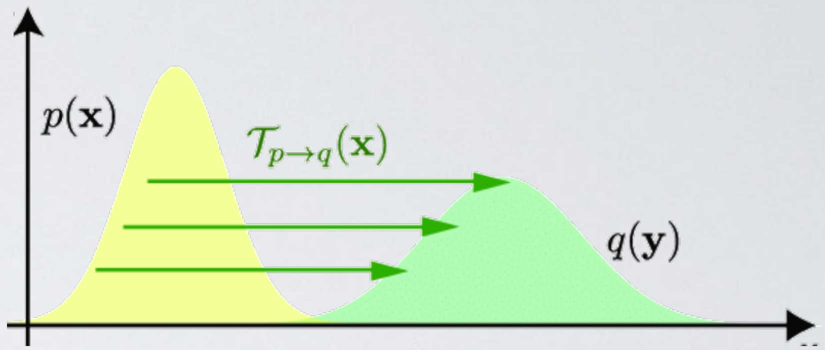
$$\sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)|$$



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

MORE ON IPMS

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

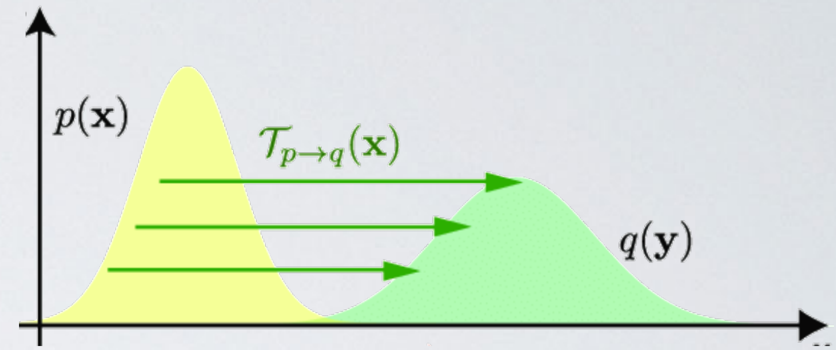


- Wasserstein distance (W_1)
  ($\mathscr{F}$ is all K-Lipschitz functions)
  - Sensitive to quality, diversity; but biased and slow convergence
- Fréchet Gaussian distance (FGD)

MORE ON IPMS

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

- Wasserstein distance (W_1)
  ($\mathcal{F}$ is all K-Lipschitz functions)



  - Sensitive to quality, diversity; but biased and slow convergence

- Fréchet Gaussian distance (FGD)

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

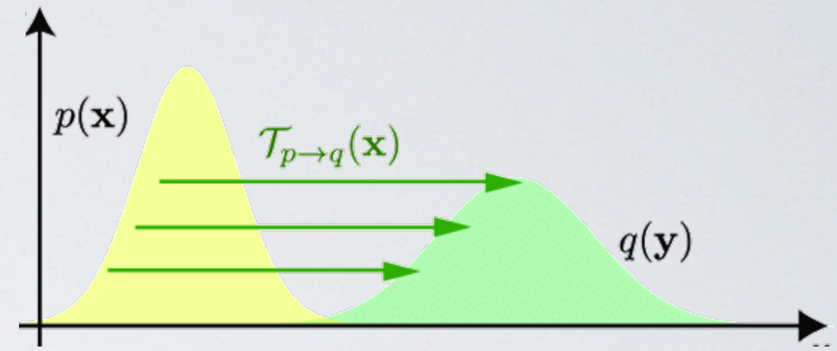
$\{\mathbf{x}_{\mathrm{real}}\}$ $\{\mathbf{x}_{\mathrm{gen}}\}$

  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations

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)

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

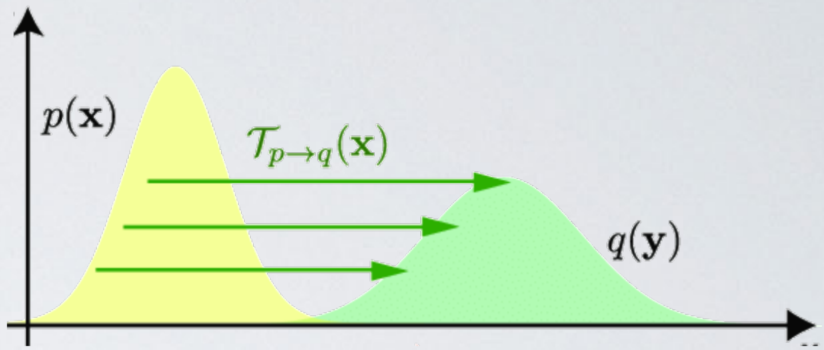
$\{\mathbf{x}_{\text{real}}\}$ $\{\mathbf{x}_{\text{gen}}\}$

  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID), efficient, sensitive to quality + diversity; but access only up to 2nd order moments

MORE ON IPMS

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

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



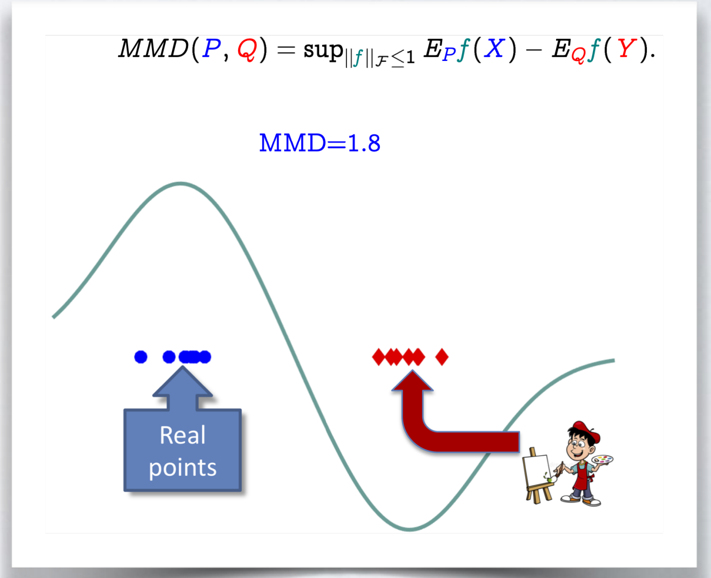
- Fréchet Gaussian distance (FGD)
    - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
    - Standard in computer vision (FID), efficient, sensitive to quality + diversity; but access only up to 2nd order moments

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

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

- Maximum Mean Discrepancy (MMD)
  ($\mathcal{F}$ is unit ball in reproducing Kernel Hilbert space (RKHS) for a chosen kernel $k(x, y)$)

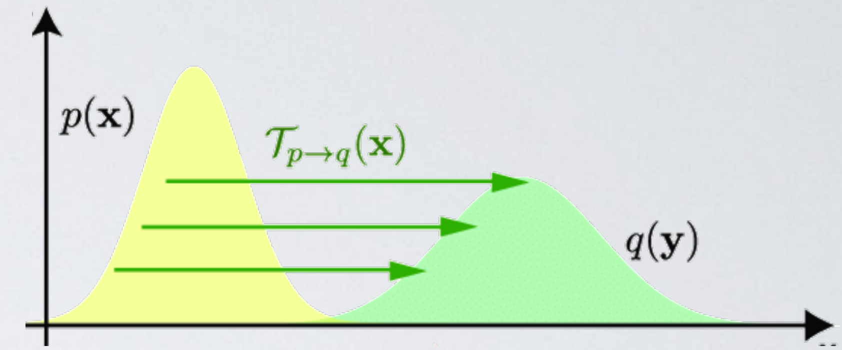
Gretton 2020



MORE ON IPMS

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

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



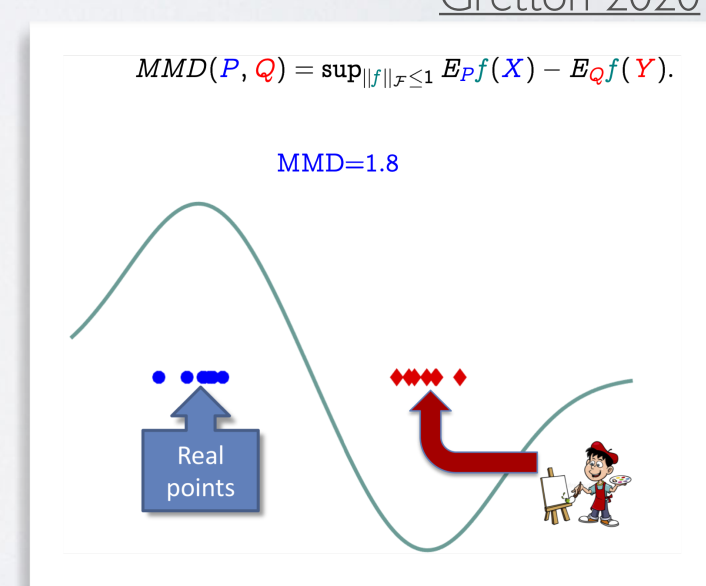
- Fréchet Gaussian distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID), efficient, sensitive to quality + diversity; but access only up to 2nd order moments

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

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

- 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

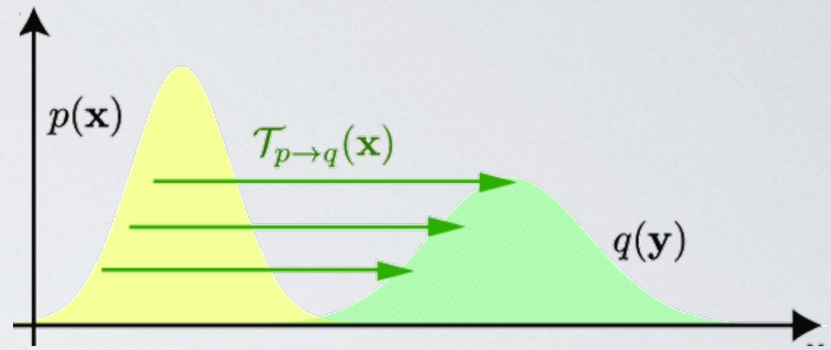
Gretton 2020



MORE ON IPMS

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

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



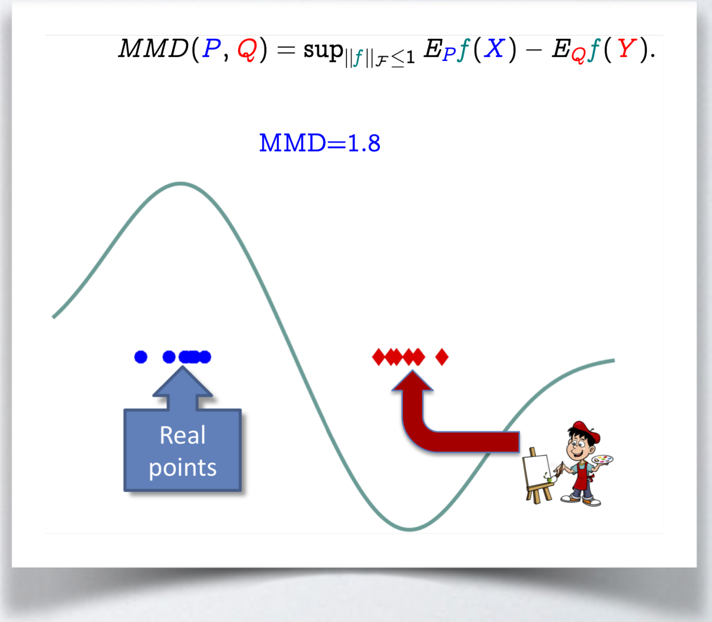
- Fréchet Gaussian distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID), efficient, sensitive to quality + diversity; but access only up to 2nd order moments

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

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

- 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, used in computer vision (KID) but depends on kernel

Gretton 2020



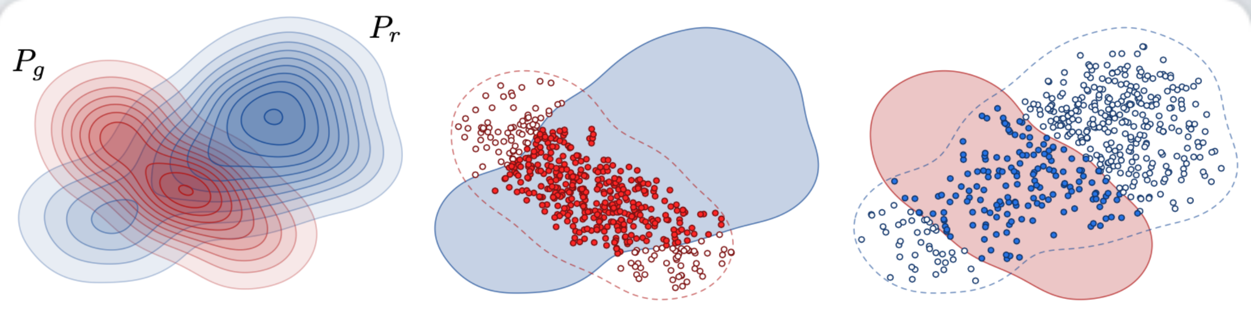
MORE METRICS

MORE METRICS

- Precision and recall (Kynkäänniemi et al 2019)

MORE METRICS

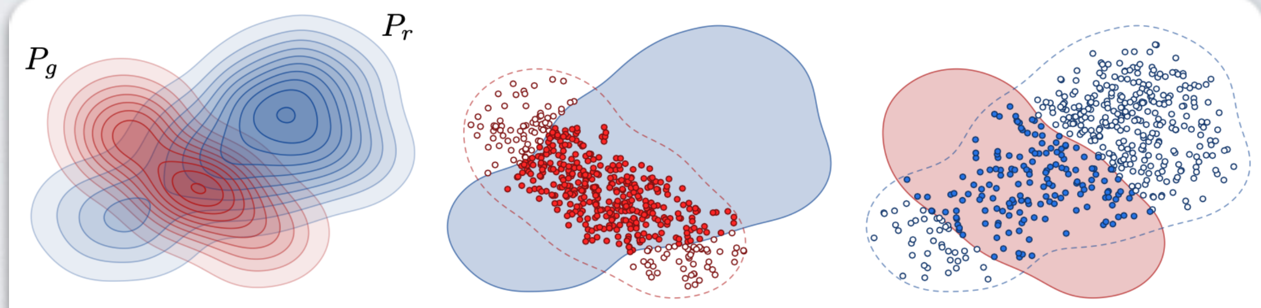
- Precision and recall (Kynkäänniemi et al 2019)
    - Estimate real and generated manifold



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

MORE METRICS

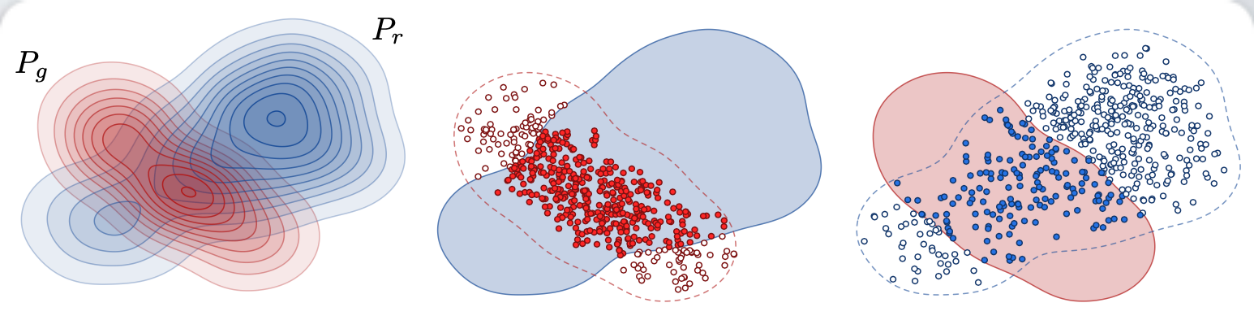
- Precision and recall (Kynkäänniemi et al 2019)
    - Estimate real and generated manifold
    - Can disentangle quality and diversity



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

MORE METRICS

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

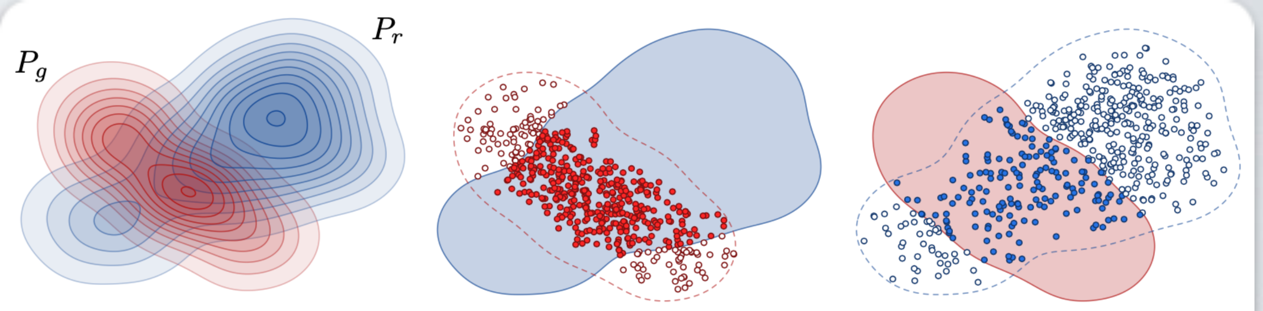


(a) Example distributions (b) Precision (c) Recall

- Classifier-based metrics: train a classifier between real and generated data
  Friedman 2003, Paz and Oquab 2017 (C2ST), Krause and Shih (2021)

MORE METRICS

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

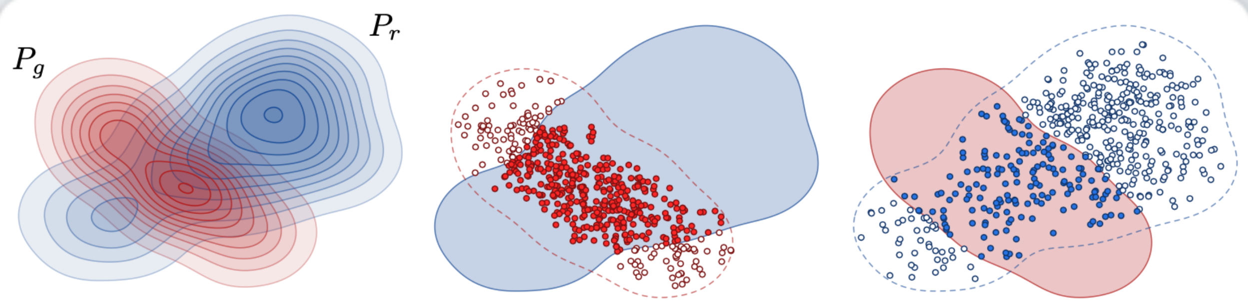


(a) Example distributions (b) Precision (c) Recall

- 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

MORE METRICS

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

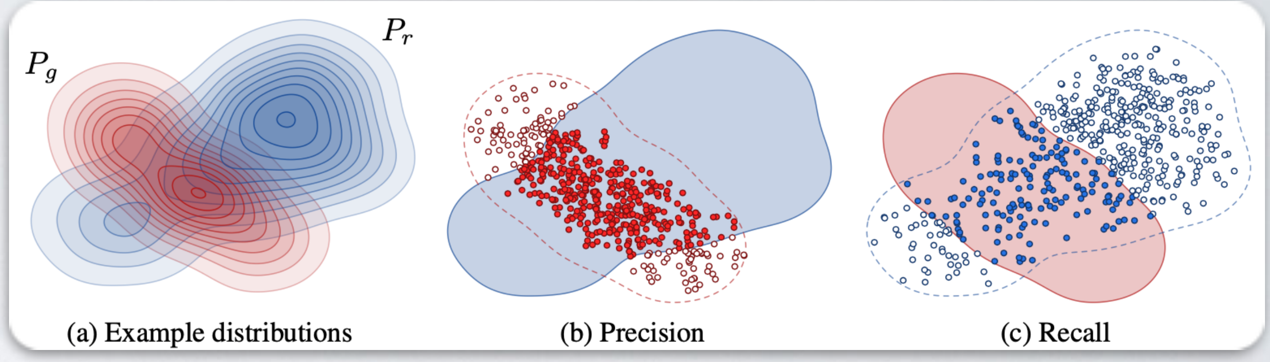


(a) Example distributions (b) Precision (c) Recall

- 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

MORE METRICS

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



- 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

FEATURE SELECTION

FEATURE SELECTION

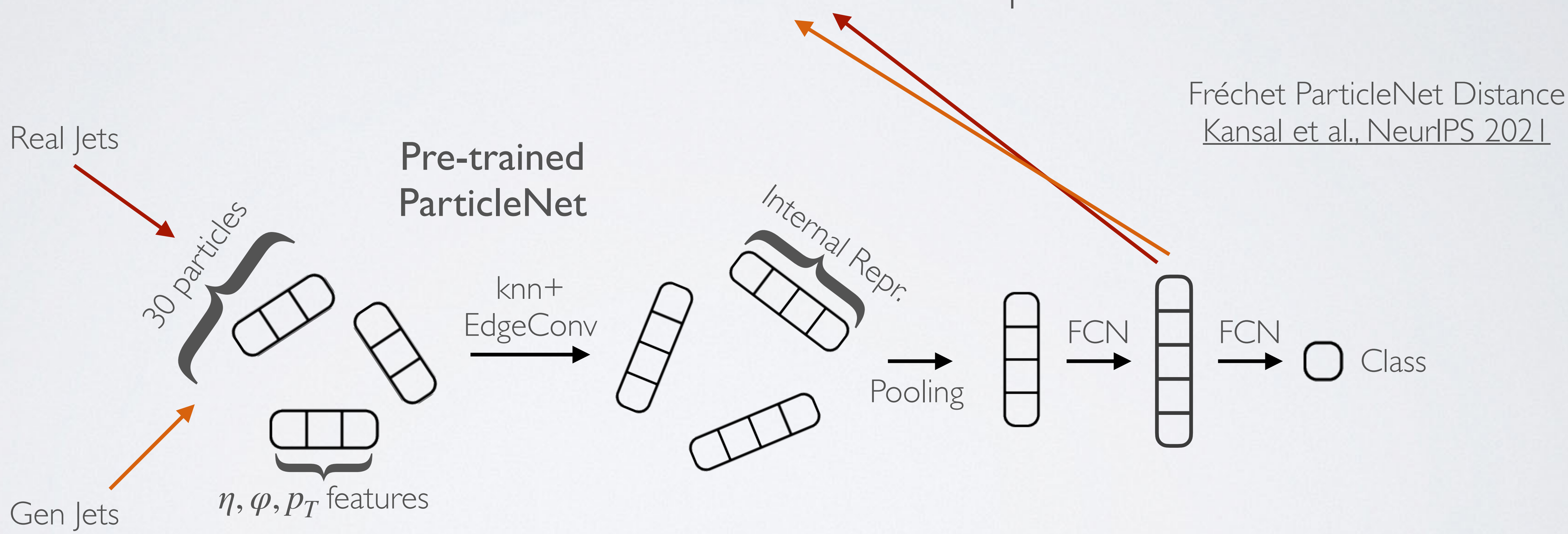
- Typically raw data (particle / hit features) is very high dimensional

FEATURE SELECTION

- Typically raw data (particle / hit features) is very high dimensional
- Not necessarily what we care about

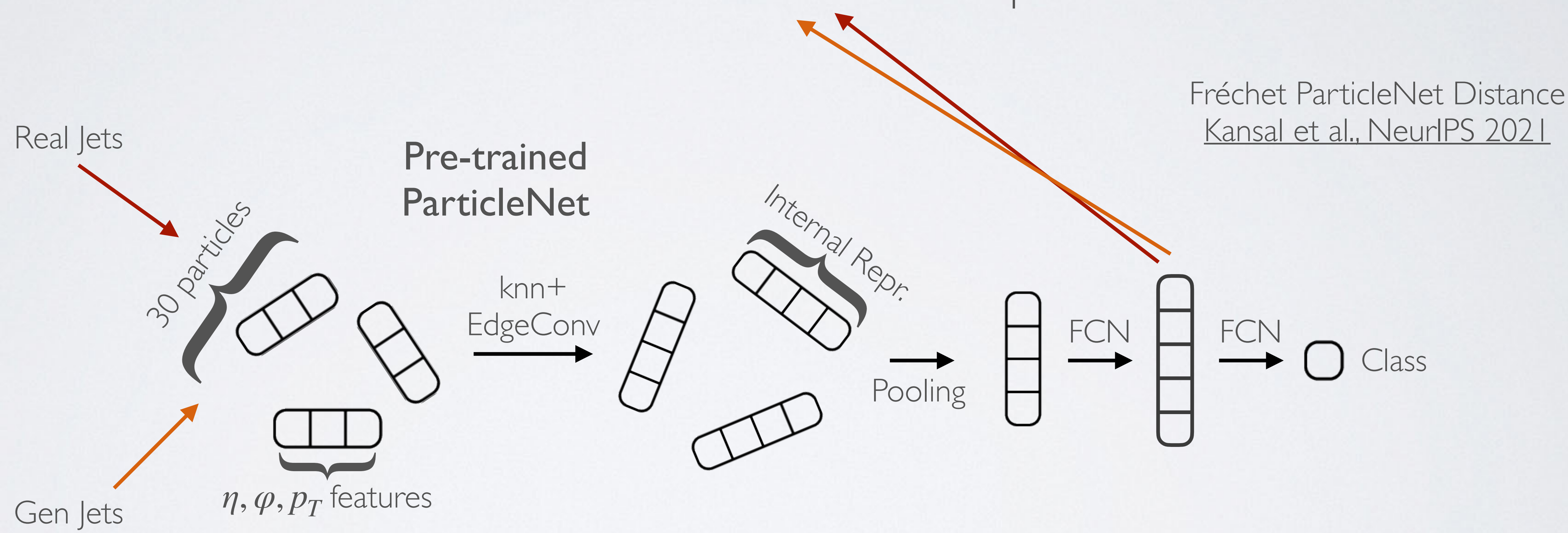
FEATURE SELECTION

- 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



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

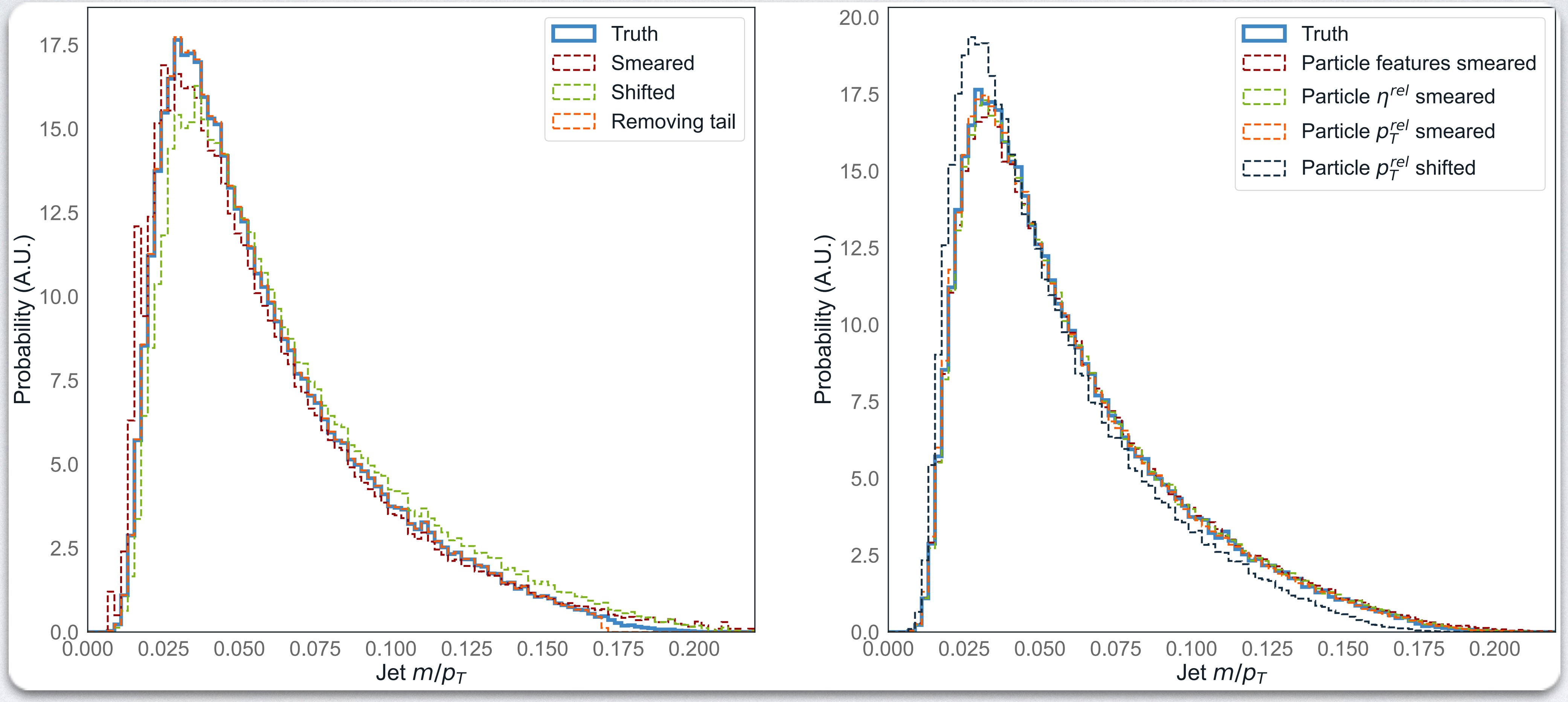
JET DISTRIBUTIONS

- Sample of gluon jets to test sensitivity of metrics

RK et al. 2022

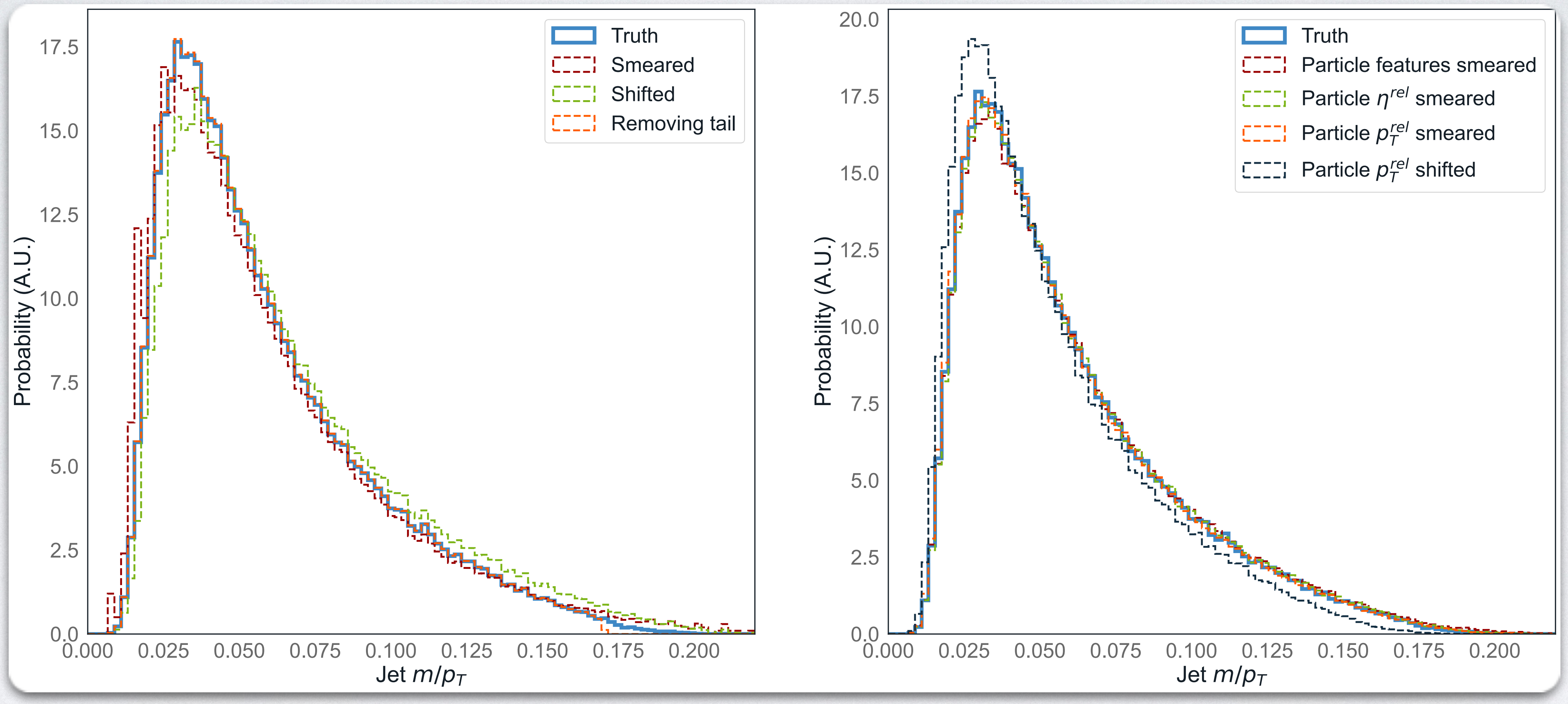
JET DISTRIBUTIONS

- Sample of gluon jets to test sensitivity of metrics
- We distort true distribution by: 1) Re-weighting in mass + 2) Smearing/shifting particle feature



JET DISTRIBUTIONS

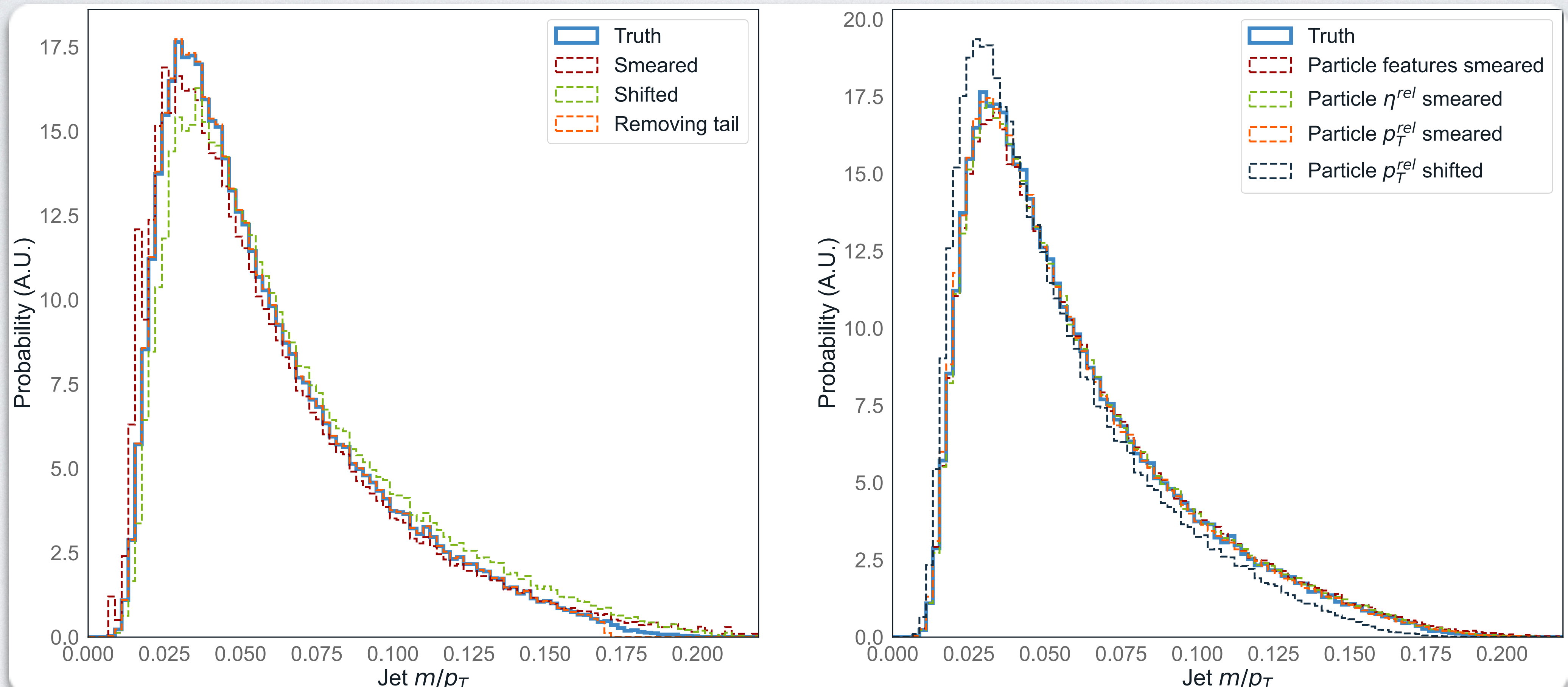
- Sample of gluon jets to test sensitivity of metrics
- We distort true distribution by: 1) Re-weighting in mass + 2) Smearing/shifting particle feature



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

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 feature



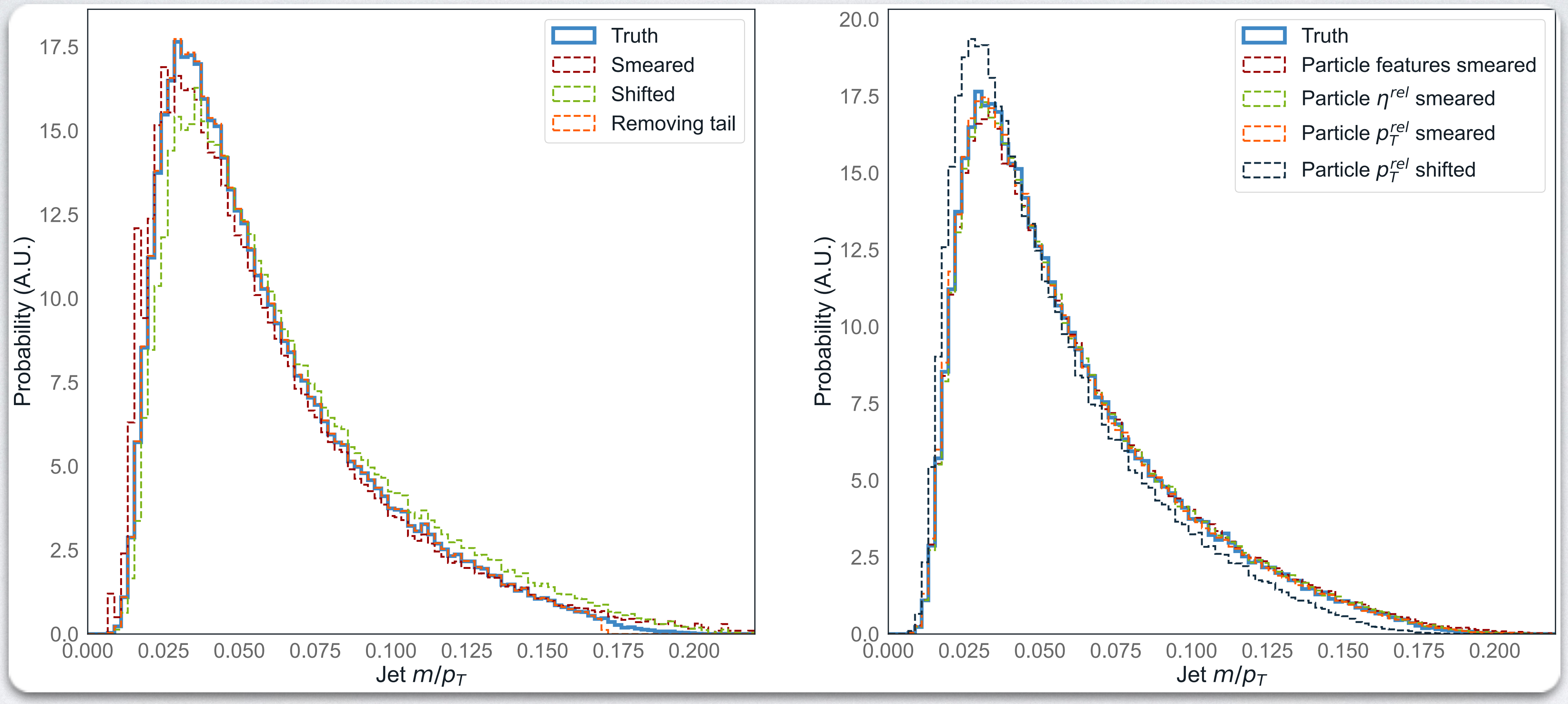
- We look at sensitivity of metrics to distortions, using:

1. Energy Flow Polynomials (EFPs) (d ≤ 4)

RK et al. 2022

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 feature



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

1. Energy Flow Polynomials (EFPs) ($d \leq 4$)
2. ParticleNet activations

RESULTS

RK et al. 2022

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
$W_1^M \times 10^3$								
Sign.								
Wasserstein EFP								
Sign.								
FGD_∞ EFP $\times 10^3$								
Sign.								
MMD EFP $\times 10^3$								
Sign.								
Precision EFP								
Sign.								
Recall EFP								
Sign.								
Wasserstein PN								
Sign.								
FGD_∞ PN $\times 10^3$								
Sign.								
MMD PN $\times 10^3$								
Sign.								
Precision PN								
Sign.								
Recall PN								
Sign.								
Classifier LLF AUC								
Classifier HLF AUC								

RESULTS

Most sensitive metric per distribution in bold

RK et al. 2022

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

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

Metric	Truth	Smeared	Shifted	Removing tail	Particle features smeared	Particle $\eta^{\rm rel}$ smeared	Particle $p_{\rm T}^{\rm rel}$ smeared	Particle $p_{\rm T}^{\rm rel}$ 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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge
- EFPs and PNet activations performance similar

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge
- EFPs and PNet activations performance similar
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge
- EFPs and PNet activations performance similar
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing
- Classifiers, low-level (LLF) and high-level features (HLF), identify particle feature distortions but miss distribution-level discrepancies

Most sensitive metric per distribution in bold

RESULTS

RK et al. 2022

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge
- EFPs and PNet activations performance similar
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing
- Classifiers, low-level (LLF) and high-level features (HLF), identify particle feature distortions but miss distribution-level discrepancies
- FGD is the most sensitive to all distortions

Most sensitive metric per distribution in bold

RESULTS

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

- W_1^M - looking at 1D mass distribution only - is somewhat sensitive to all
- Wasserstein is sensitive to most, but slow to converge
- EFPs and PNet activations performance similar
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing
- Classifiers, low-level (LLF) and high-level features (HLF), identify particle feature distortions but miss distribution-level discrepancies
- FGD is the most sensitive to all distortions
- MMD reasonably sensitive to most

RK et al. 2022

EVALUATION TAKEAWAYS

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF test for all alternative hypotheses

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF test for all alternative hypotheses
  - His suggestion: use multiple, covering the relevant alternatives

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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 Physics Distance (FPD)**, using EFPs and shower-shape variables, for overall model evaluation and comparison

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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 Physics Distance (FPD)**, using EFPs and shower-shape variables, for overall model evaluation and comparison
  - Multivariate so directly applicable to conditional evaluation

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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 Physics Distance (FPD)**, using EFPs and shower-shape variables, for overall model evaluation and comparison
  - Multivariate so directly applicable to conditional evaluation
- But FGD can miss shape discrepancies, so use as well:

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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 Physics Distance (FPD)**, using EFPs and shower-shape variables, for overall model evaluation and comparison
  - Multivariate so directly applicable to conditional evaluation
- But FGD can miss shape discrepancies, so use as well:
  - Kernel Physics Distance (KPD) (MMD)

EVALUATION TAKEAWAYS

- Re-iterating Cousins 2016: no best GOF 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 Physics Distance (FPD)**, using EFPs and shower-shape variables, for overall model evaluation and comparison
  - Multivariate so directly applicable to conditional evaluation
- But FGD can miss shape discrepancies, so use as well:
  - Kernel Physics Distance (KPD) (MMD)
  - And continue with 1D distributions (W_1)

JET SIMULATION

RK et al., NeurIPS 2021

zenodo.org/record/5502543

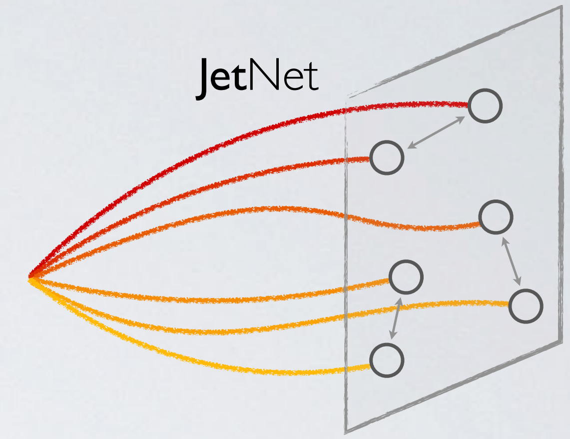
DATASET

zenodo.org/record/5502543

DATASET

RK et al., NeurIPS 2021

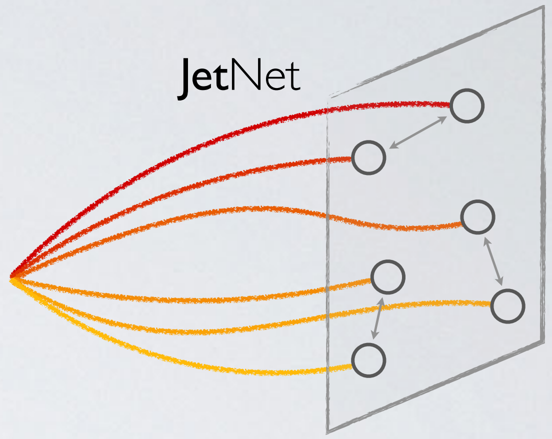
- Test-bench: Pythia-simulated high p_T jets ("JetNet")



RK et al., NeurIPS 2021

zenodo.org/record/5502543

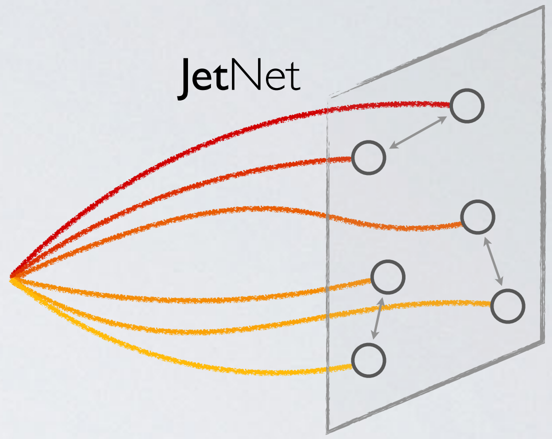
DATASET



- Test-bench: Pythia-simulated high p_T jets ("JetNet")
- 30 highest p_T particles, $(\eta^{rel}, \phi^{rel}, p_T^{rel})$ features

zenodo.org/record/5502543

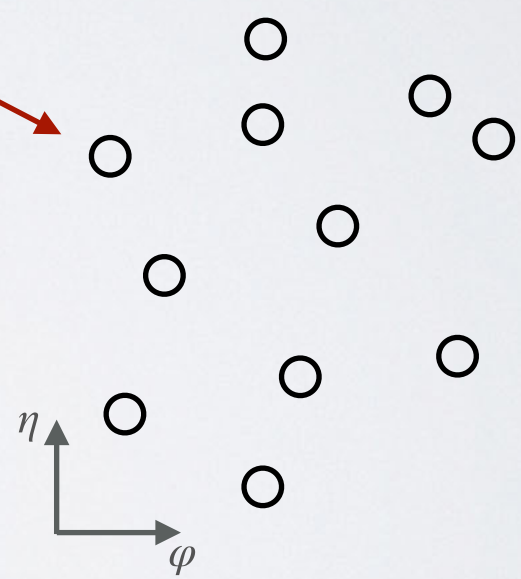
DATASET



- Test-bench: Pythia-simulated high p_T jets ("JetNet")

- 30 highest p_T particles, $(\eta^{rel}, \phi^{rel}, p_T^{rel})$ features

Particle Cloud

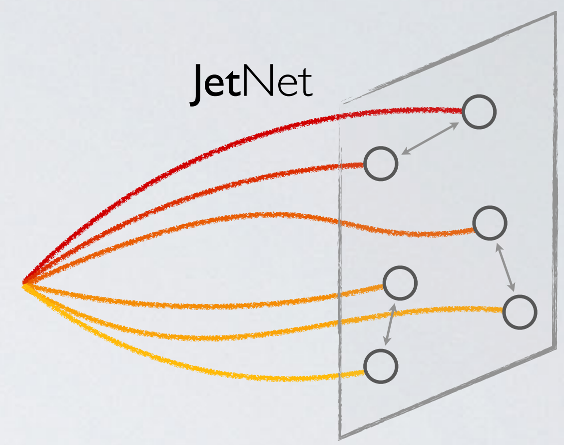


RK et al., NeurIPS 2021

zenodo.org/record/5502543

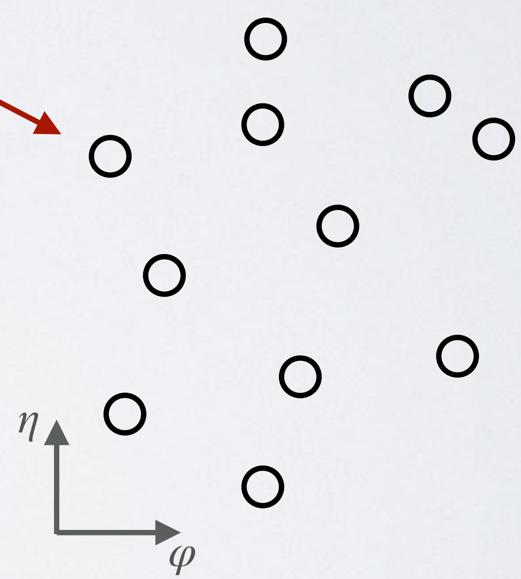
DATASET

- Test-bench: Pythia-simulated high p_T jets ("JetNet")



- 30 highest p_T particles, $(\eta^{rel}, \phi^{rel}, p_T^{rel})$ features

Particle Cloud



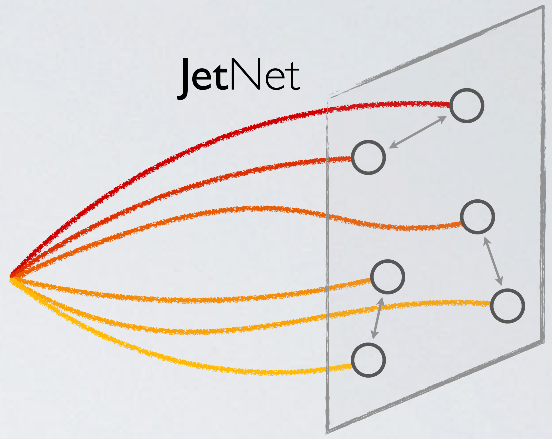
- **Gen particle → Reco jet**

zenodo.org/record/5502543

DATASET

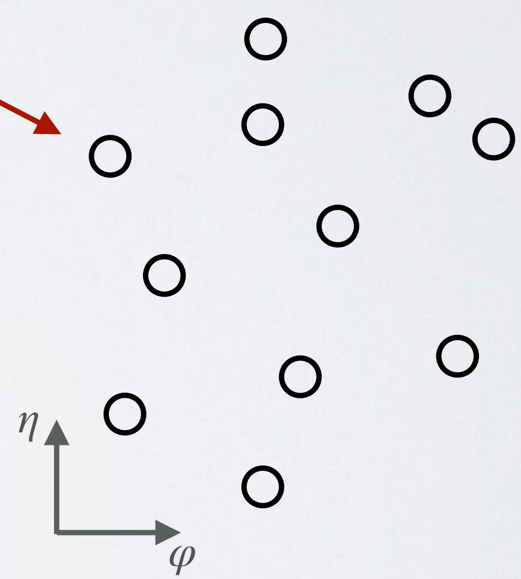
RK et al., NeurIPS 2021

- Test-bench: Pythia-simulated high p_T jets ("JetNet")

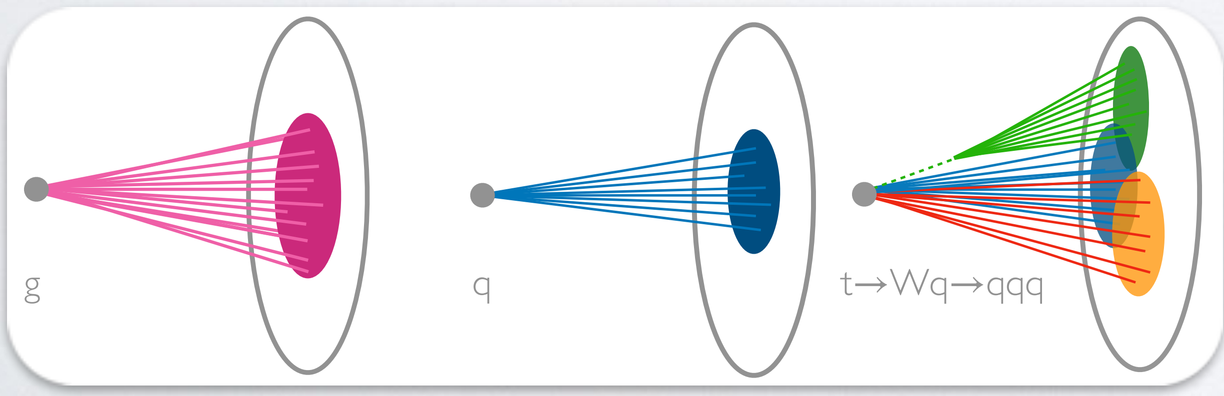


- 30 highest p_T particles, $(\eta^{rel}, \phi^{rel}, p_T^{rel})$ features

Particle Cloud



- **Gen particle → Reco jet**

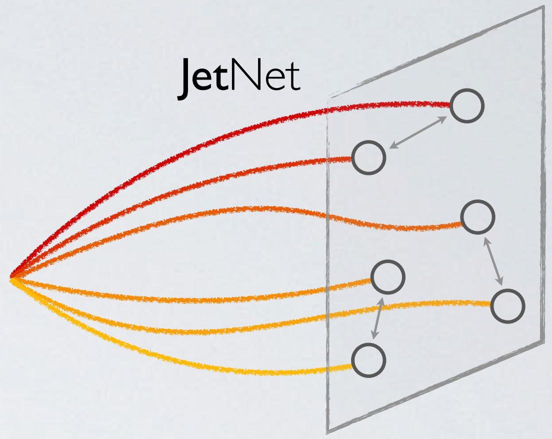


zenodo.org/record/5502543

DATASET

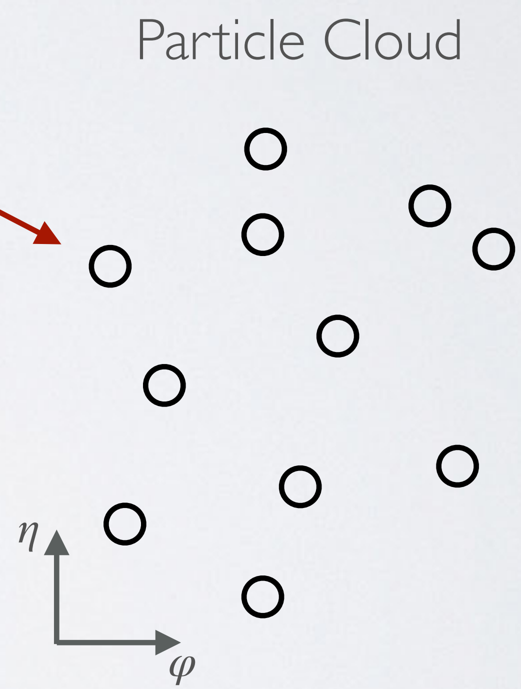
RK et al., NeurIPS 2021

- Test-bench: Pythia-simulated high p_T jets ("JetNet")

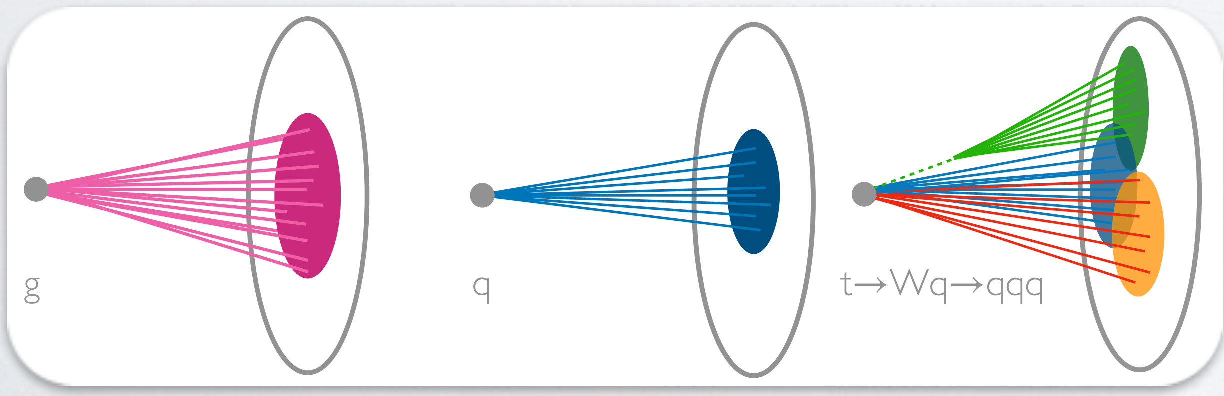


JetNet Library:
25k downloads, [1, 2, 3]

- 30 highest p_T particles, $(\eta^{rel}, \phi^{rel}, p_T^{rel})$ features



- **Gen particle → Reco jet**



APPROACH 1: MPGAN

RK et al., NeurIPS 2021

APPROACH 1: MPGAN

- Majority of work, while successful, is image-based

RK et al., NeurIPS 2021

APPROACH 1: MPGAN

- Majority of work, while successful, is image-based

- Difficult to scale to HL-LHC and apply to e.g CMS high-granularity calorimeter

APPROACH 1: MPGAN

- Majority of work, while successful, is image-based

- Difficult to scale to HL-LHC and apply to e.g CMS high-granularity calorimeter

- We develop a particle cloud, graph-based approach

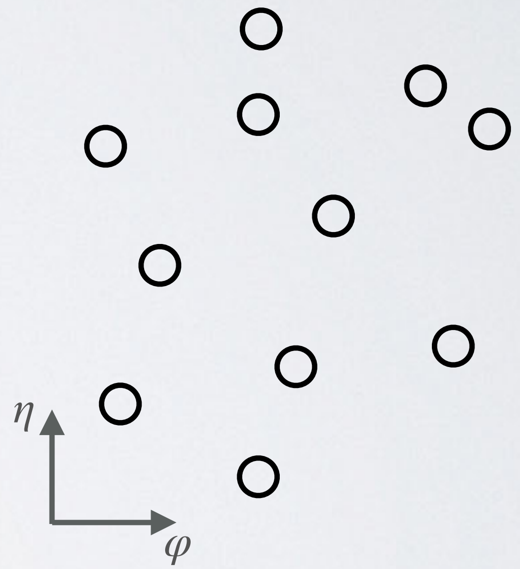
RK et al., NeurIPS 2021

APPROACH 1: MPGAN

- Majority of work, while successful, is image-based
- Difficult to scale to HL-LHC and apply to e.g CMS high-granularity calorimeter
- We develop a particle cloud, graph-based approach
- Key ideas:

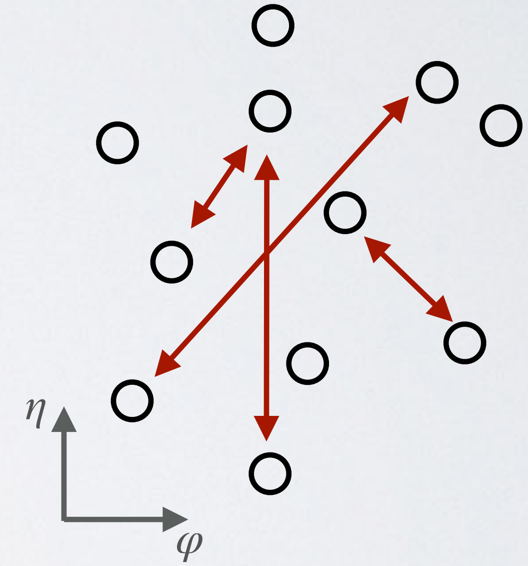
APPROACH 1: MPGAN

- Majority of work, while successful, is image-based
- Difficult to scale to HL-LHC and apply to e.g CMS high-granularity calorimeter
- We develop a particle cloud, graph-based approach
- Key ideas:
  - Natural, sparse, and flexible representation for data



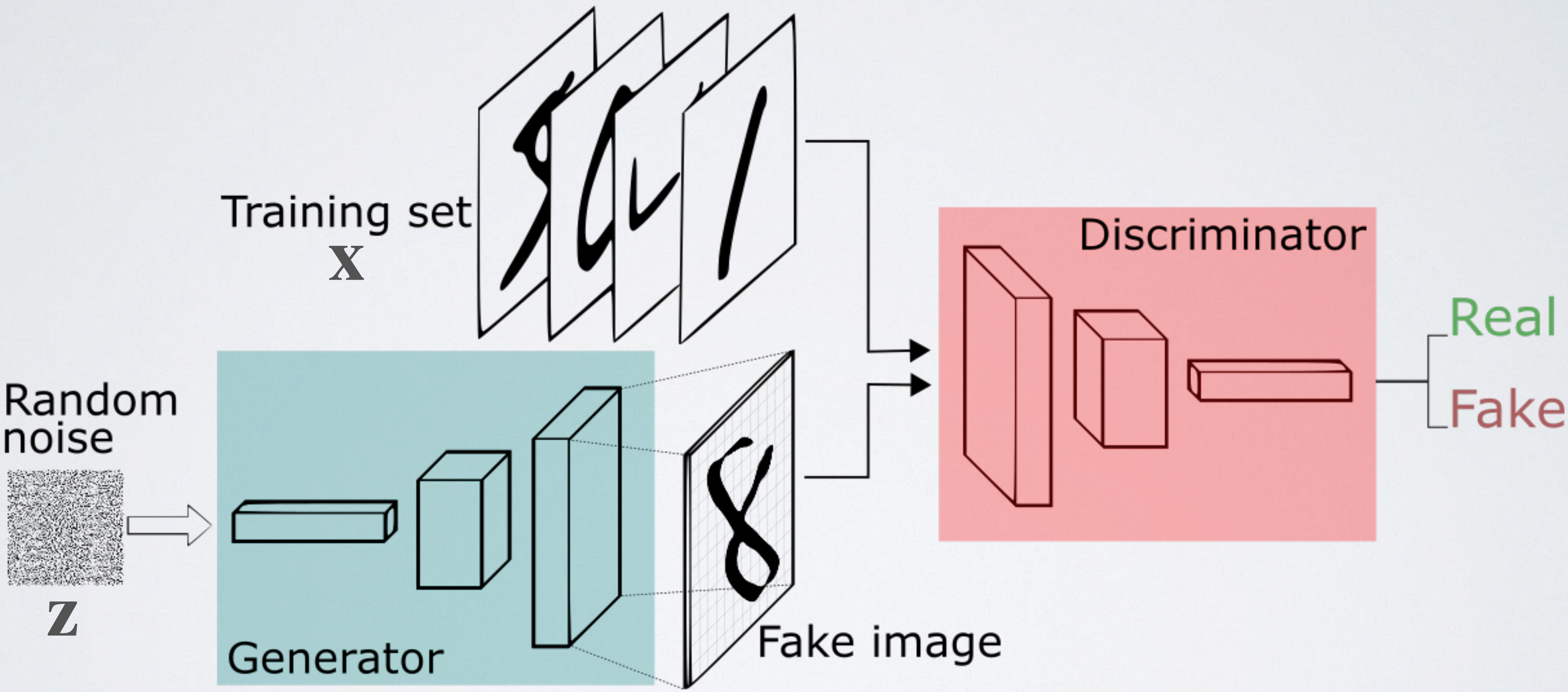
APPROACH 1: MPGAN

- Majority of work, while successful, is image-based
- Difficult to scale to HL-LHC and apply to e.g CMS high-granularity calorimeter
- We develop a particle cloud, graph-based approach
- Key ideas:
    - Natural, sparse, and flexible representation for data
    - Learn global features *and* inter-particle correlations (i.e. jet, shower structure)



GANS:
GENERATIVE ADVERSARIAL NETWORKS

GANS:
GENERATIVE ADVERSARIAL NETWORKS



MPGAN

RK et al., NeurIPS 2021

MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

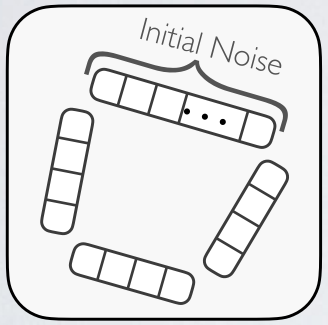
MP Generator

MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

MP Generator

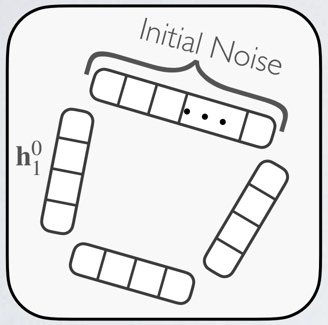


MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

MP Generator

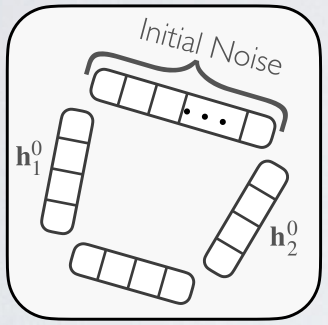


MPGAN

RK et al., NeurIPS 2021

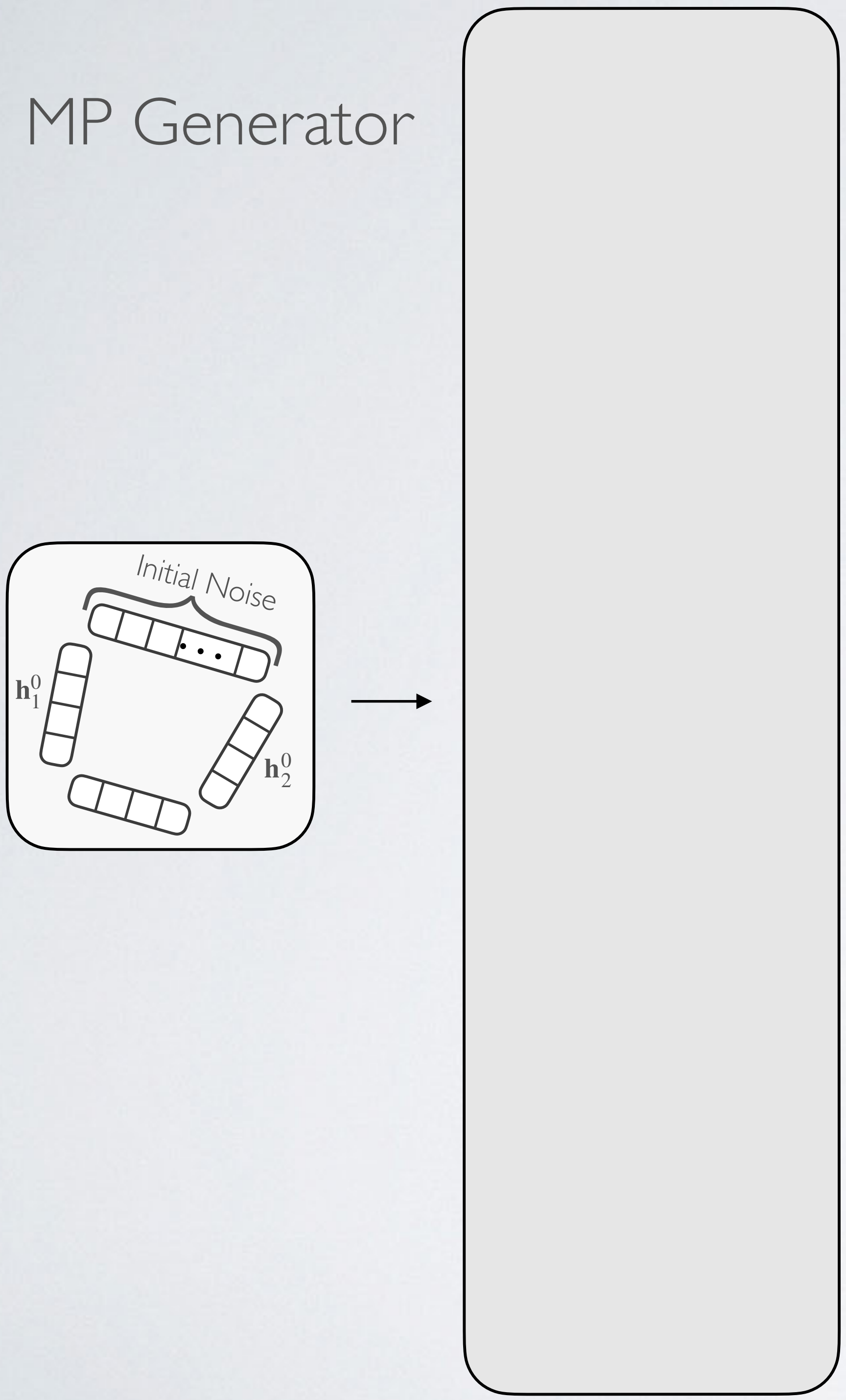
- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

MP Generator



MPGAN

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

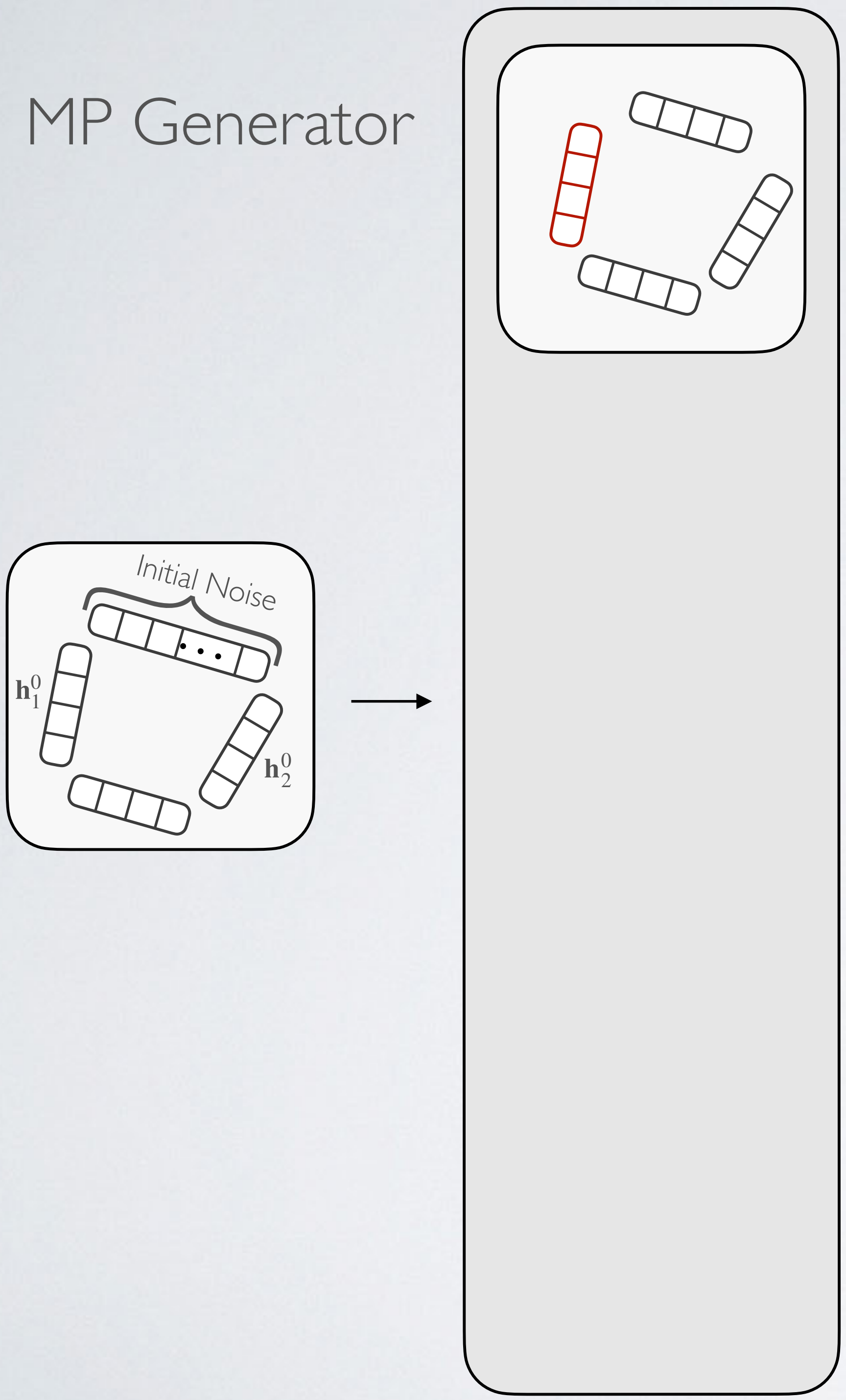


MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

MP Generator



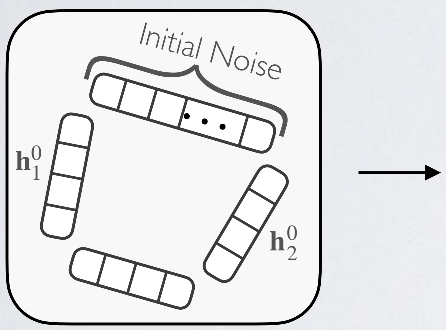
MPGAN

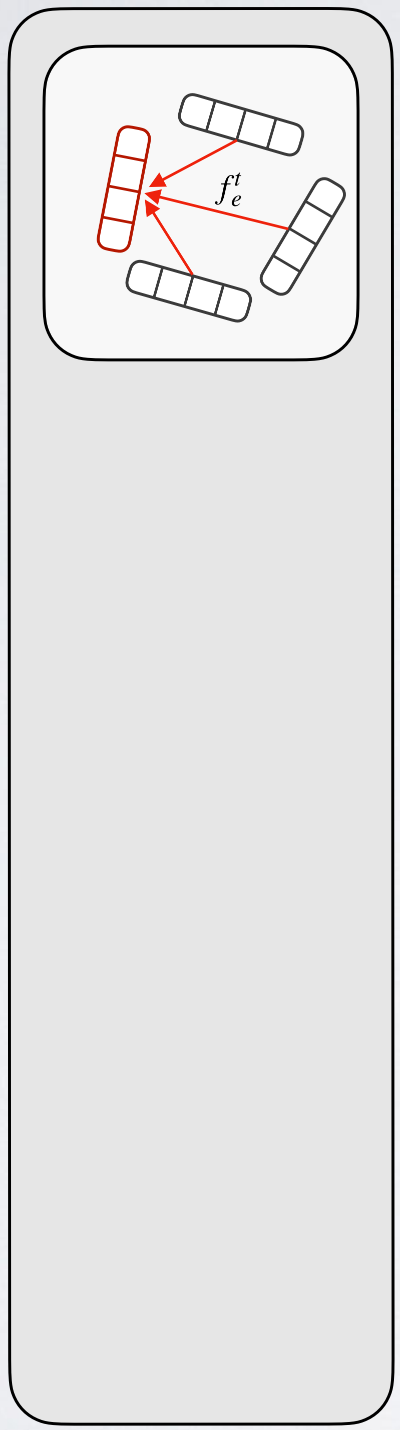
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

MP Generator





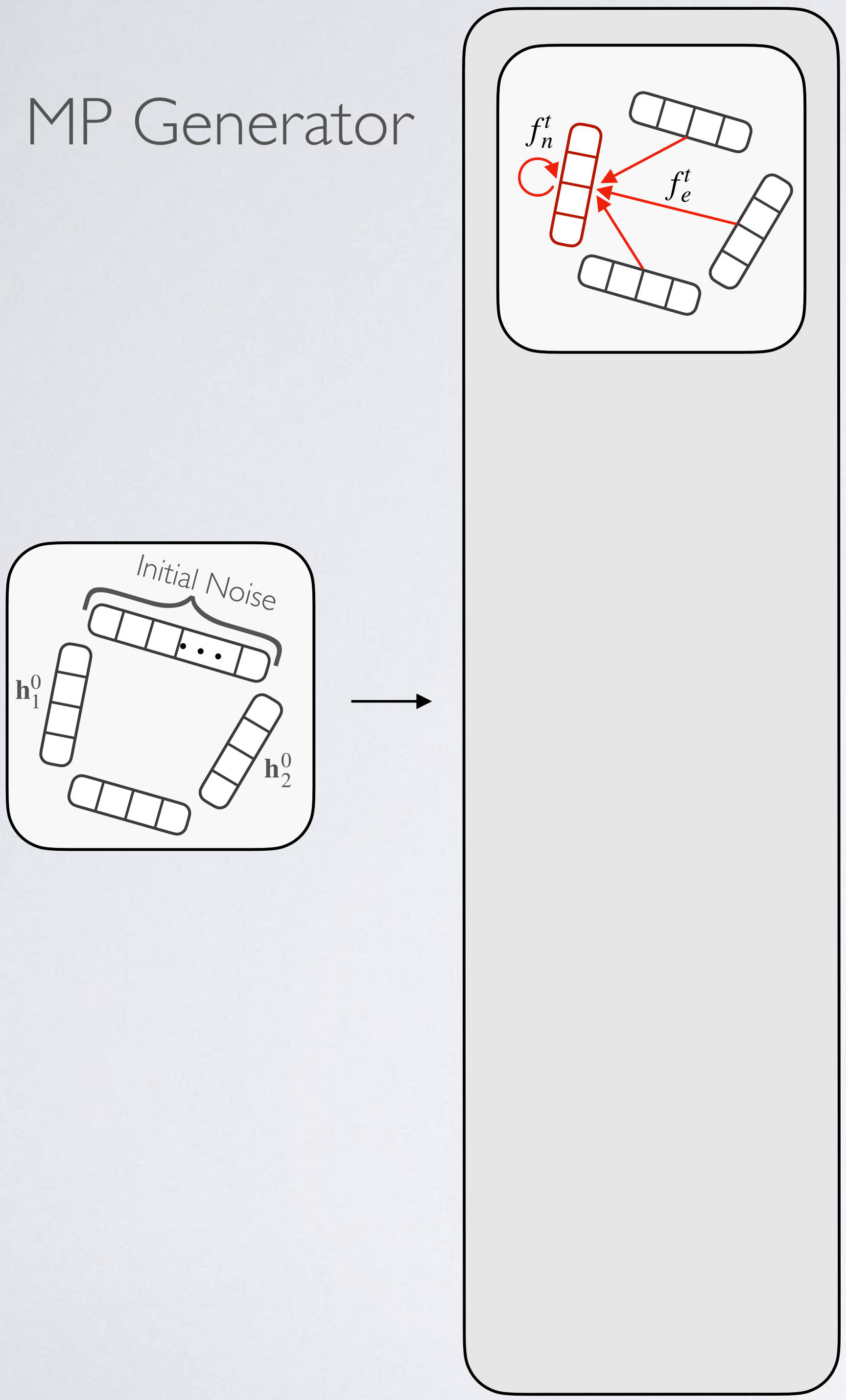
MPGAN

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MPGAN

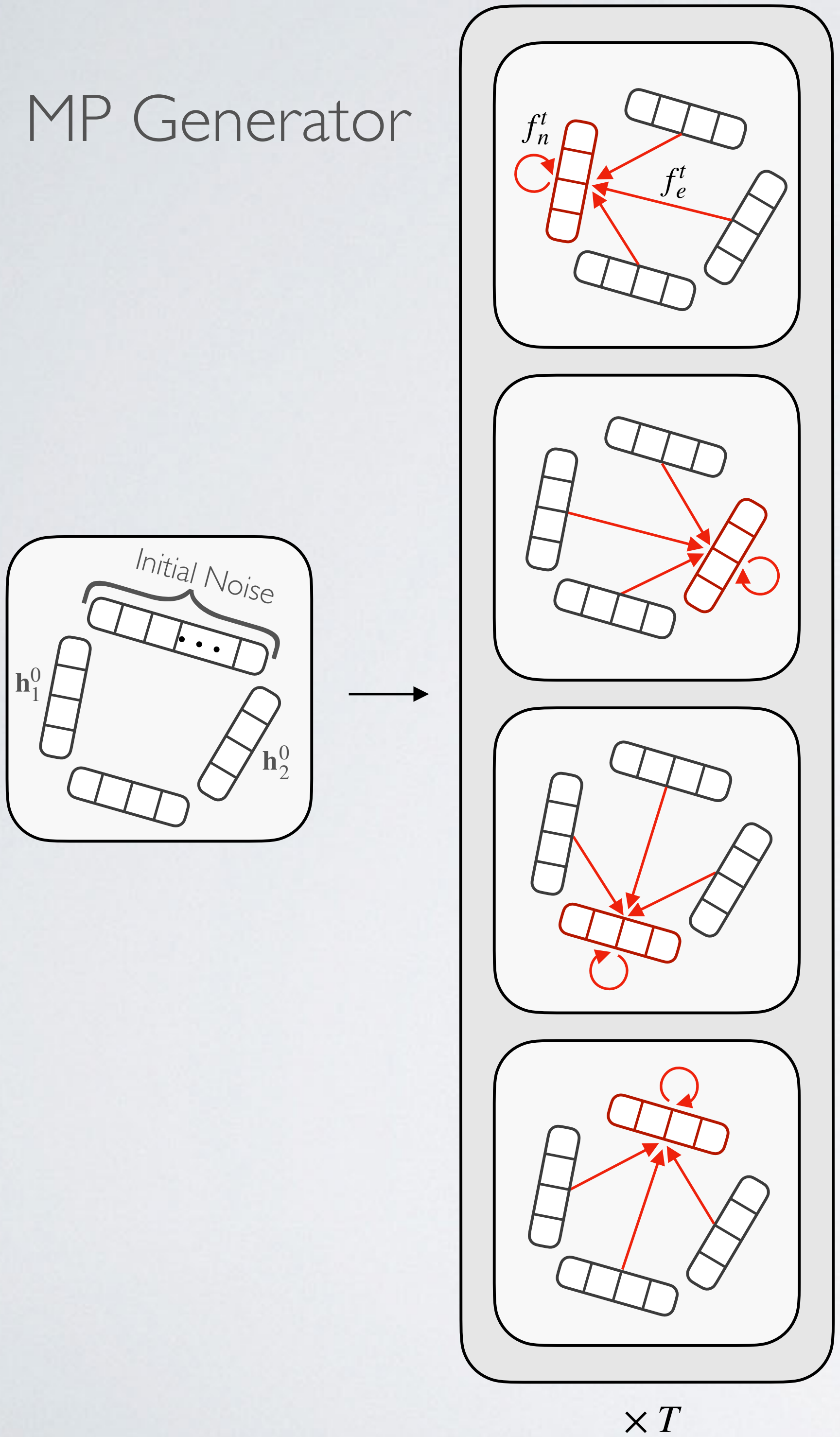
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MPGAN

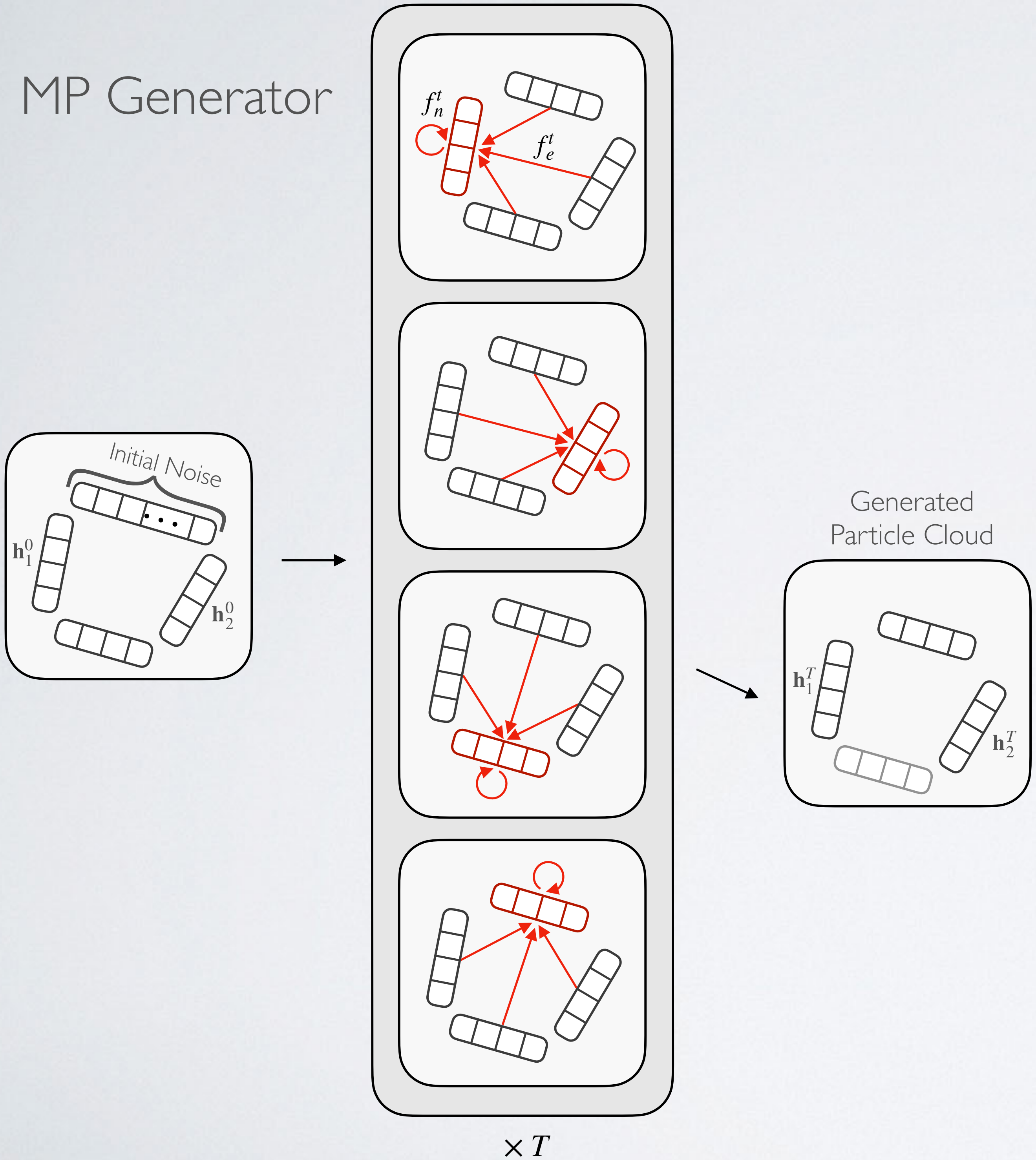
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MPGAN

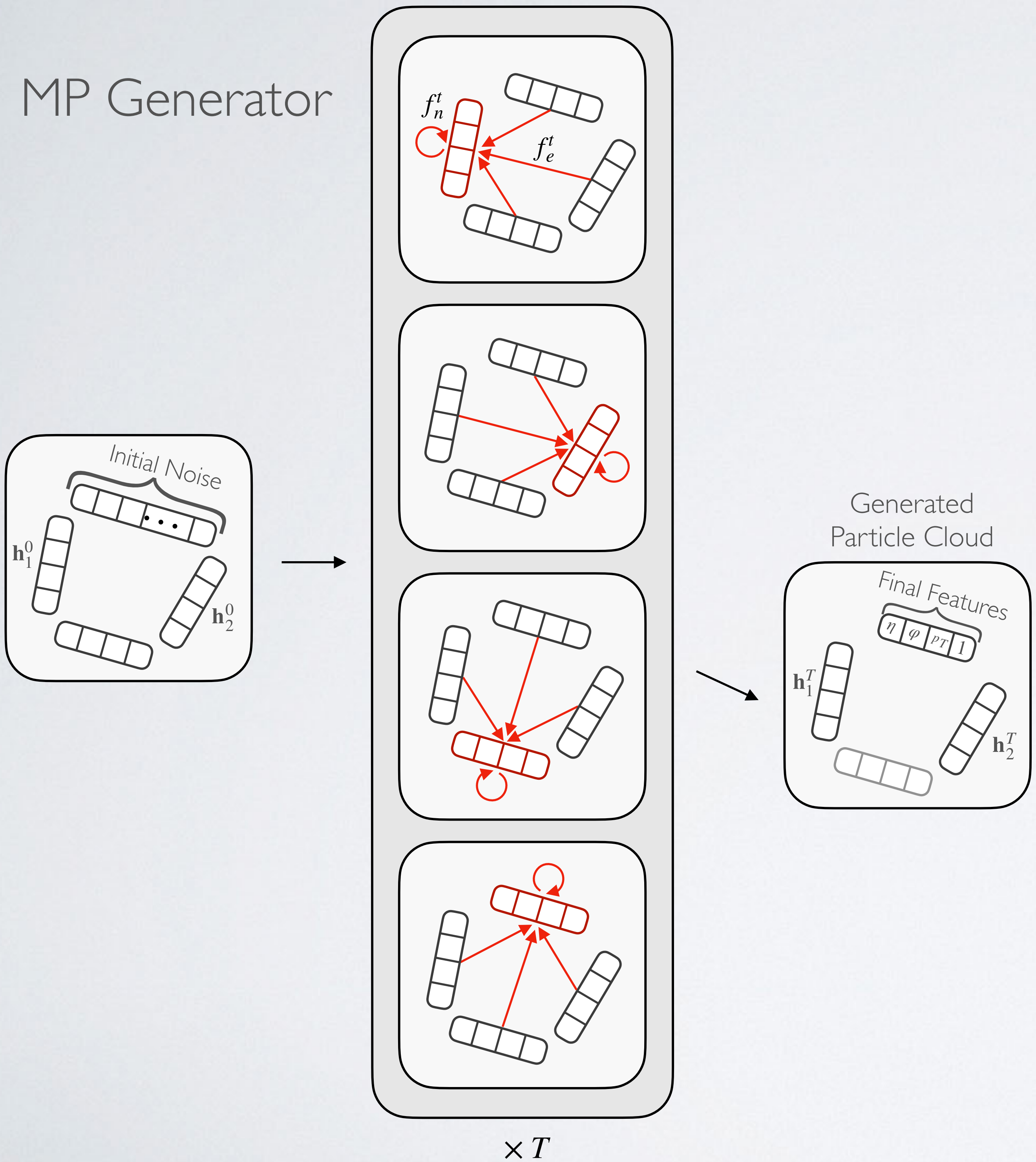
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MPGAN

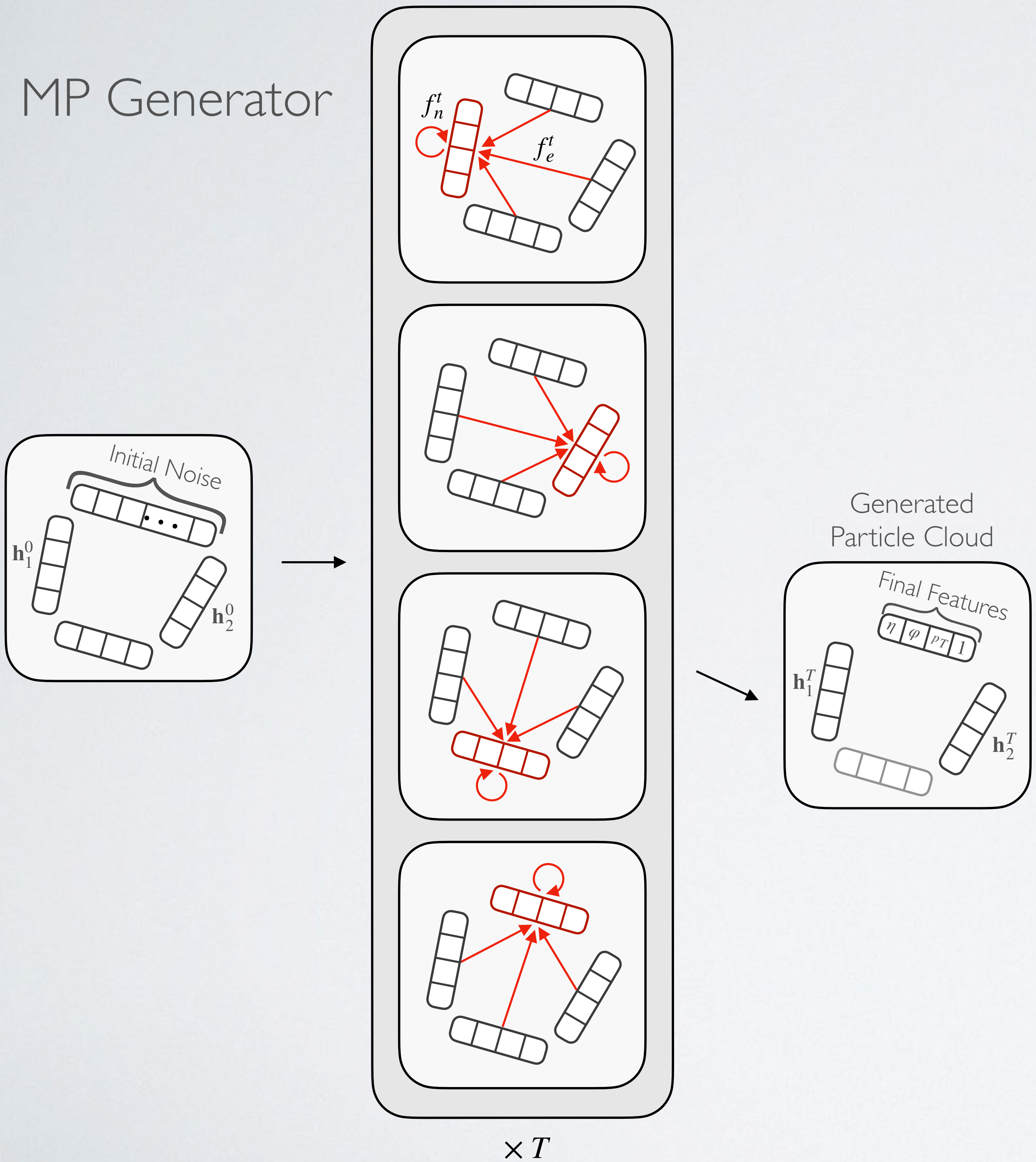
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MP Discriminator

MPGAN

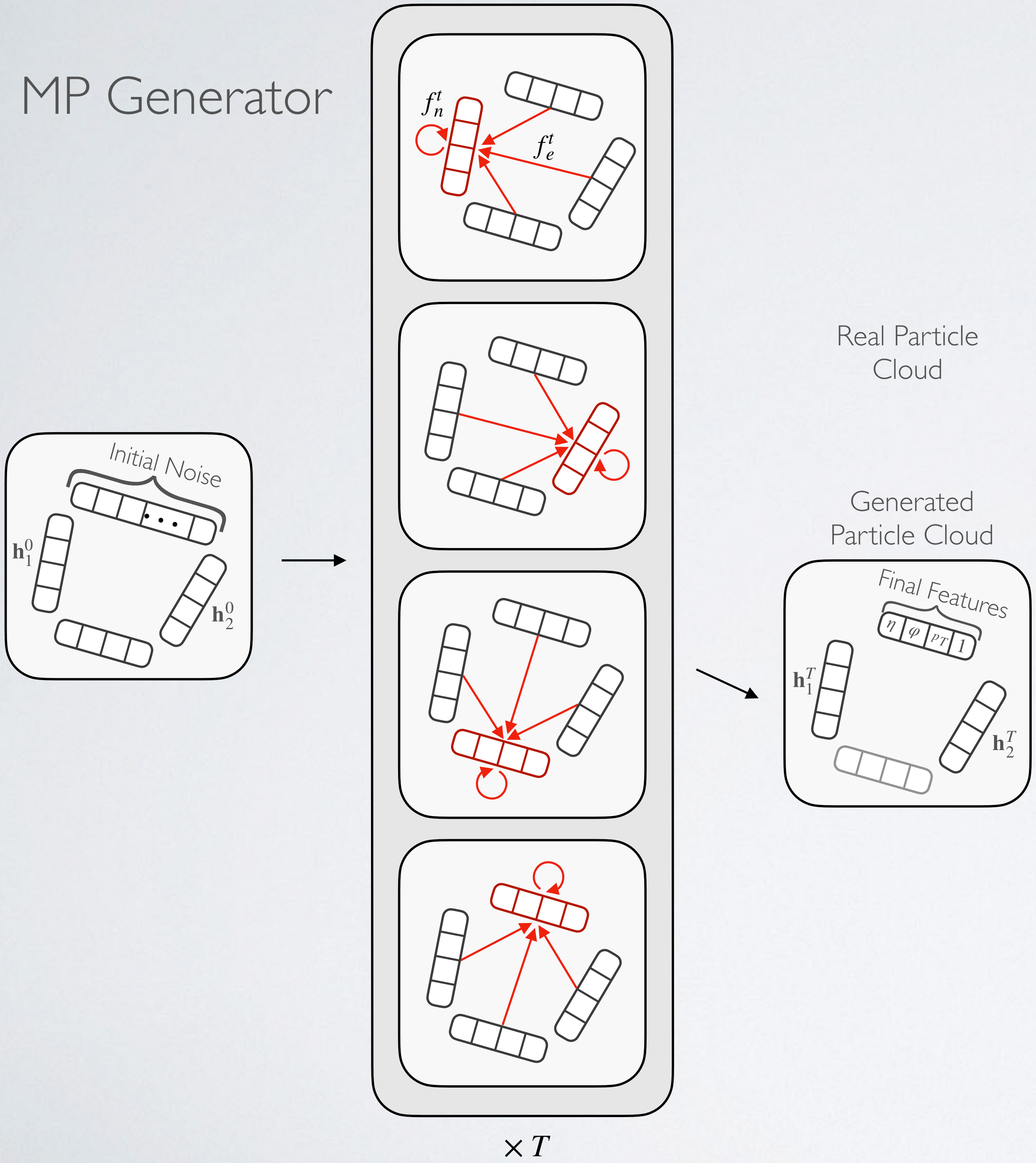
RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

MP Generator



MP Discriminator

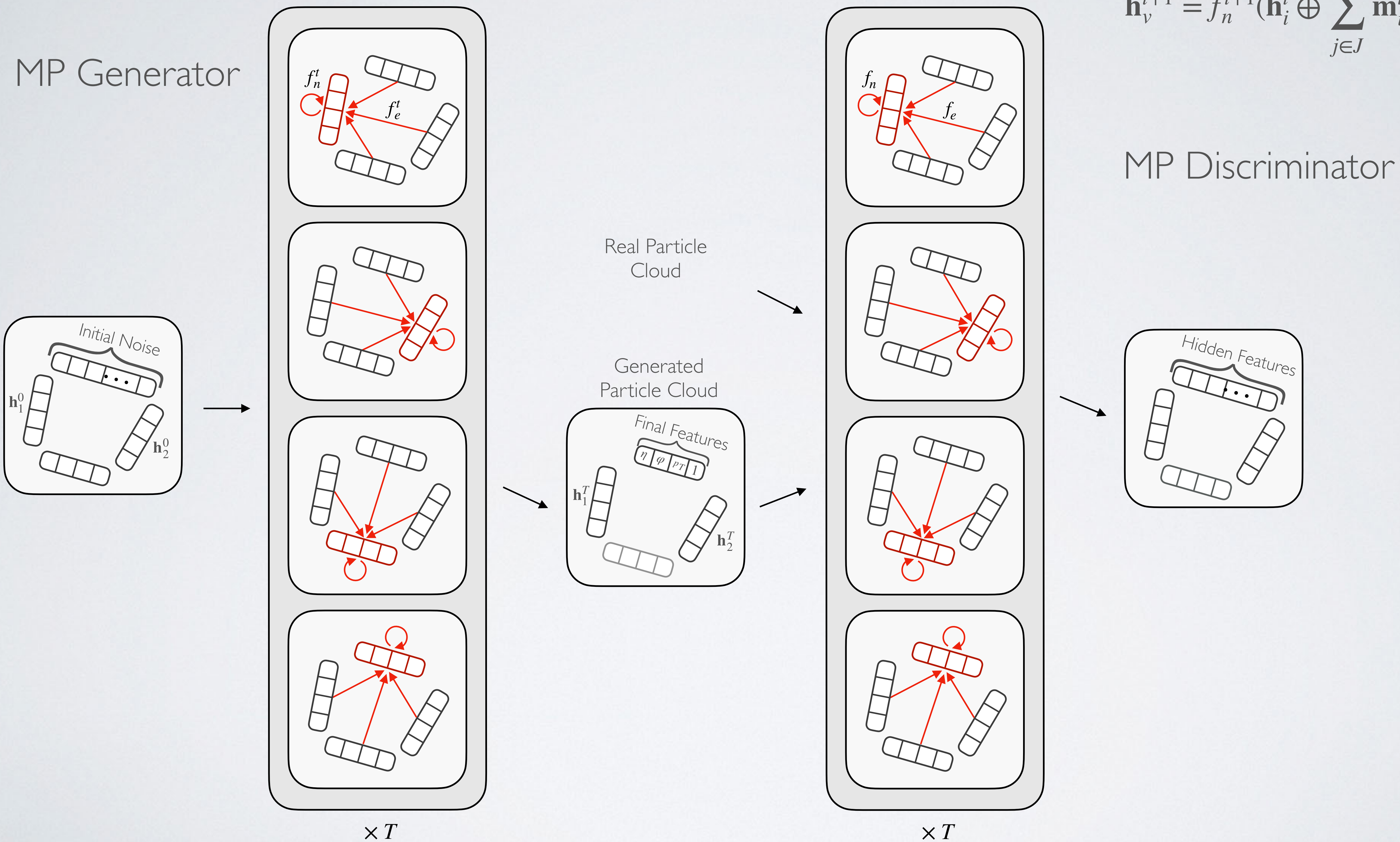
MPGAN

RK et al., NeurIPS 2021

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

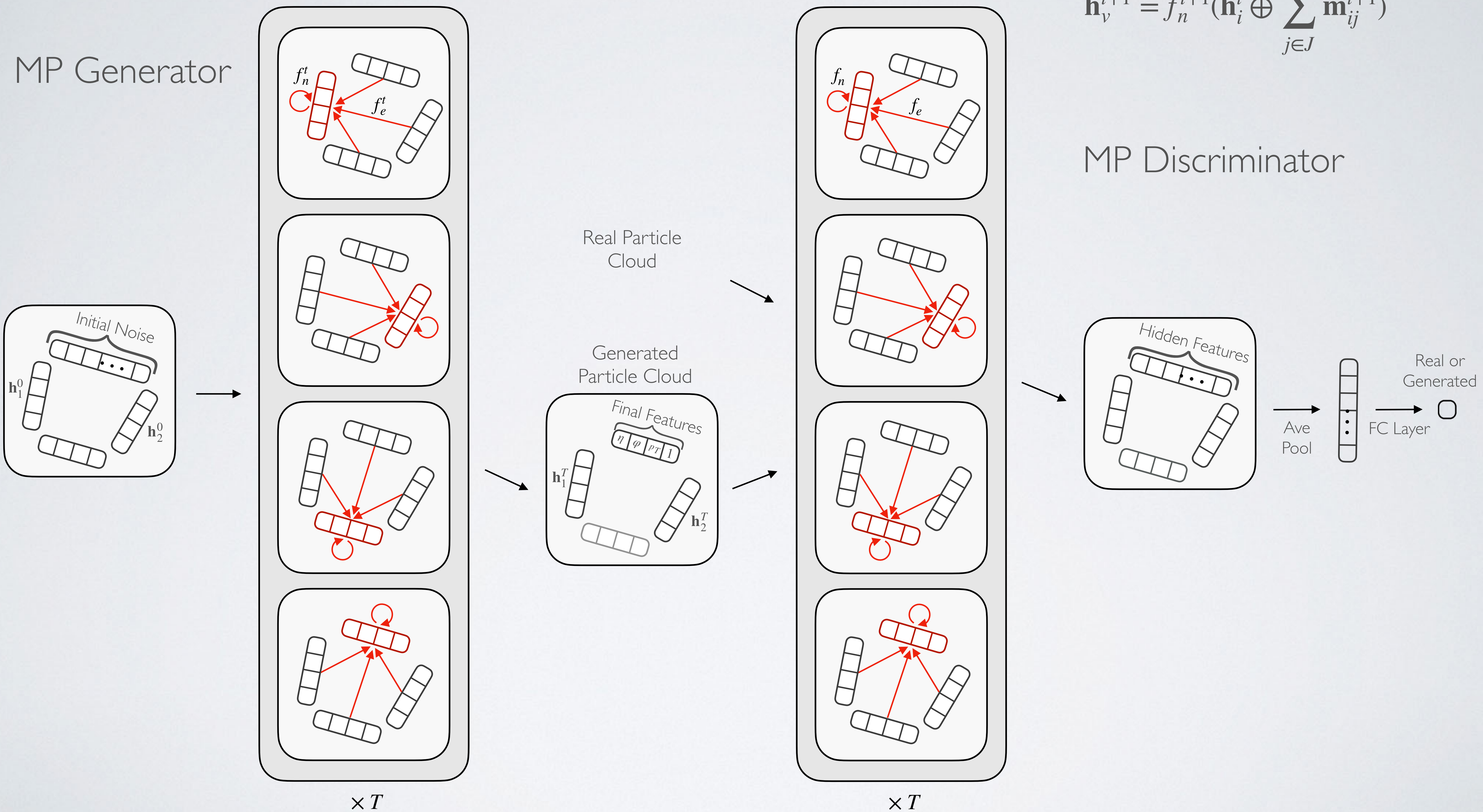


MPGAN

- We develop a GAN with a fully-connected message-passing (MP) generator and discriminator

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathbf{h}_i^t \oplus \mathbf{h}_j^t)$$

$$\mathbf{h}_v^{t+1} = f_n^{t+1}(\mathbf{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1})$$

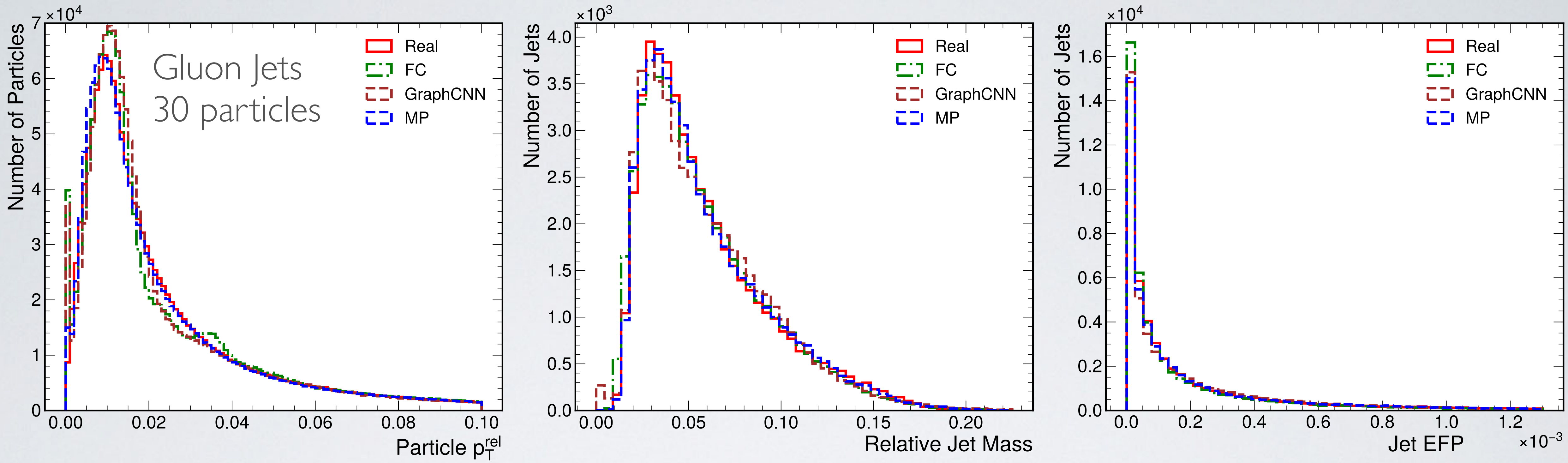


MPGAN: RESULTS

RK et al., NeurIPS 2021

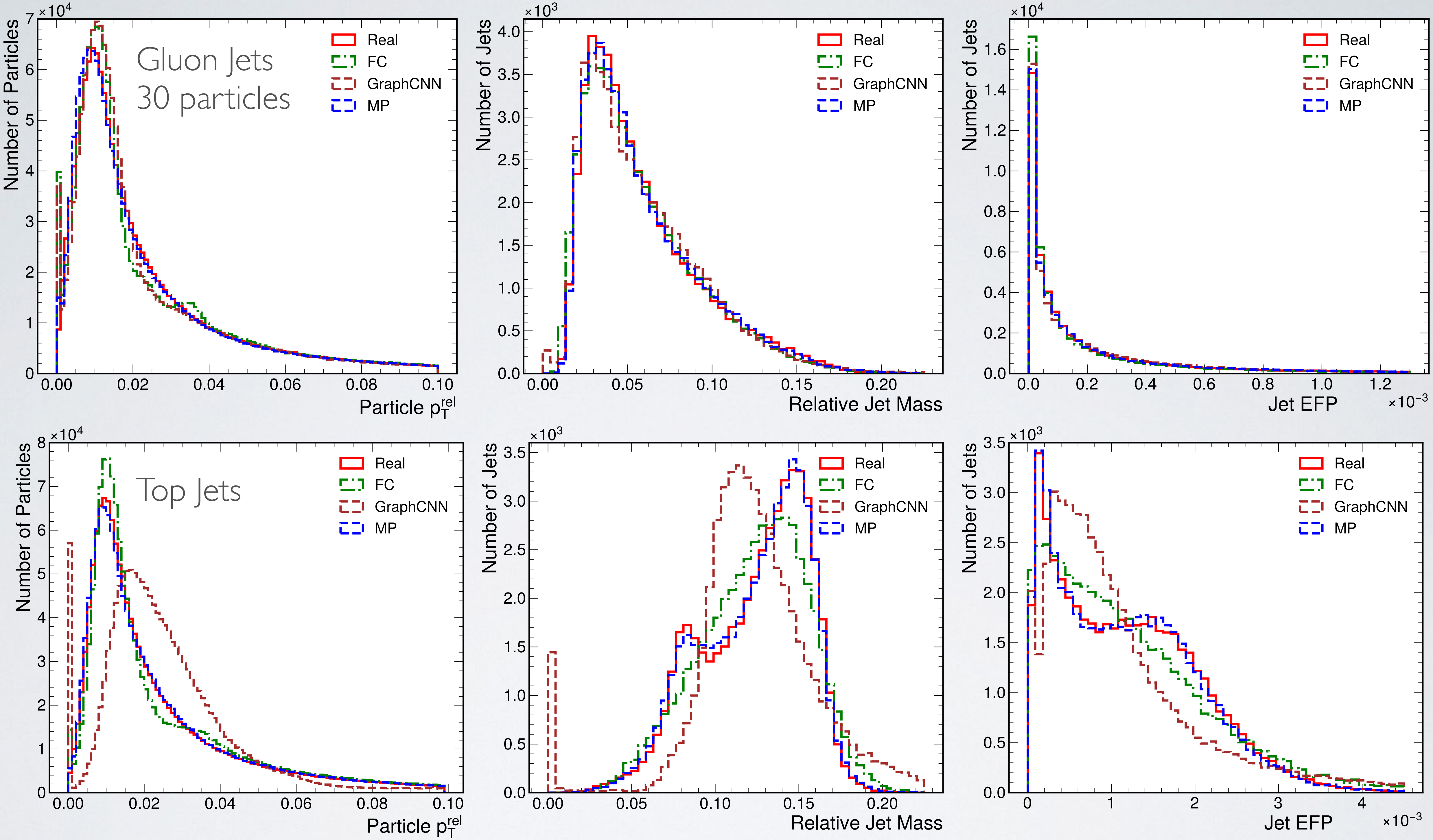
MPGAN: RESULTS

RK et al., NeurIPS 2021



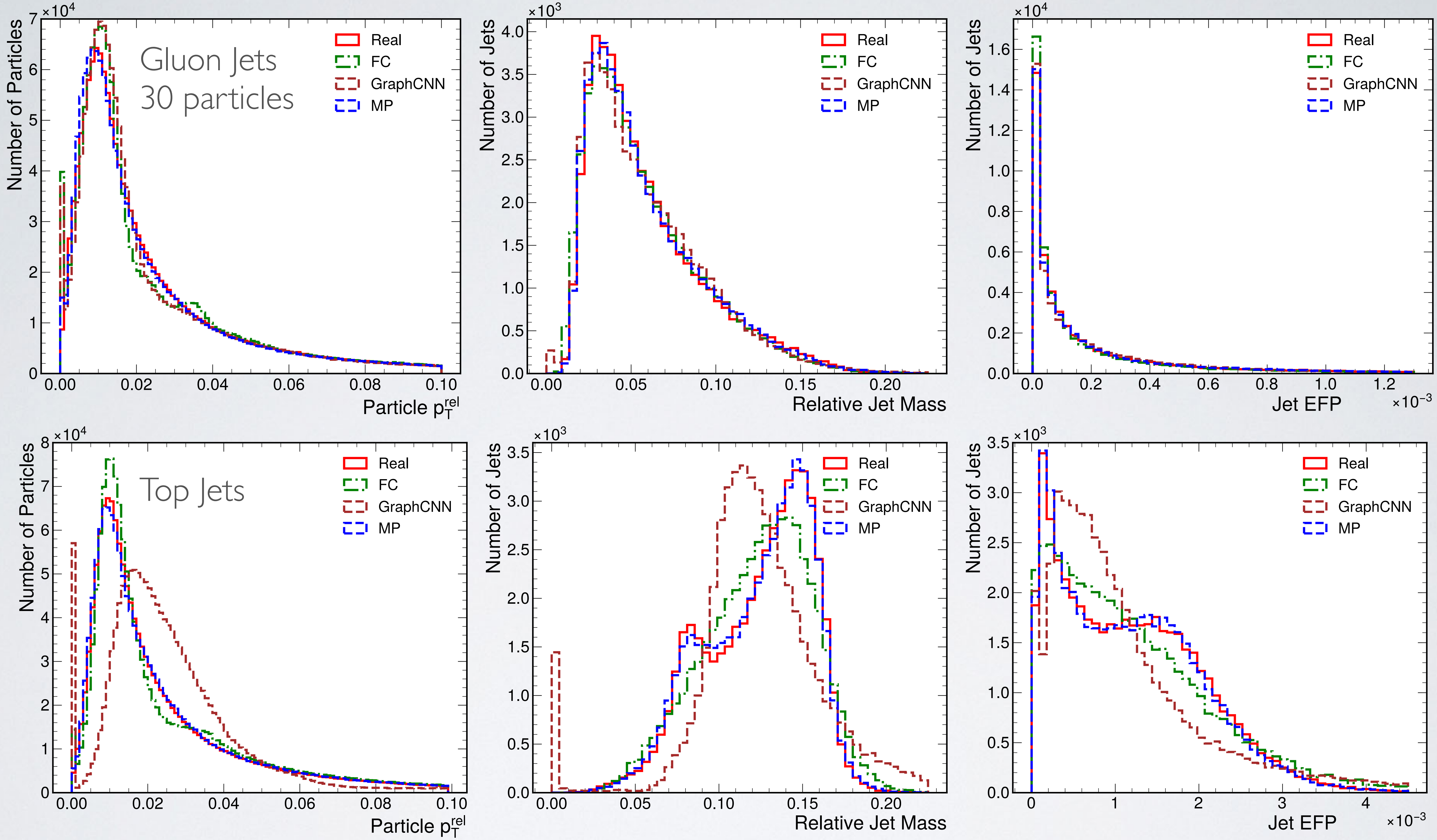
MPGAN: RESULTS

RK et al., NeurIPS 2021



MPGAN: RESULTS

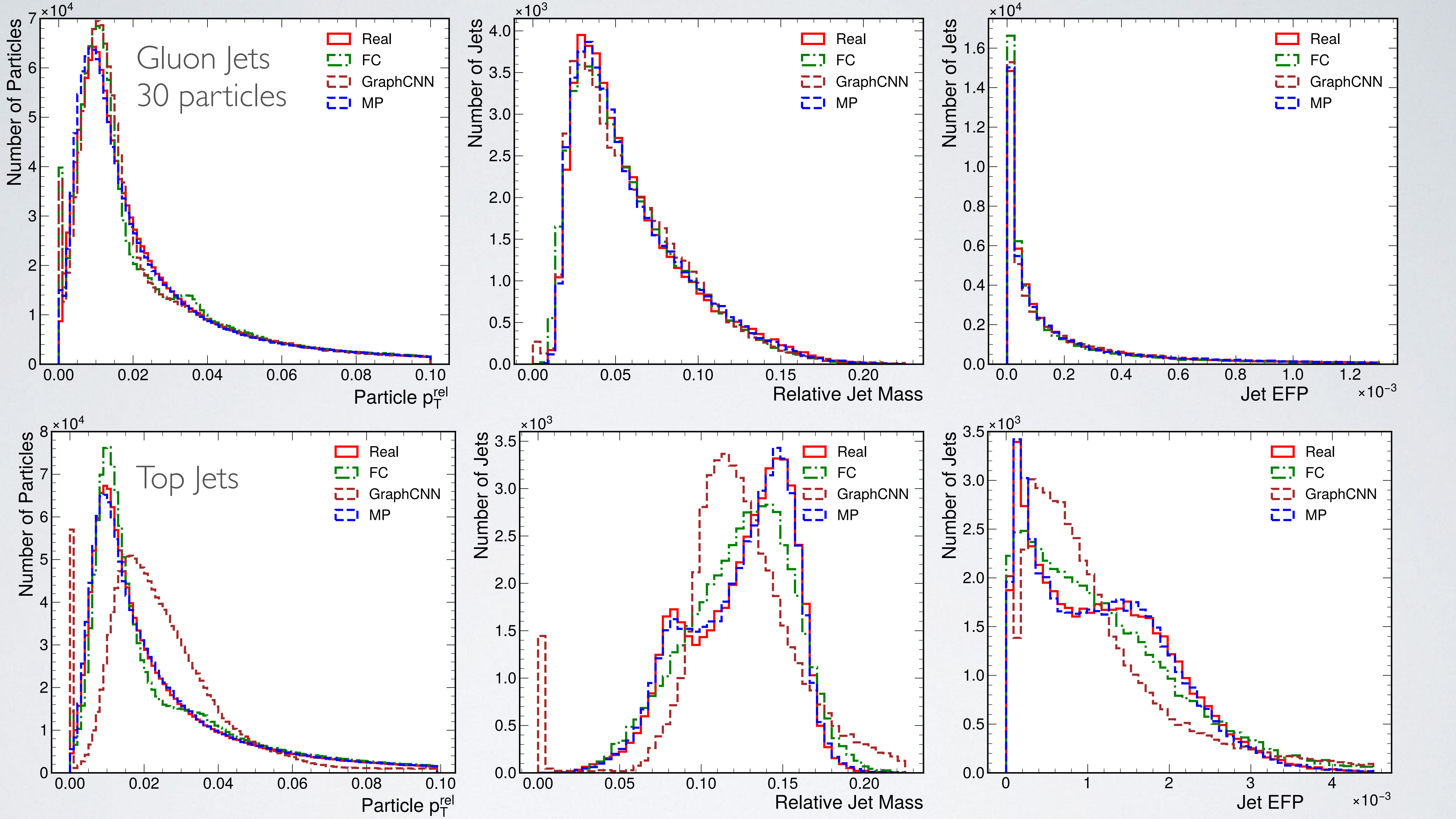
RK et al., NeurIPS 2021



- MPGAN (blue) learns real (red) distributions well

MPGAN: RESULTS

RK et al., NeurIPS 2021



- MPGAN (blue) learns real (red) distributions well
- Outperforms all existing point cloud GANs (metrics in backup)

APPROACH 2: GAPT

RK et al., 2022

APPROACH 2: GAPT

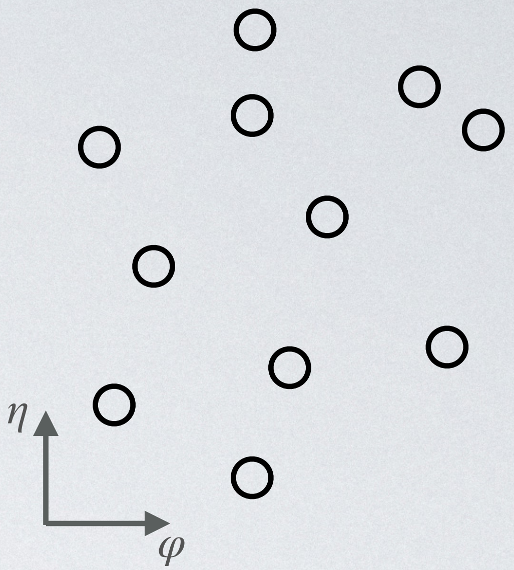
RK et al., 2022

- Retain key ideas of MPGAN

APPROACH 2: GAPT

RK et al., 2022

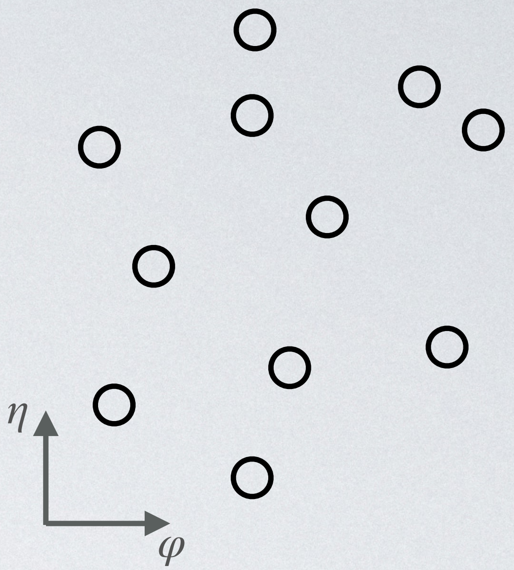
- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions



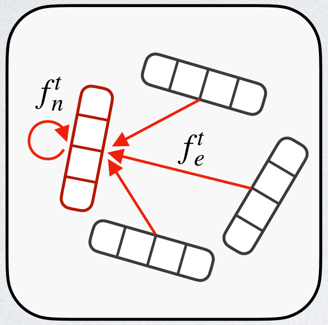
APPROACH 2: GAPT

RK et al., 2022

- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

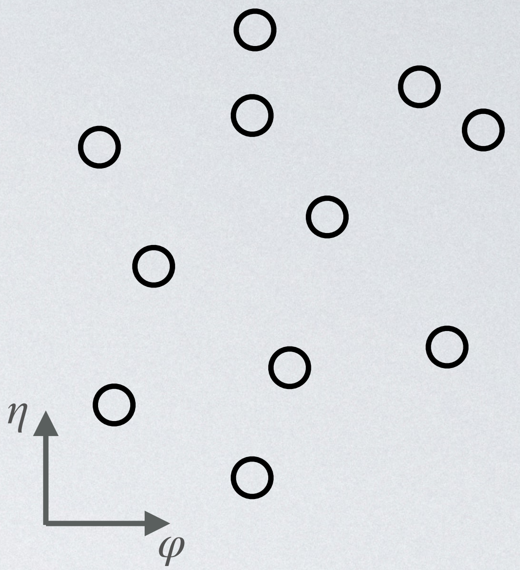


Message passing



APPROACH 2: GAPT

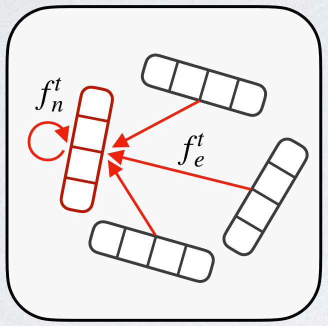
RK et al., 2022



- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

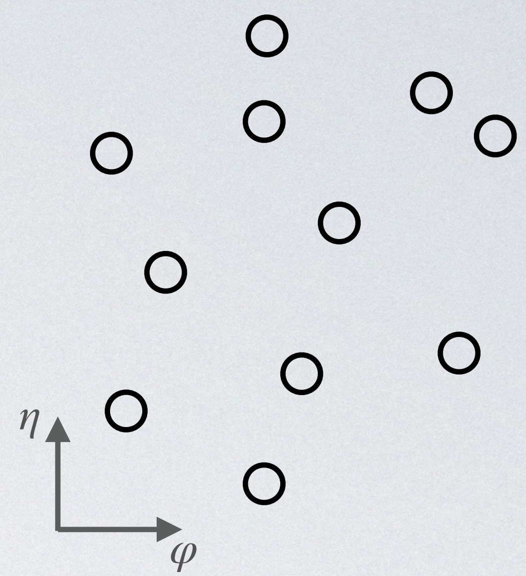
Self-attention (set transformers, Lee et al. ICML 2019)

Message passing



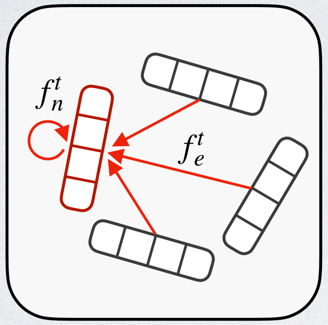
APPROACH 2: GAPT

RK et al., 2022

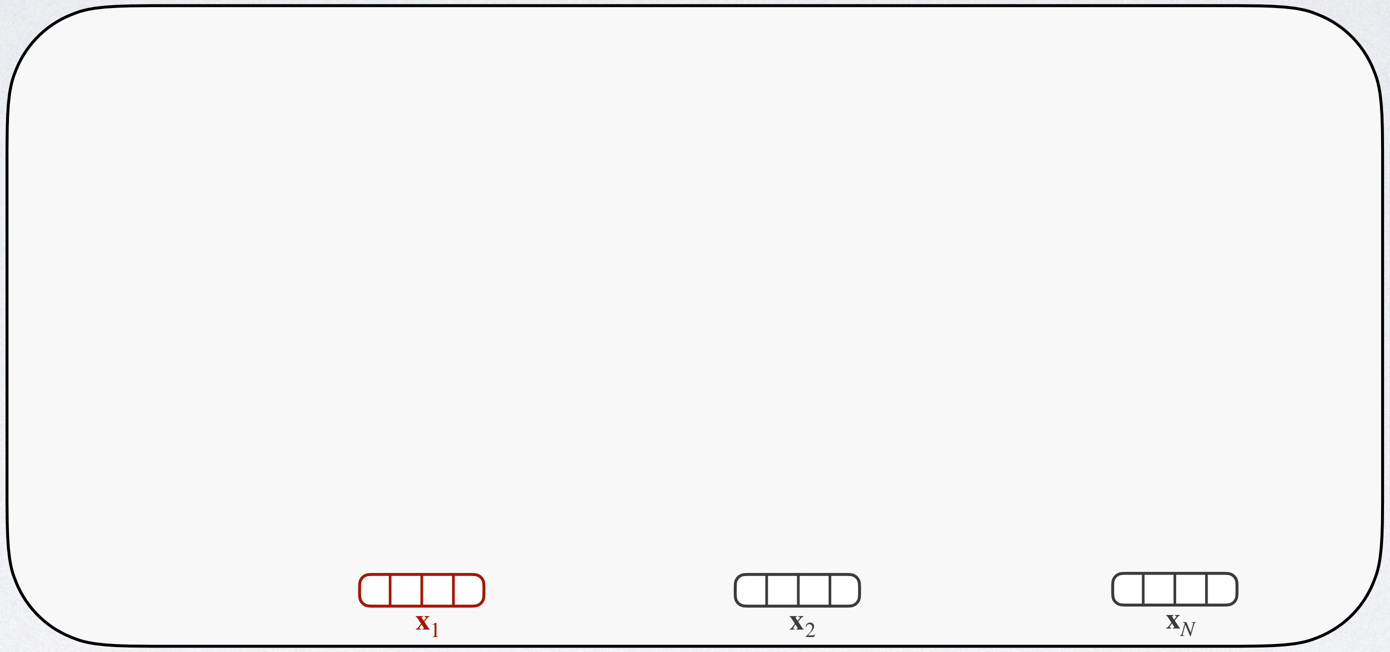


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

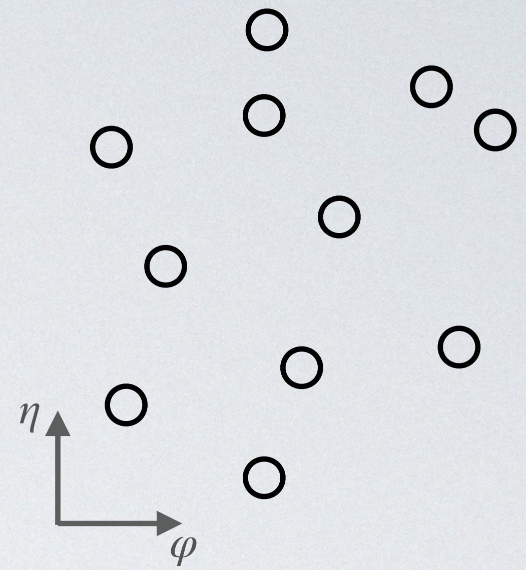
Message passing



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

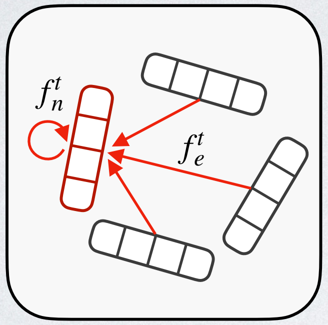


APPROACH 2: GAPT

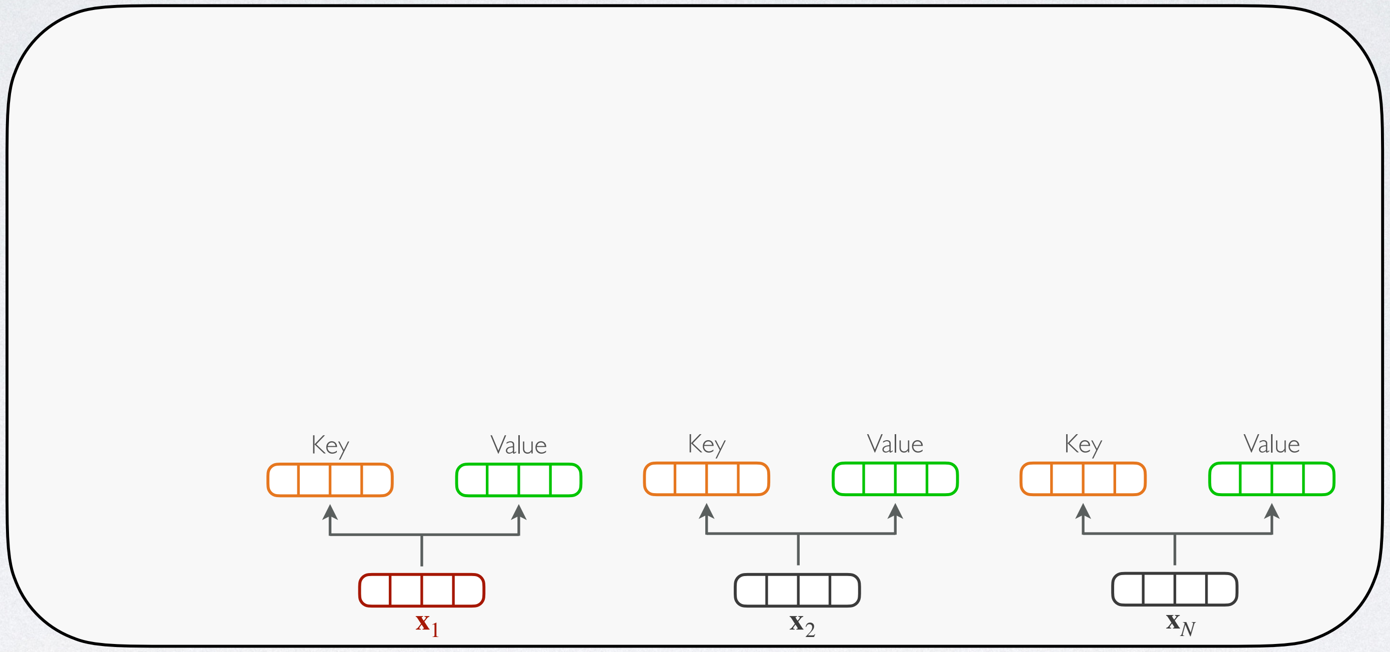


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing

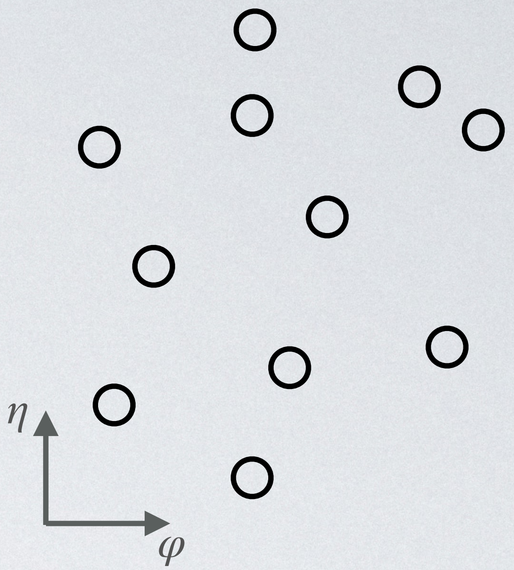


Self-attention (set transformers, Lee et al. ICML 2019)



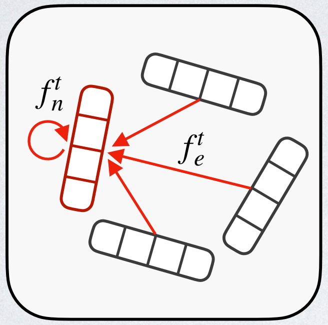
APPROACH 2: GAPT

RK et al., 2022

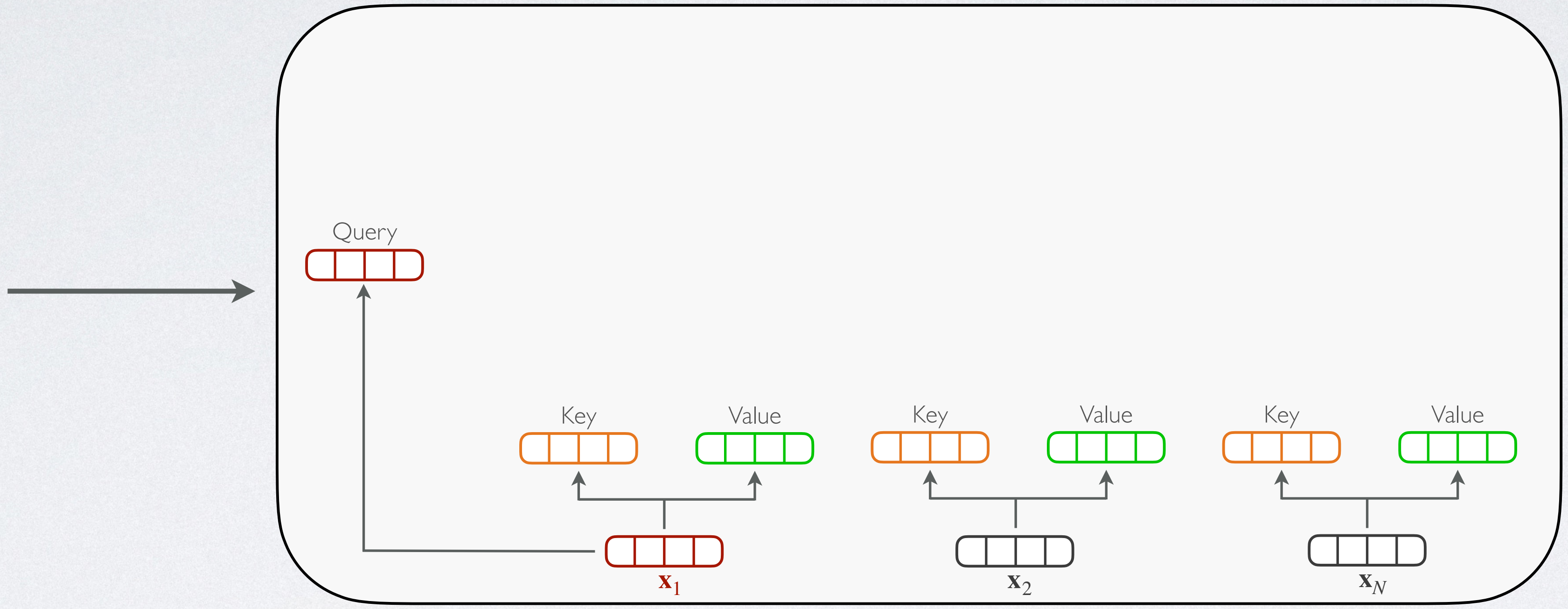


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing

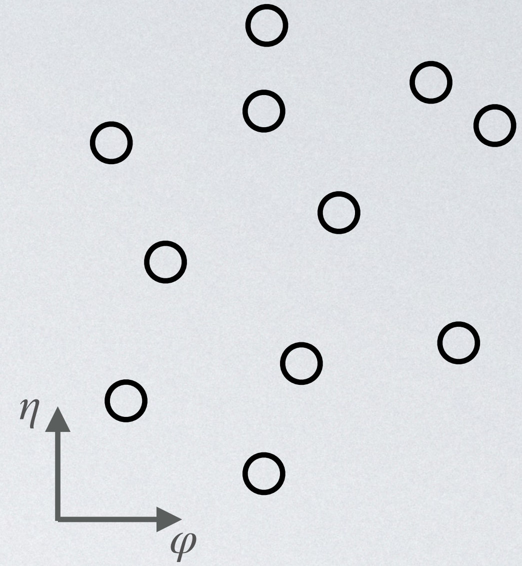


Self-attention (set transformers, Lee et al. ICML 2019)



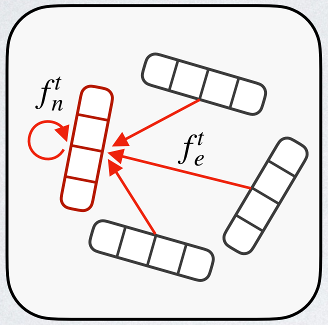
APPROACH 2: GAPT

RK et al., 2022

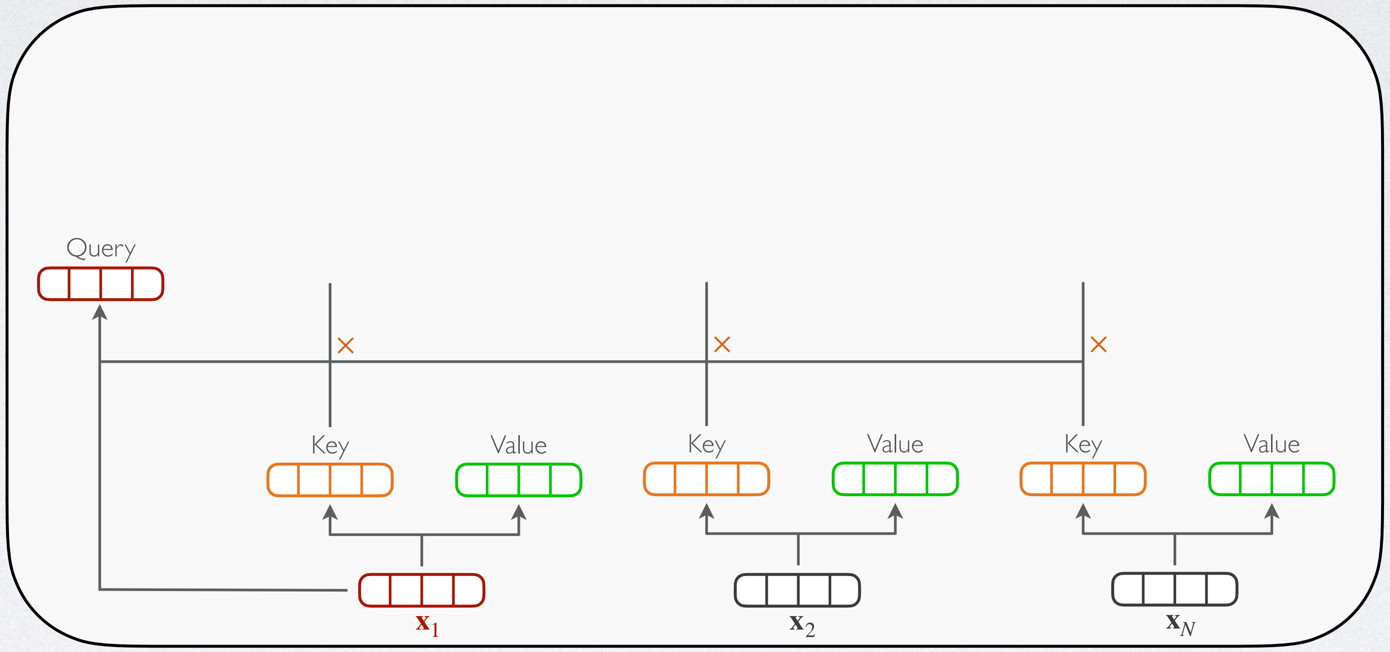


- Retain key ideas of MPGAN
    - Particle cloud data, fully connected particle interactions

Message passing

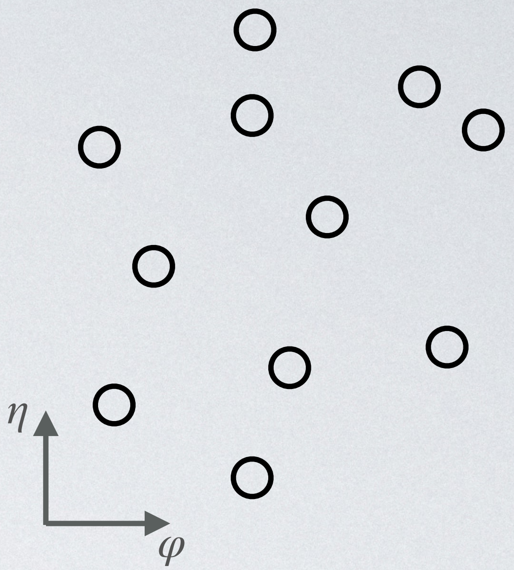


Self-attention (set transformers, Lee et al. ICML 2019)



APPROACH 2: GAPT

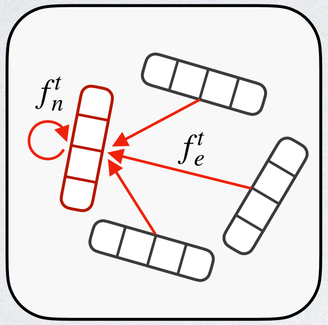
RK et al., 2022

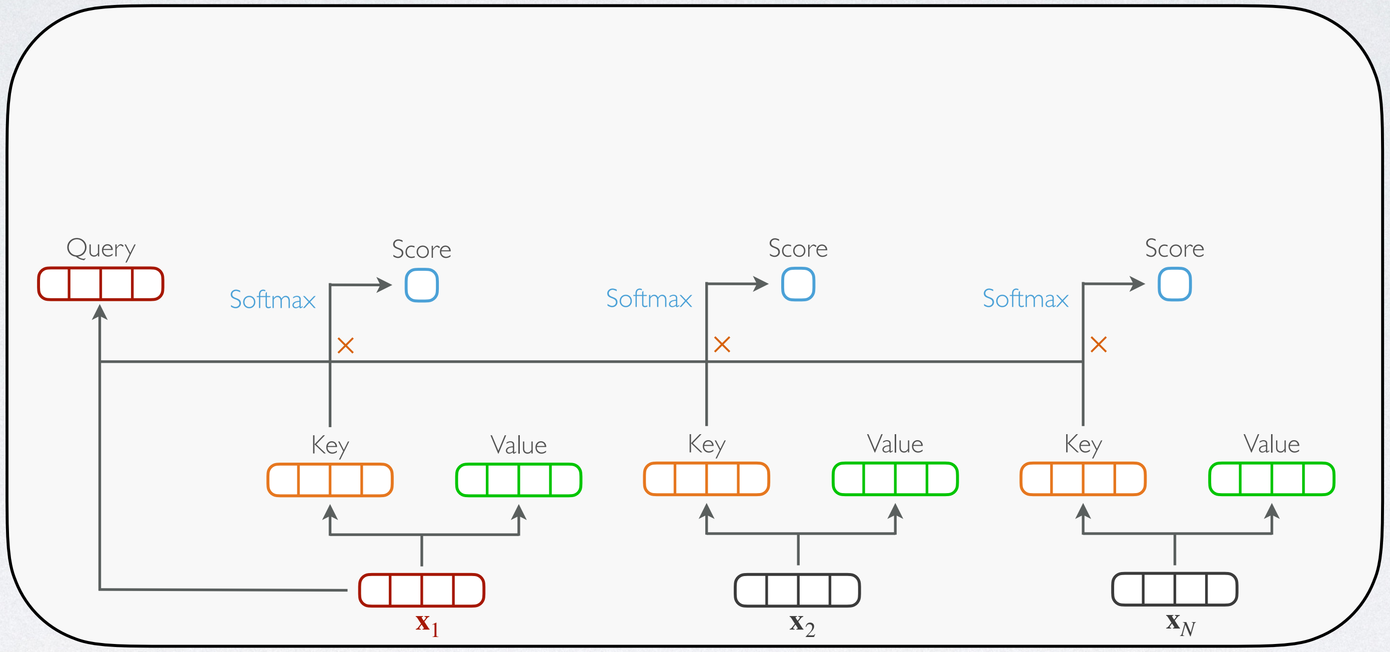


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

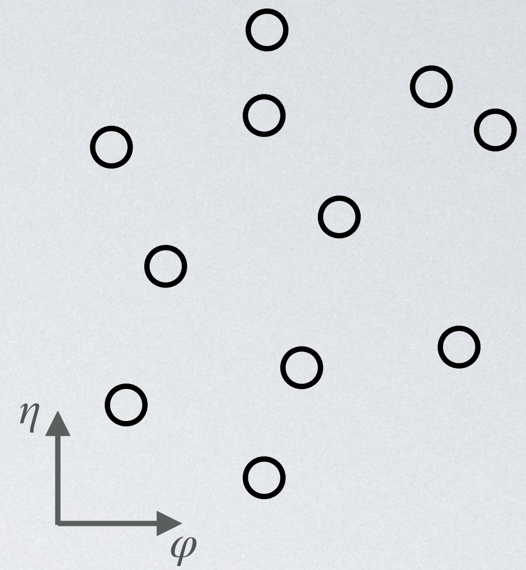
Self-attention (set transformers, Lee et al. ICML 2019)

Message passing



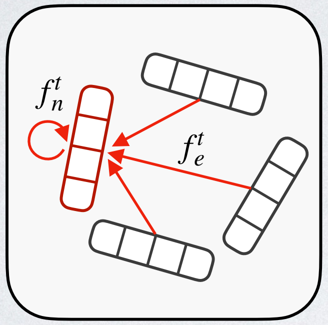


APPROACH 2: GAPT

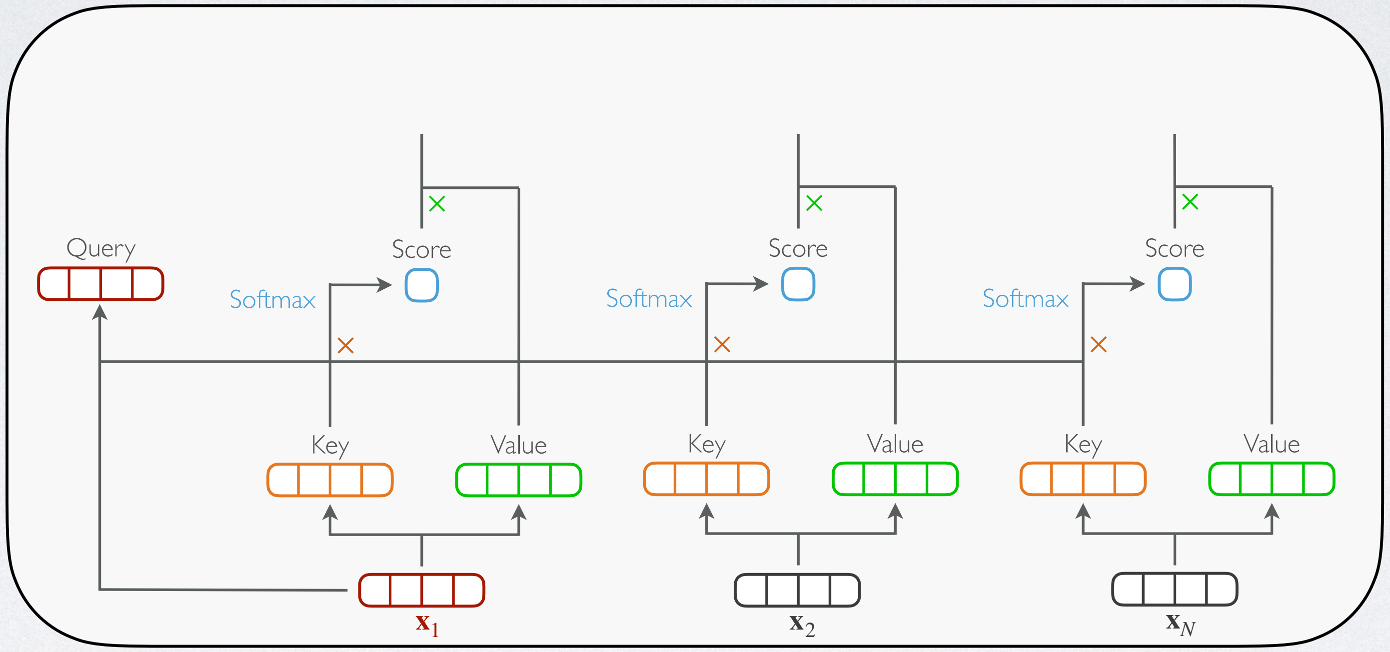


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing

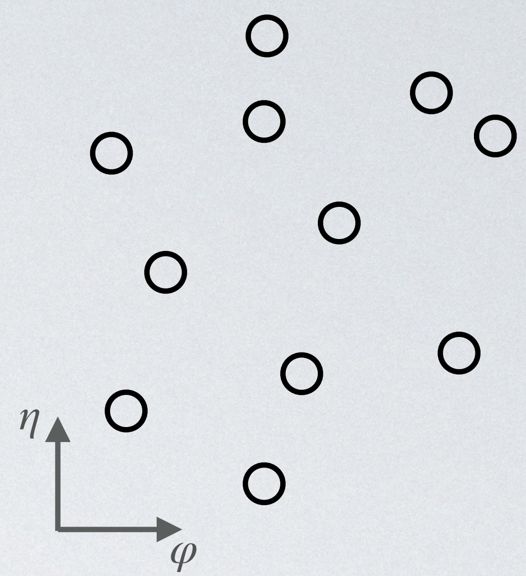


Self-attention (set transformers, Lee et al. ICML 2019)



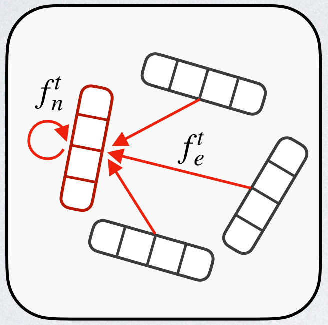
APPROACH 2: GAPT

RK et al., 2022

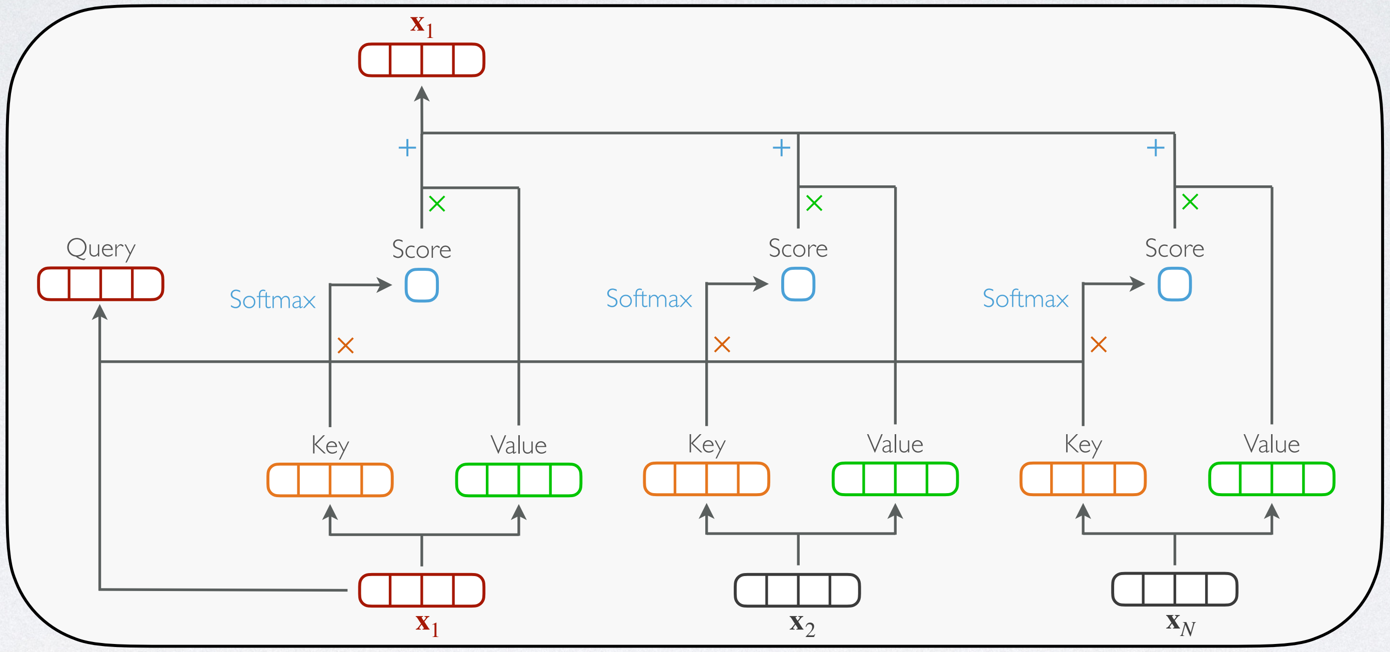


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing

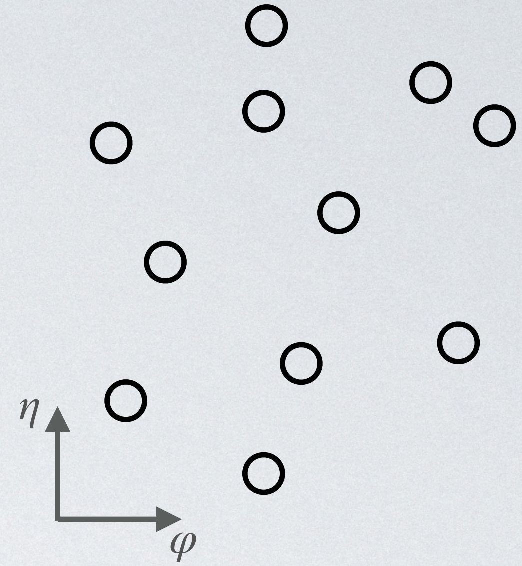


Self-attention (set transformers, Lee et al. ICML 2019)

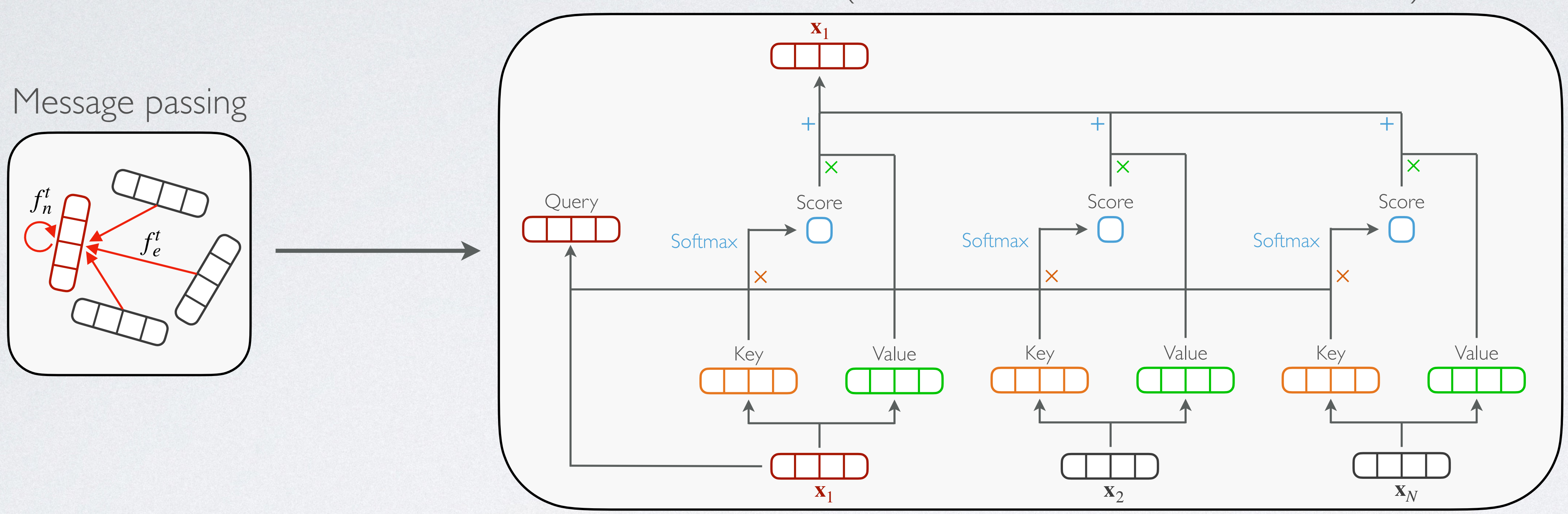


APPROACH 2: GAPT

RK et al., 2022



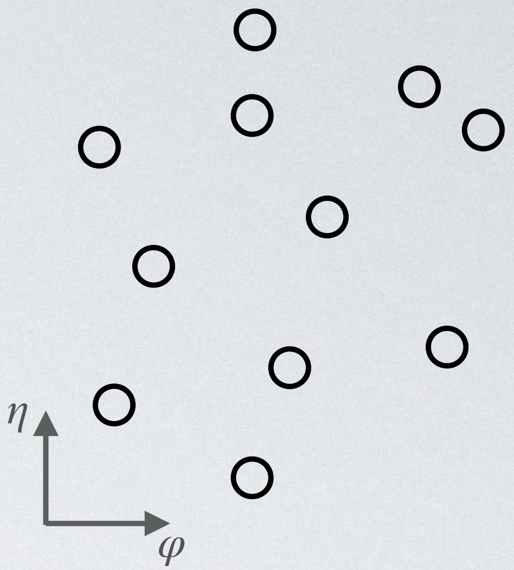
- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions



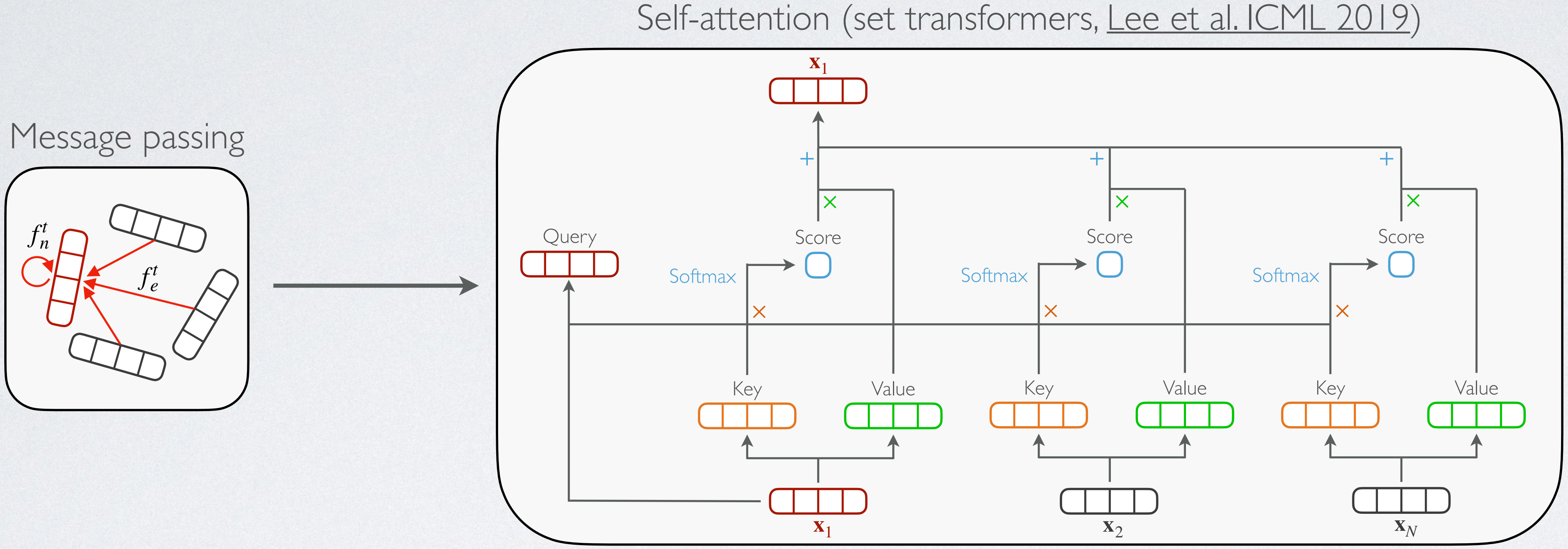
- Based on GAST (Stelzner et al. 2020), "Generative adversarial particle transformer" (GAPT)

RK et al., 2022

APPROACH 2: GAPT



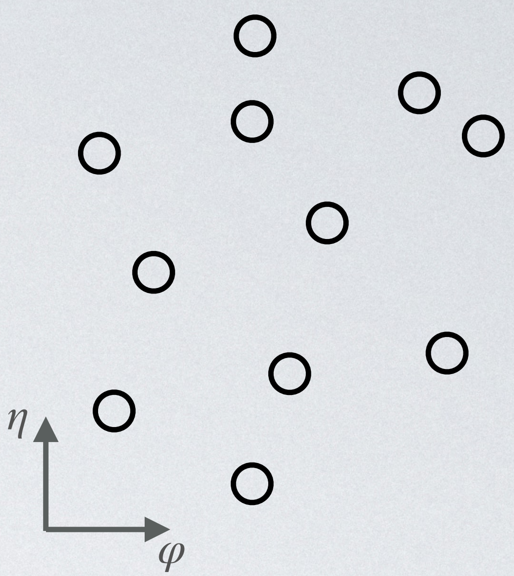
- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions



- Based on GAST (Stelzner et al. 2020), "Generative adversarial particle transformer" (GAPT)
- 5-15x faster than MPGAN

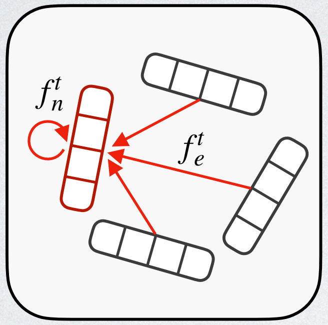
APPROACH 2: GAPT

[RK et al., 2022](#)

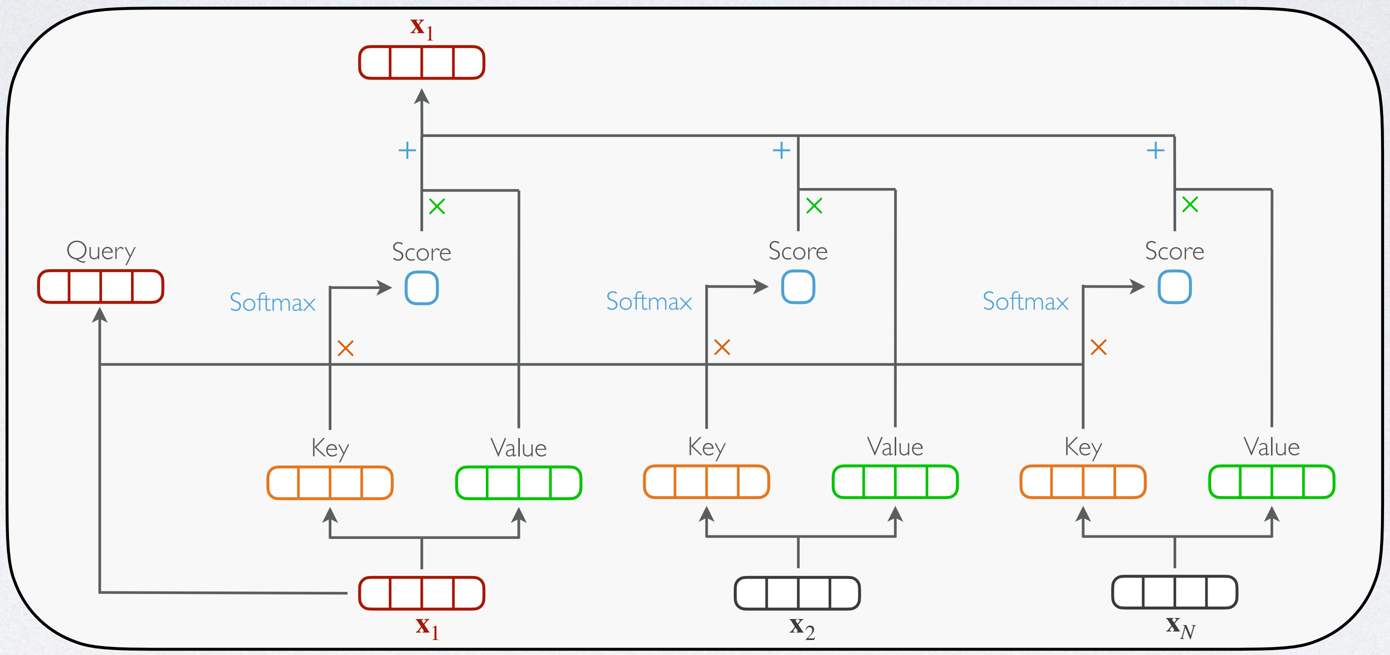


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing



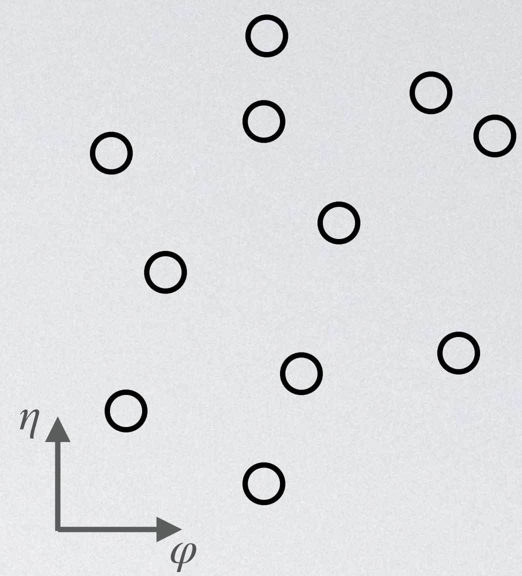
Self-attention (set transformers, Lee et al. ICML 2019)



- Based on GAST (Stelzner et al. 2020), "Generative adversarial particle transformer" (GAPT)
- 5-15x faster than MPGAN
- MPGAN and naive GAPT scale as $O(N^2)$ with # of nodes

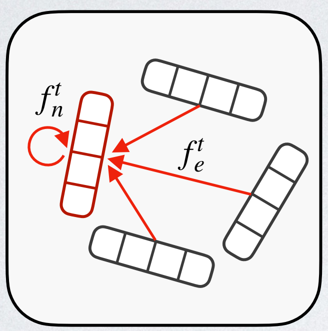
APPROACH 2: GAPT

RK et al., 2022

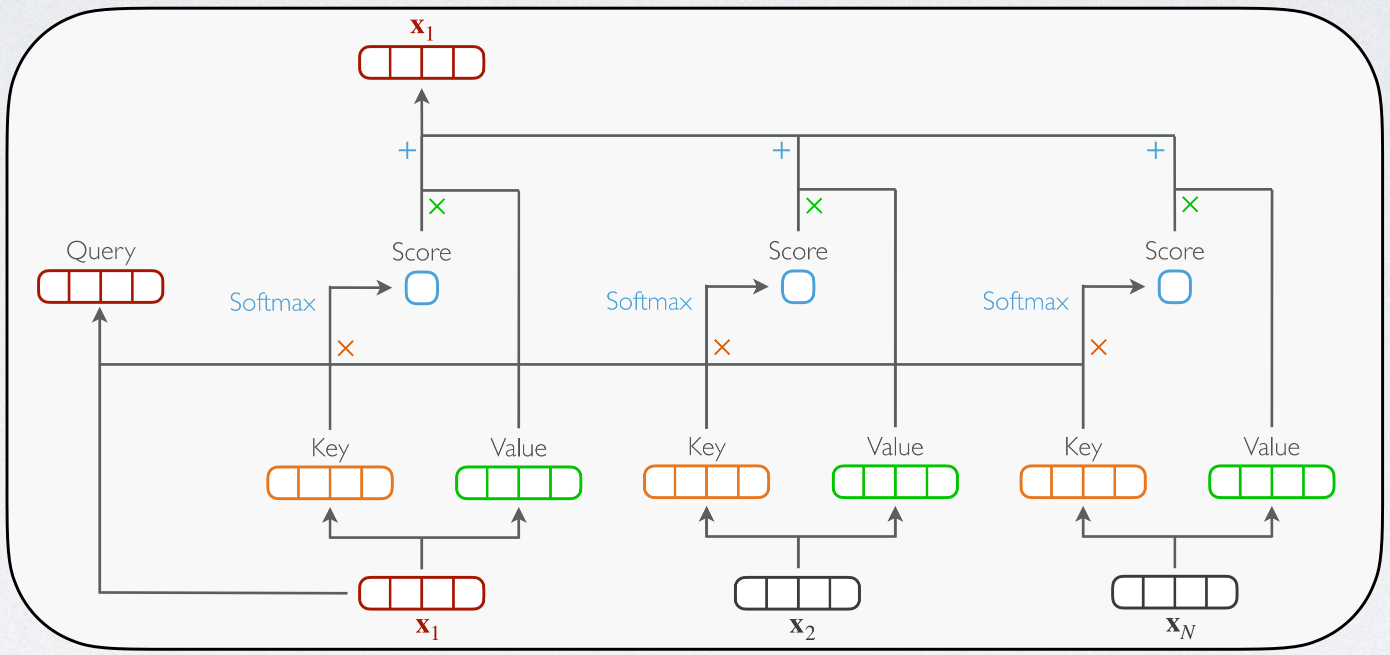


- Retain key ideas of MPGAN
  - Particle cloud data, fully connected particle interactions

Message passing

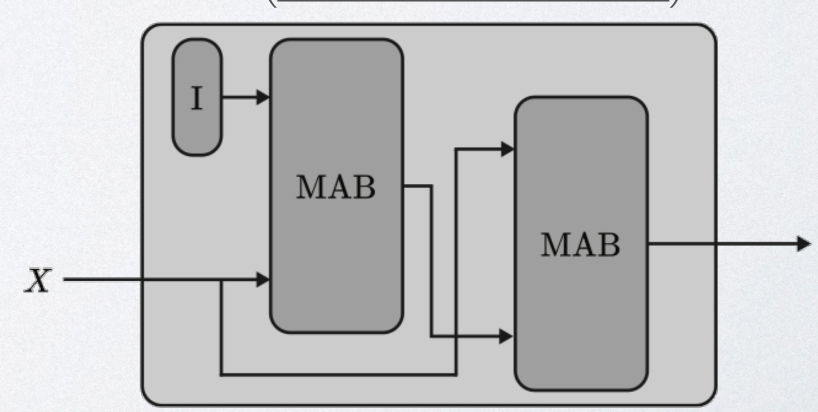


Self-attention (set transformers, Lee et al. ICML 2019)



- Based on GAST (Stelzner et al. 2020), "Generative adversarial particle transformer" (GAPT)
- 5-15x faster than MPGAN
- MPGAN and naive GAPT scale as $O(N^2)$ with # of nodes
- But linear scaling with induced self-attention blocks (ISAB)

ISAB (Lee et al. ICML 2019)

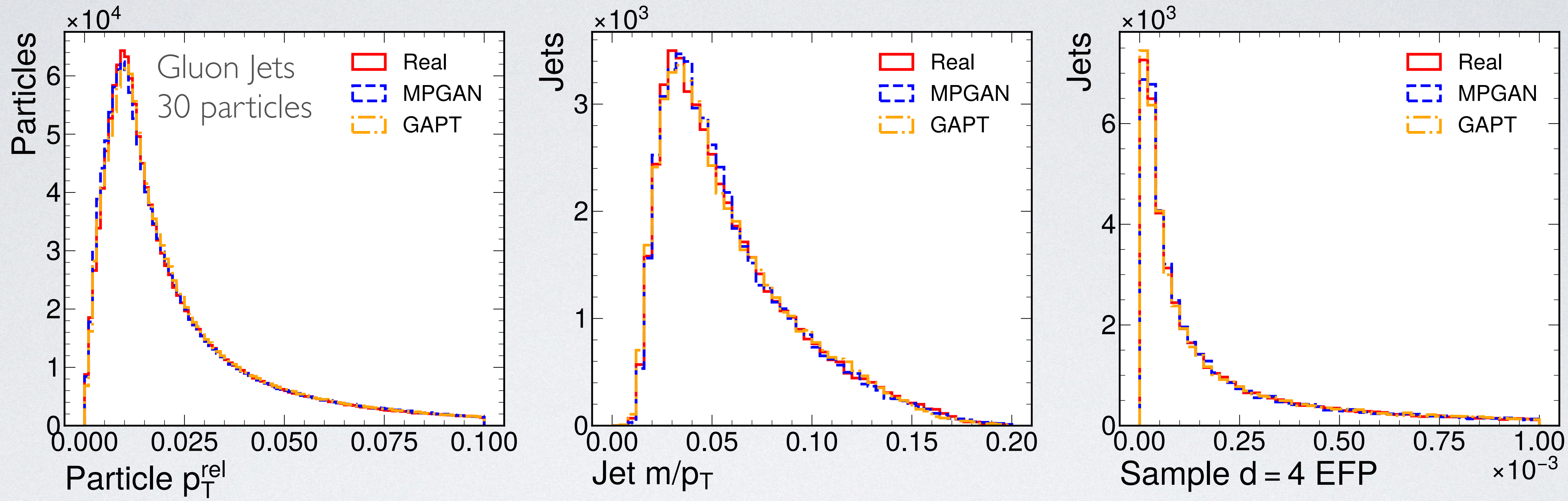


MPGAN VS GAPT

RK et al., 2022

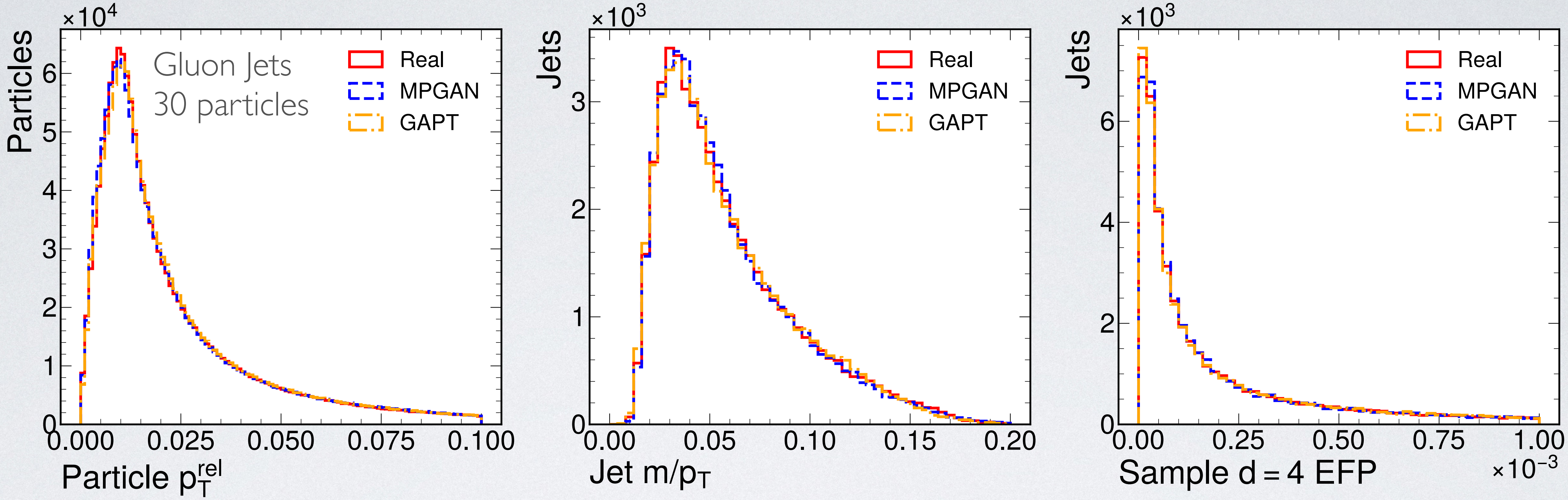
MPGAN VS GAPT

RK et al., 2022



MPGAN VS GAPT

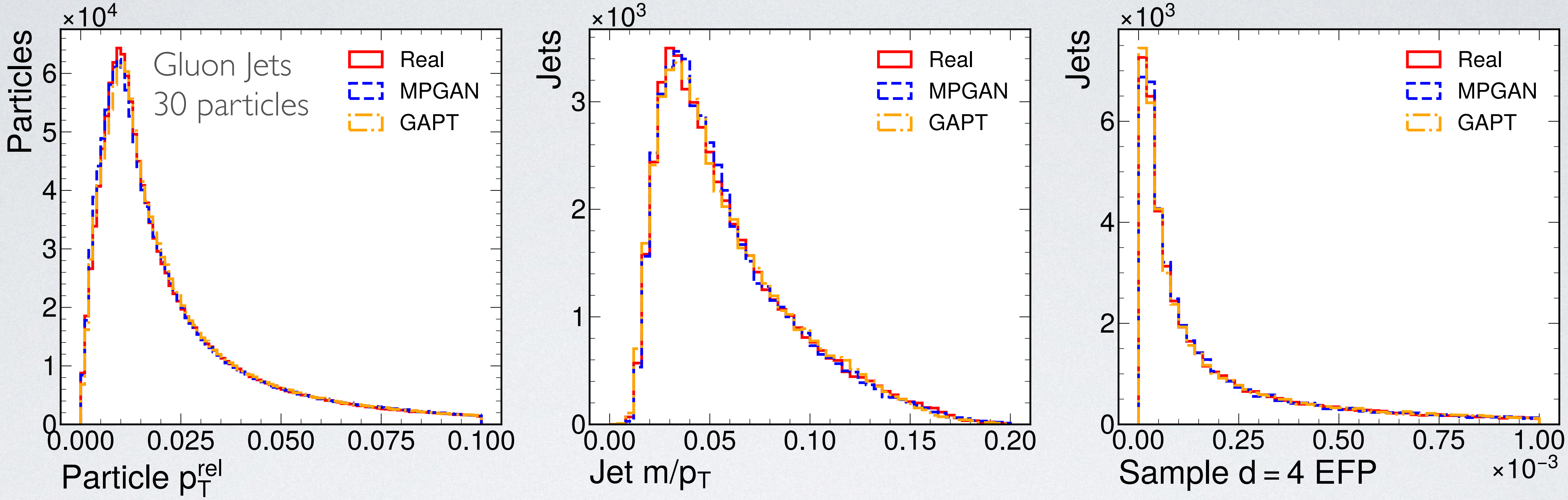
RK et al., 2022



- Both well performing, difficult to discern visually

RK et al., 2022

MPGAN VS GAPT



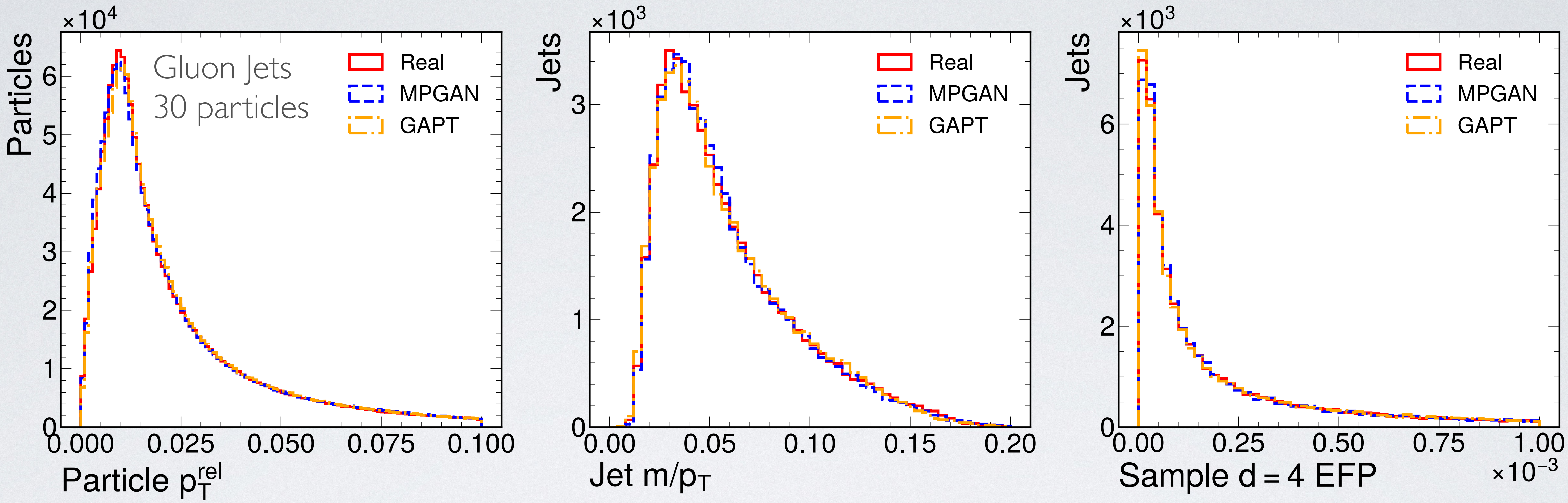
	FPD (10^{-3})	KPD (10^{-3})	W1-M(10^{-5})	Inference time per jet (μs)*
Truth	0.08 ± 0.03	−0.006 ± 0.005	0.28 ± 0.05	-
MPGAN	0.30 ± 0.06	−0.001 ± 0.004	0.54 ± 0.06	41
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08	9

- Both well performing, difficult to discern visually

*On an A6000

MPGAN VS GAPT

RK et al., 2022



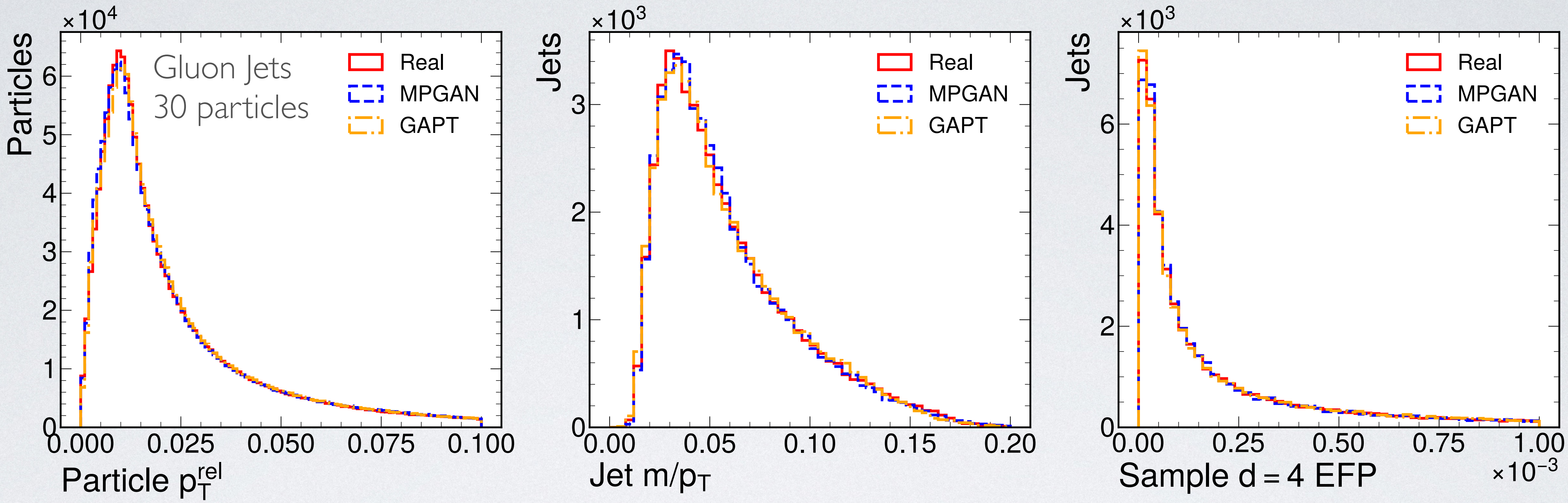
	FPD (10^{-3})	KPD (10^{-3})	W1-M(10^{-5})	Inference time per jet (μs)*
Truth	0.08 ± 0.03	−0.006 ± 0.005	0.28 ± 0.05	-
MPGAN	0.30 ± 0.06	−0.001 ± 0.004	0.54 ± 0.06	41
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08	9

- Both well performing, difficult to discern visually
- FPD necessary to differentiate performance - MPGAN samples are higher quality

*On an A6000

MPGAN VS GAPT

RK et al., 2022

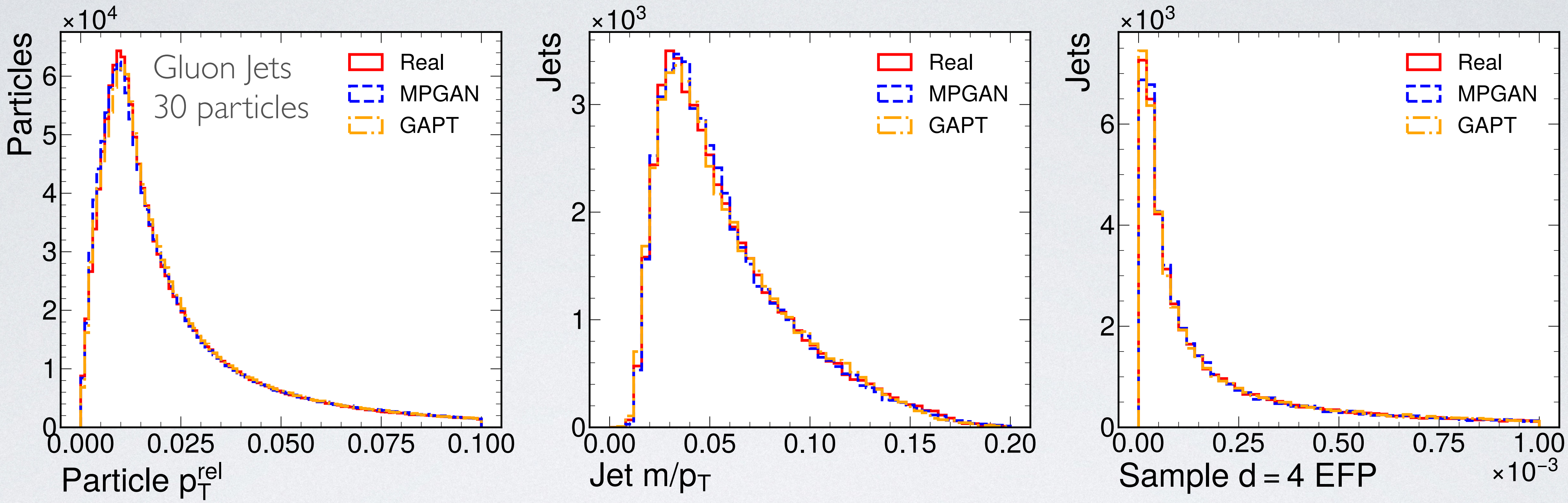


	FPD (10^{-3})	KPD (10^{-3})	W1-M(10^{-5})	Inference time per jet (μs)*
Truth	0.08 ± 0.03	−0.006 ± 0.005	0.28 ± 0.05	-
MPGAN	0.30 ± 0.06	−0.001 ± 0.004	0.54 ± 0.06	41
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08	9

- Both well performing, difficult to discern visually
- FPD necessary to differentiate performance - MPGAN samples are higher quality
- FPD and W1-M show MPGAN isn't perfectly compatible with true jets yet

*On an A6000

MPGAN VS GAPT



	FPD (10^{-3})	KPD (10^{-3})	W1-M(10^{-5})	Inference time per jet (μs)*
Truth	0.08 ± 0.03	−0.006 ± 0.005	0.28 ± 0.05	-
MPGAN	0.30 ± 0.06	−0.001 ± 0.004	0.54 ± 0.06	41
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08	9

- Both well performing, difficult to discern visually
- FPD necessary to differentiate performance - MPGAN samples are higher quality
- FPD and W1-M show MPGAN isn't perfectly compatible with true jets yet
- GAPT is significantly faster, both $O(10^4)$ faster than FullSim

*On an A6000

CONCLUSION

CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP

CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT

CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently

CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently
- GAPT significantly faster, promising avenue for scaling to large clouds

CONCLUSION

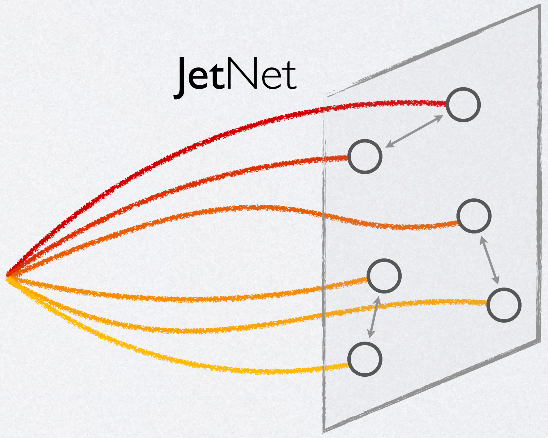
- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently
- GAPT significantly faster, promising avenue for scaling to large clouds
- Next steps:

CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently
- GAPT significantly faster, promising avenue for scaling to large clouds
- Next steps:
  - Discuss metrics with FastSim community

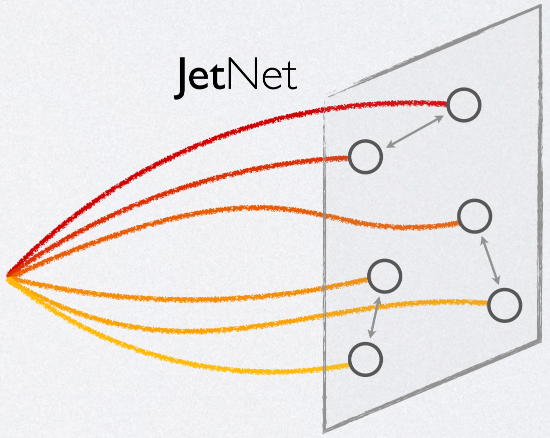
CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently
- GAPT significantly faster, promising avenue for scaling to large clouds
- Next steps:
  - Discuss metrics with FastSim community
  - FPD and KPD will be added to JetNet for easy, standard use



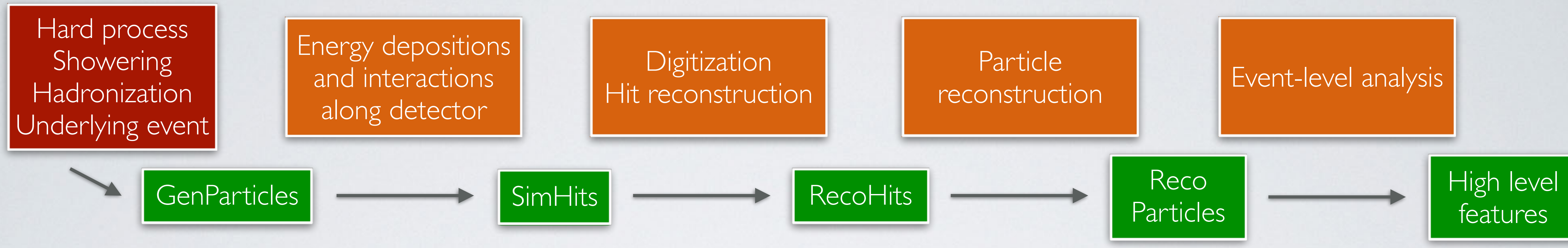
CONCLUSION

- Propose Fréchet and kernel physics distances (FPD and KPD) for evaluating generative models in HEP
- Developed two particle cloud simulators: graph-based MPGAN, attention-based GAPT
- Both very high performing, MPGAN has the edge currently
- GAPT significantly faster, promising avenue for scaling to large clouds
- Next steps:
  - Discuss metrics with FastSim community
  - FPD and KPD will be added to JetNet for easy, standard use
  - Extend GAPT to larger clouds, more datasets (esp. calorimeter showers)

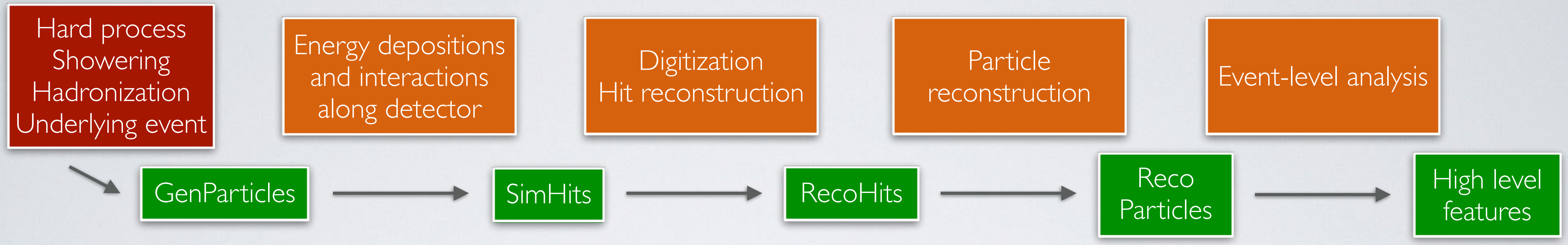


BACKUP

LHC SIMULATIONS

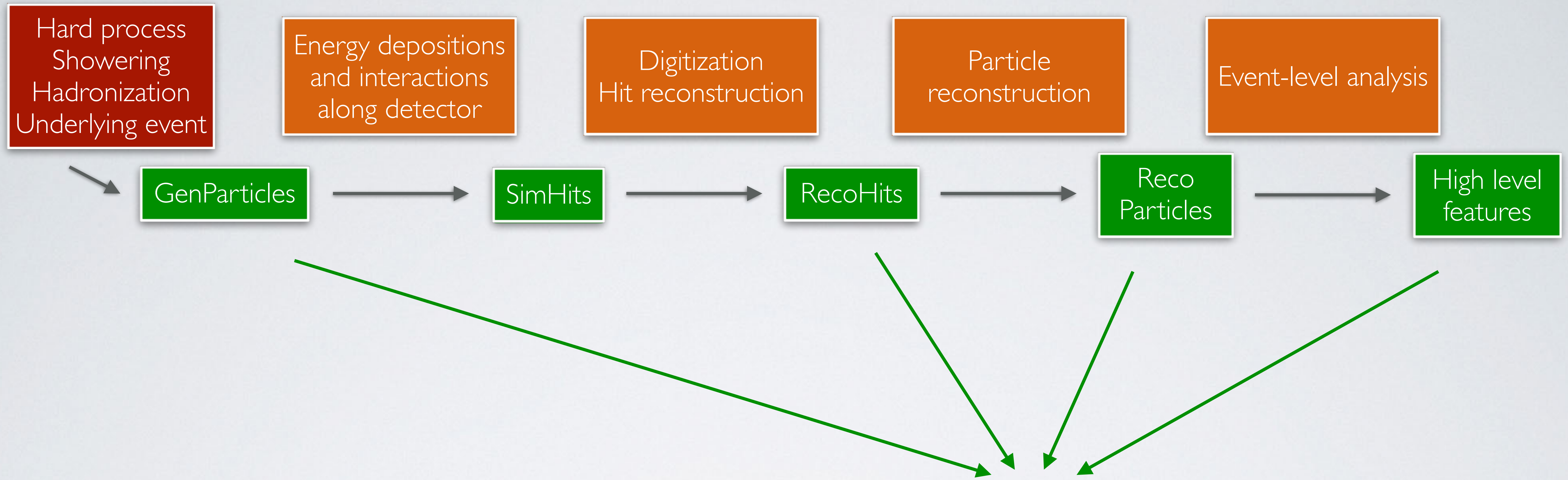


LHC SIMULATIONS



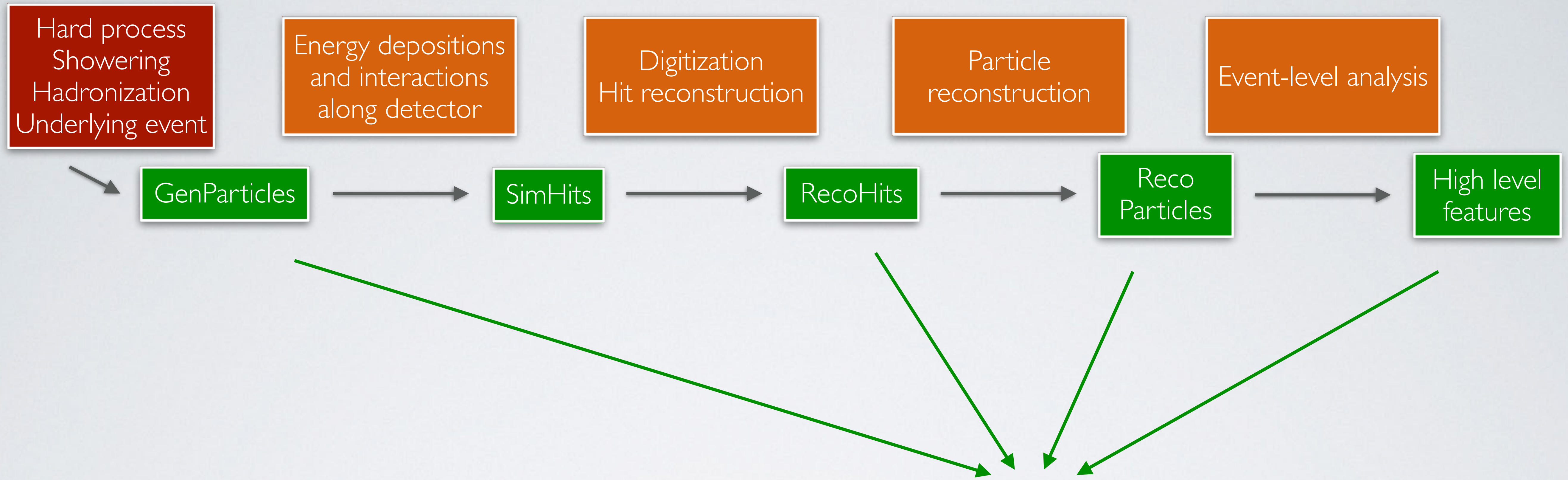
- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$

LHC SIMULATIONS



- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$

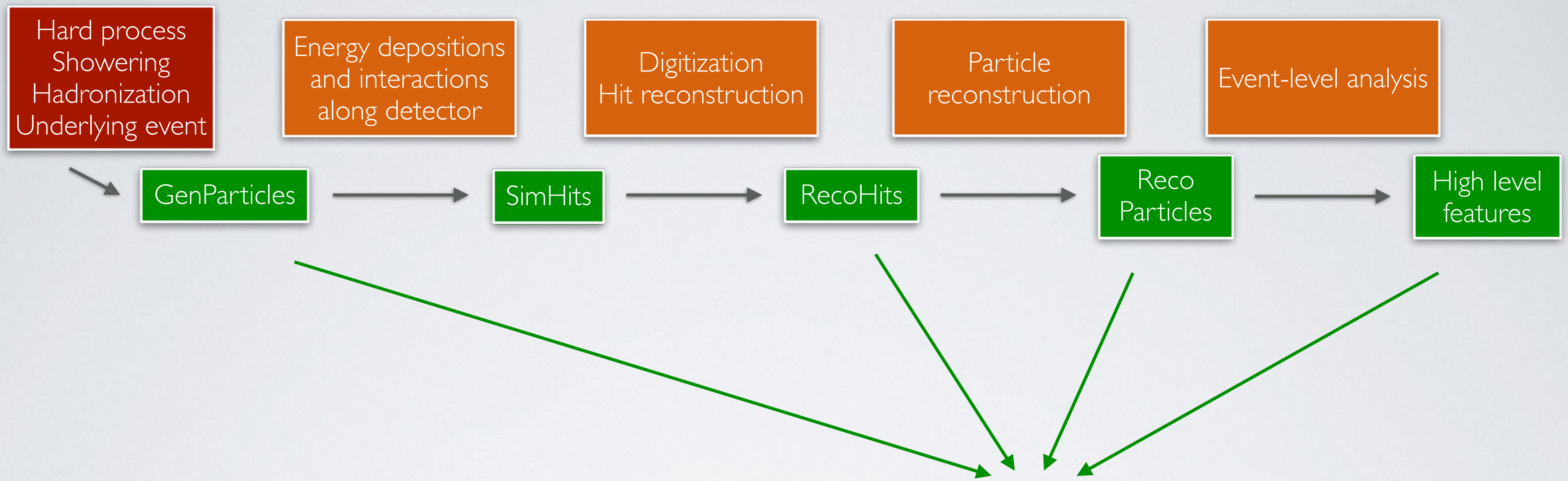
LHC SIMULATIONS



- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$

- Rich area in machine learning: **deep generative models**

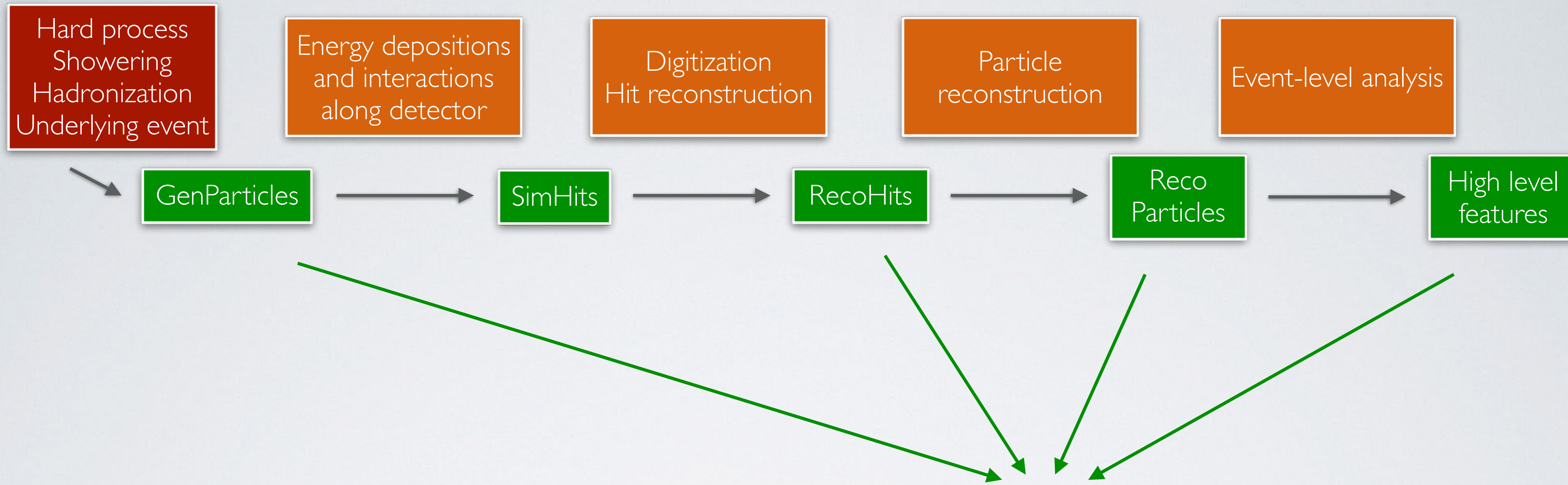
LHC SIMULATIONS



- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$

- Rich area in machine learning: **deep generative models**
    - Deep neural networks are flexible and expressive

LHC SIMULATIONS



- Want model $p_\theta(\mathbf{x})$ for underlying data distribution $p(\mathbf{x})$
- Rich area in machine learning: **deep generative models**
  - Deep neural networks are flexible and expressive
  - $p_\theta(\mathbf{x})$ typically modelled with high-capacity DNNs

MORE ON IPMS

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

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)

MORE ON IPMS

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

- Fréchet Gaussian Distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations

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

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

MORE ON IPMS

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

- Fréchet Gaussian Distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID)

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

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

MORE ON IPMS

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

- Fréchet Gaussian Distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID)
  - Computationally efficient

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

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

MORE ON IPMS

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

- Fréchet Gaussian Distance (FGD)
  - Fréchet / W_2 distance between multivariate Gaussian fitted to observations
  - Standard in computer vision (FID)
  - Computationally efficient
  - Gaussian assumption

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

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

MORE ON IPMS

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

- 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

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

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):

MORE ON IPMS

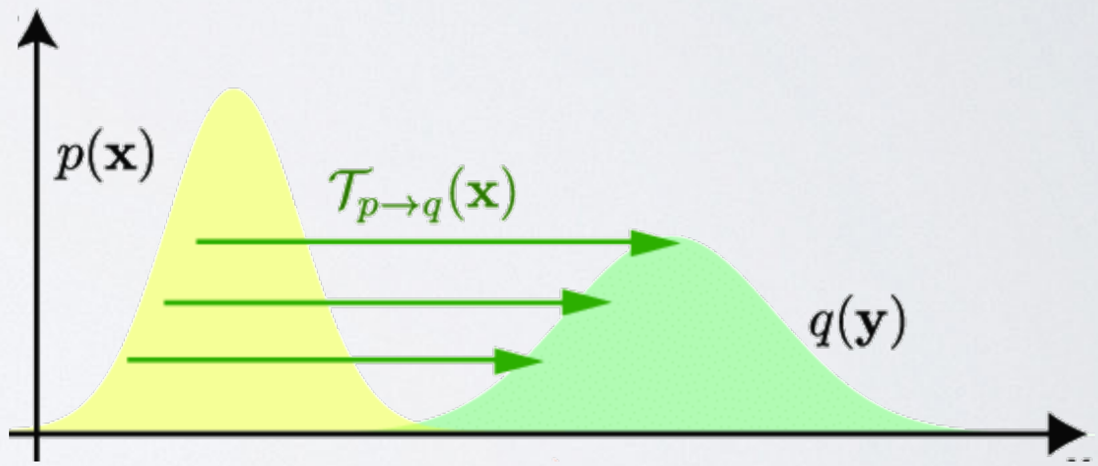
$$\sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)|$$

- Wasserstein $p-$distances (W_p):
  - $\mathcal{F}$ is all K-Lipschitz functions

MORE ON IPMS

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

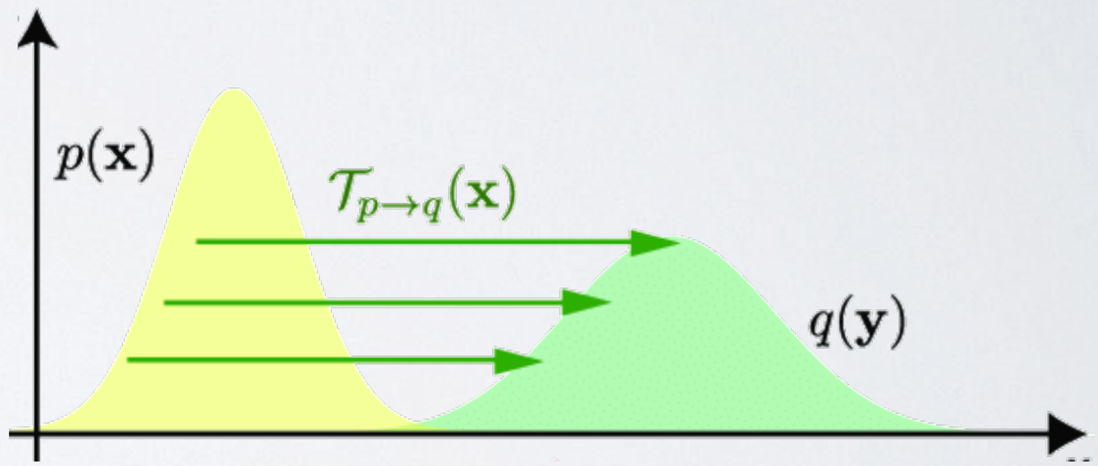
- Wasserstein $p-$distances (W_p):
    - $\mathcal{F}$ is all K-Lipschitz functions
    - "Work" needed to transport probability mass



MORE ON IPMS

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

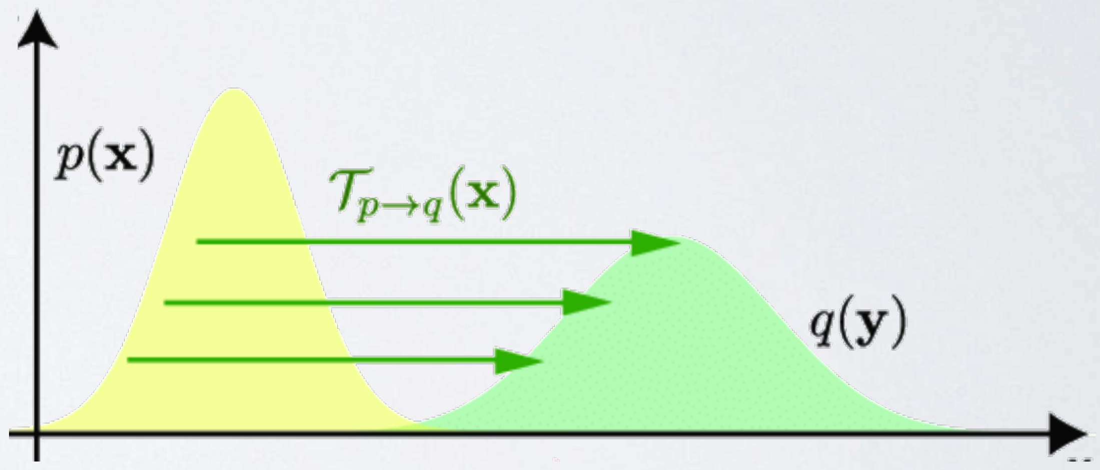
- Wasserstein $p-$distances (W_p):
  - $\mathcal{F}$ is all K-Lipschitz functions
  - "Work" needed to transport probability mass
  - Sensitive to quality and diversity



MORE ON IPMS

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

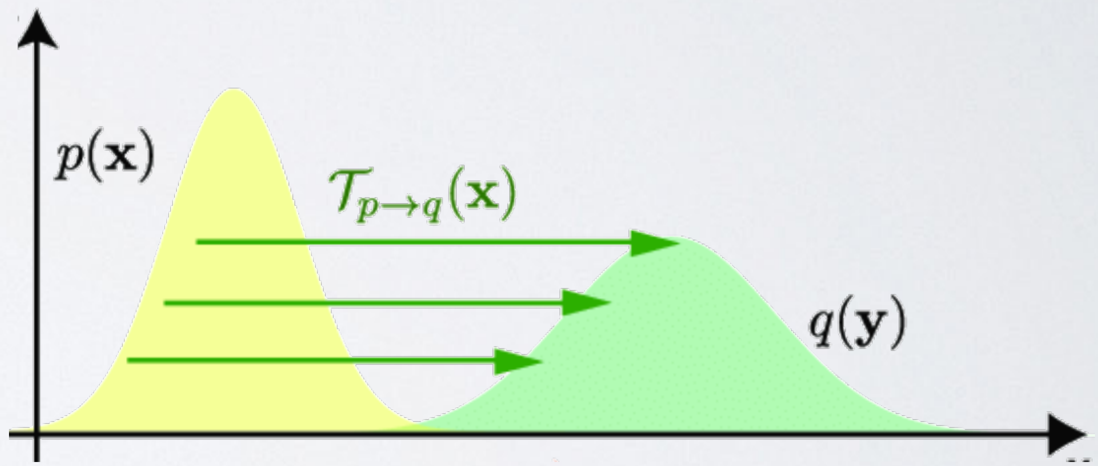
- Wasserstein $p-$distances (W_p):
  - $\mathcal{F}$ is all K-Lipschitz functions
  - "Work" needed to transport probability mass
  - Sensitive to quality and diversity
  - Computationally challenging for large N, D



MORE ON IPMS

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

- Wasserstein $p-$distances (W_p):
  - $\mathscr{F}$ 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

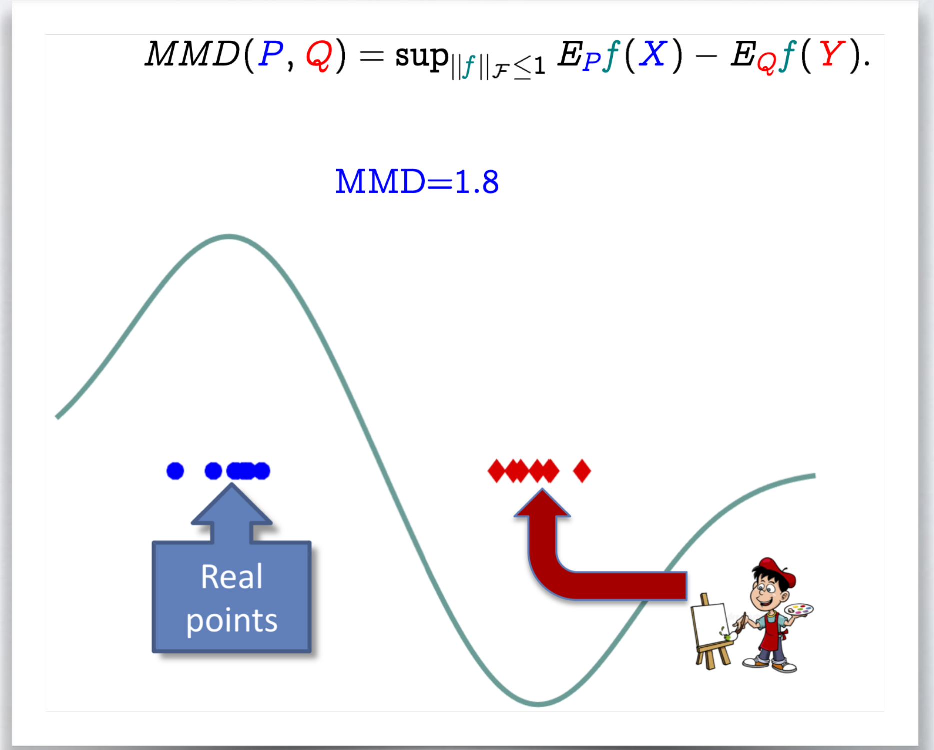
$$\sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim p_{\text{real}}} f(x) - \mathbb{E}_{y \sim p_{\text{gen}}} f(y)|$$

MORE ON IPMS

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

- Maximum mean discrepancy (MMD)

Gretton 2020

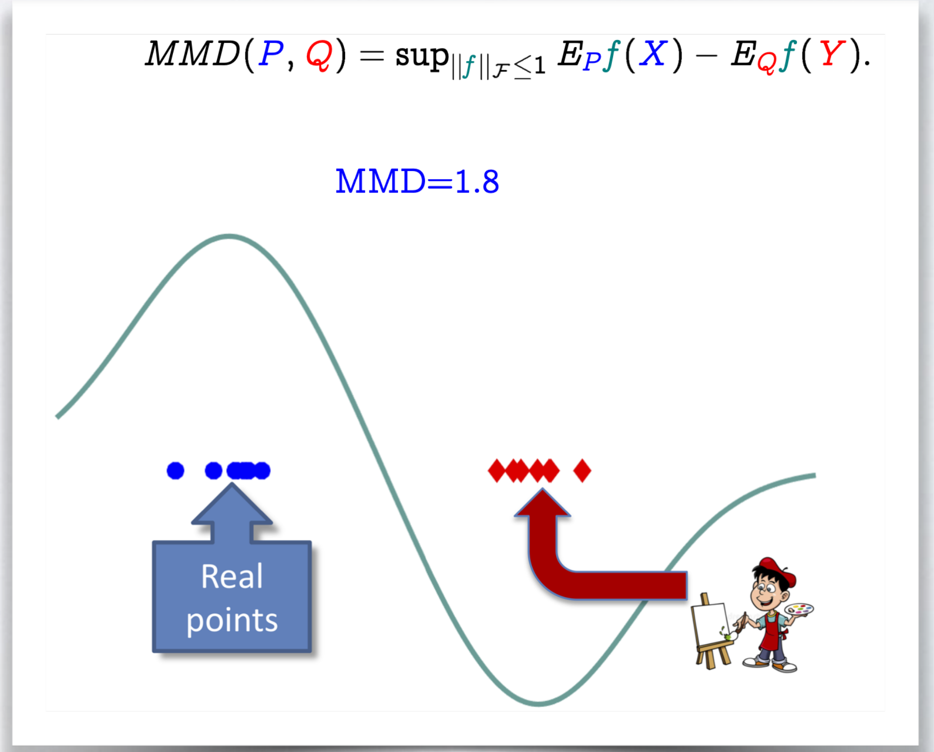


MORE ON IPMS

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

- Maximum mean discrepancy (MMD)
  - $\mathscr{F}$ is reproducing Kernel Hilbert space (RKHS) for a chosen kernel $k(x, y)$

Gretton 2020

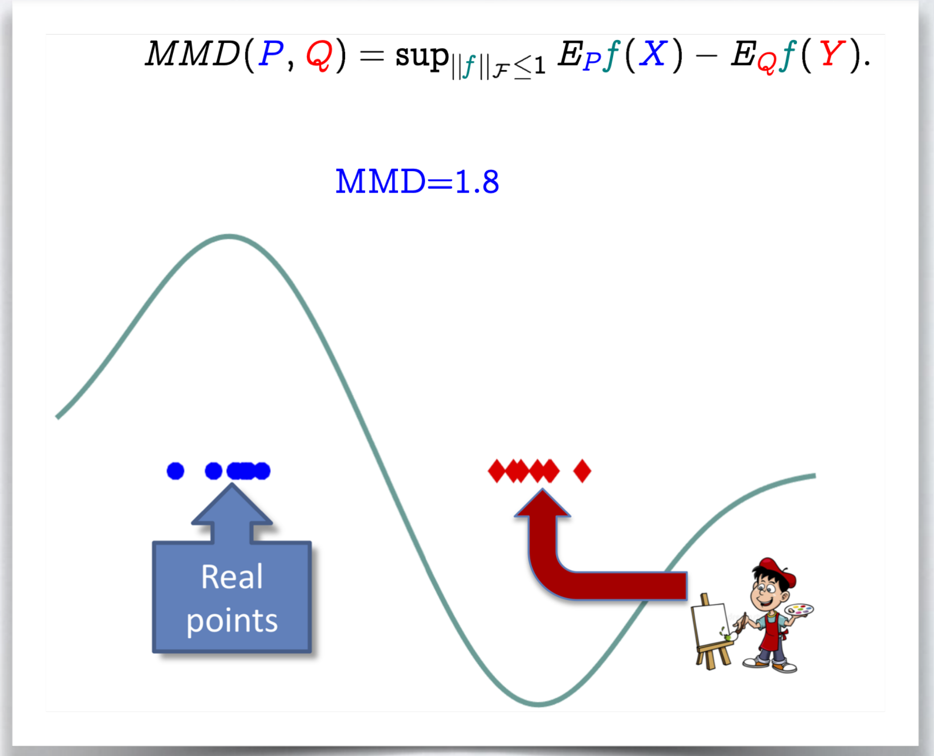


MORE ON IPMS

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

- 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 $\mathcal{F}$

Gretton 2020

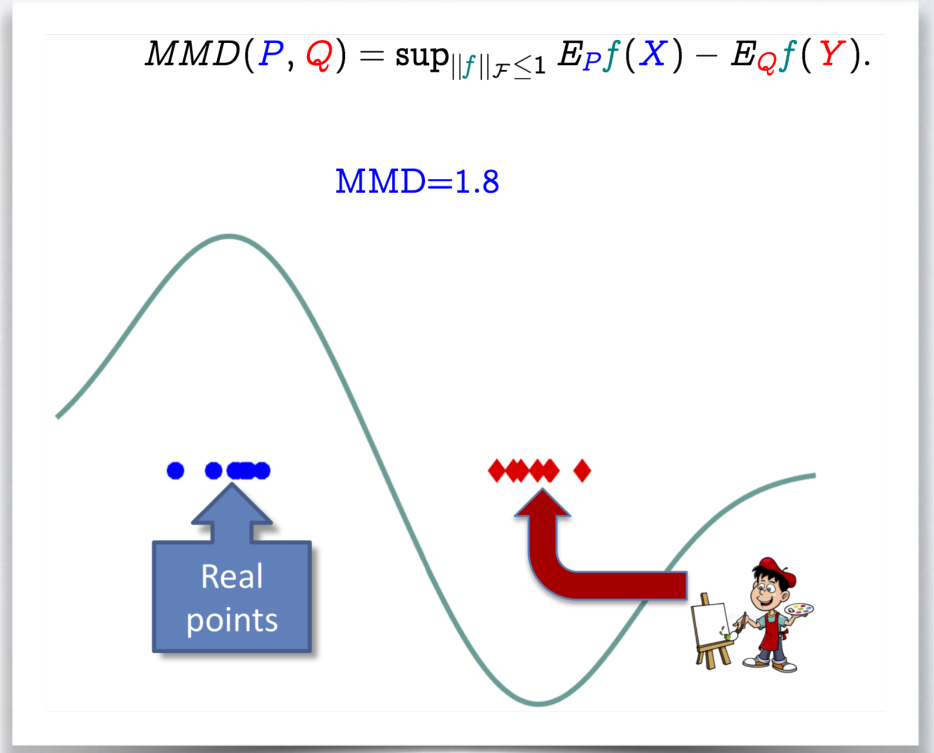


MORE ON IPMS

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

- 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

Gretton 2020

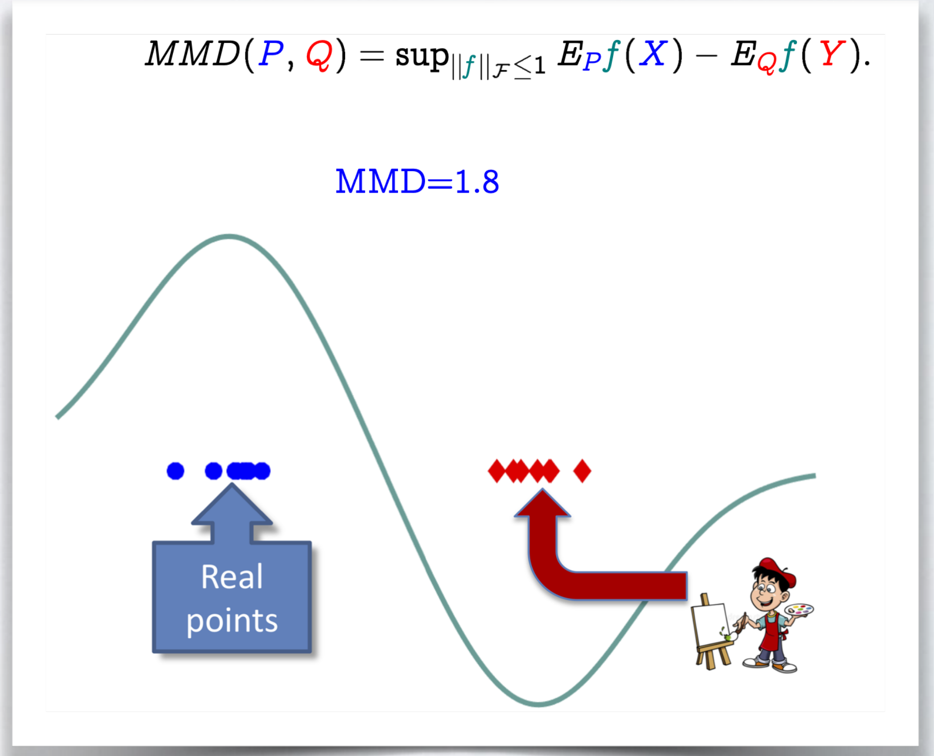


MORE ON IPMS

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

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

Gretton 2020

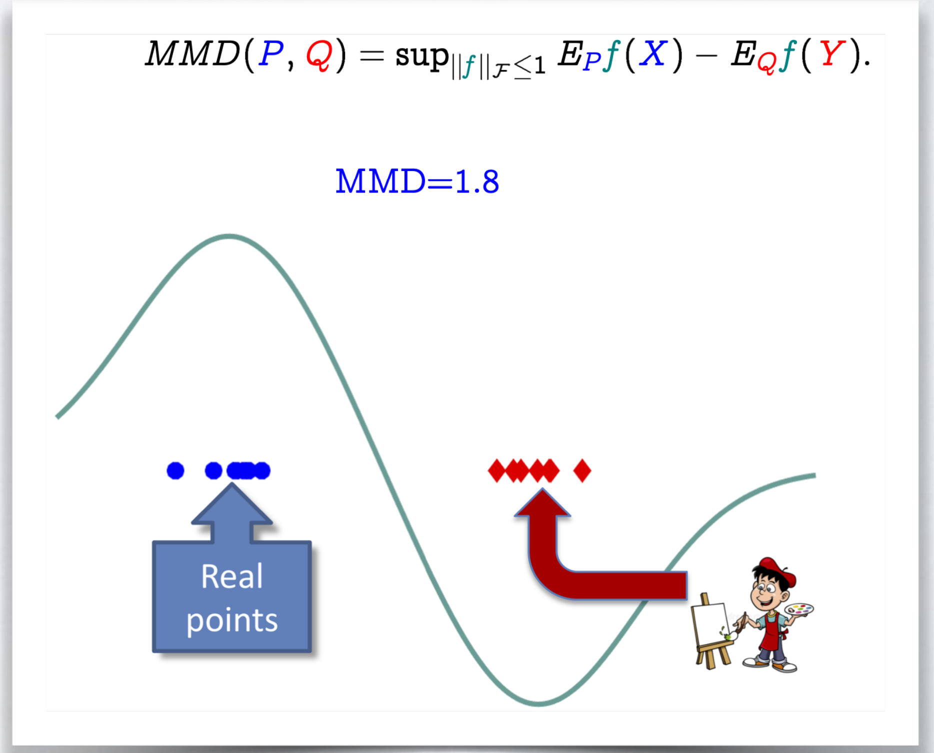


MORE ON IPMS

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

- 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

FRÉCHET <CLASSIFIER> DISTANCES

- Machine learning version of this: use classifier hidden features instead!

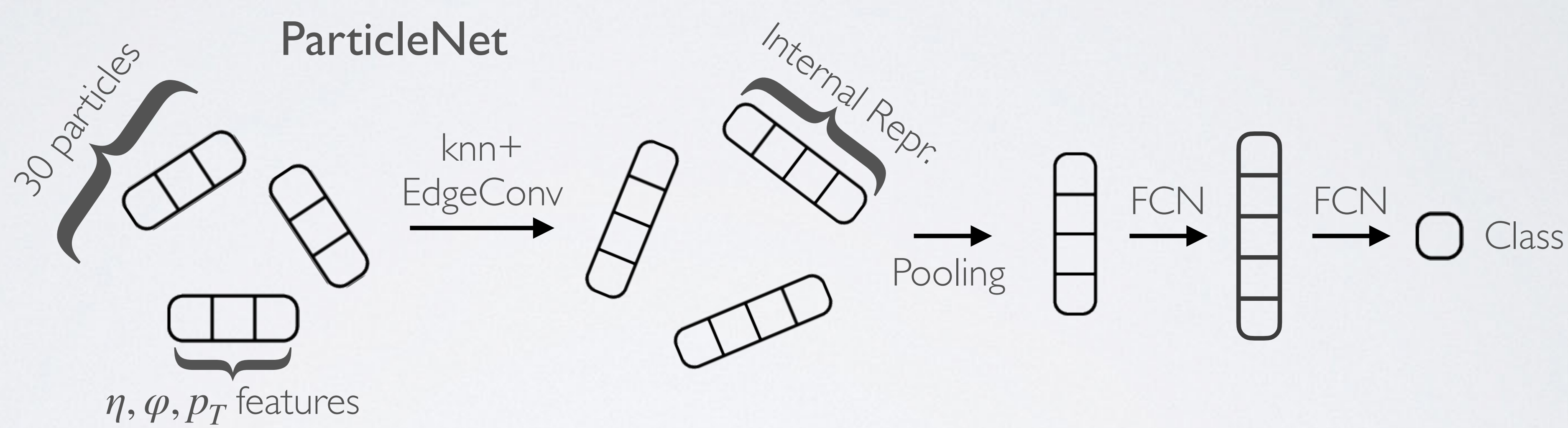
FRÉCHET <CLASSIFIER> DISTANCES

- 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:

FRÉCHET <CLASSIFIER> DISTANCES

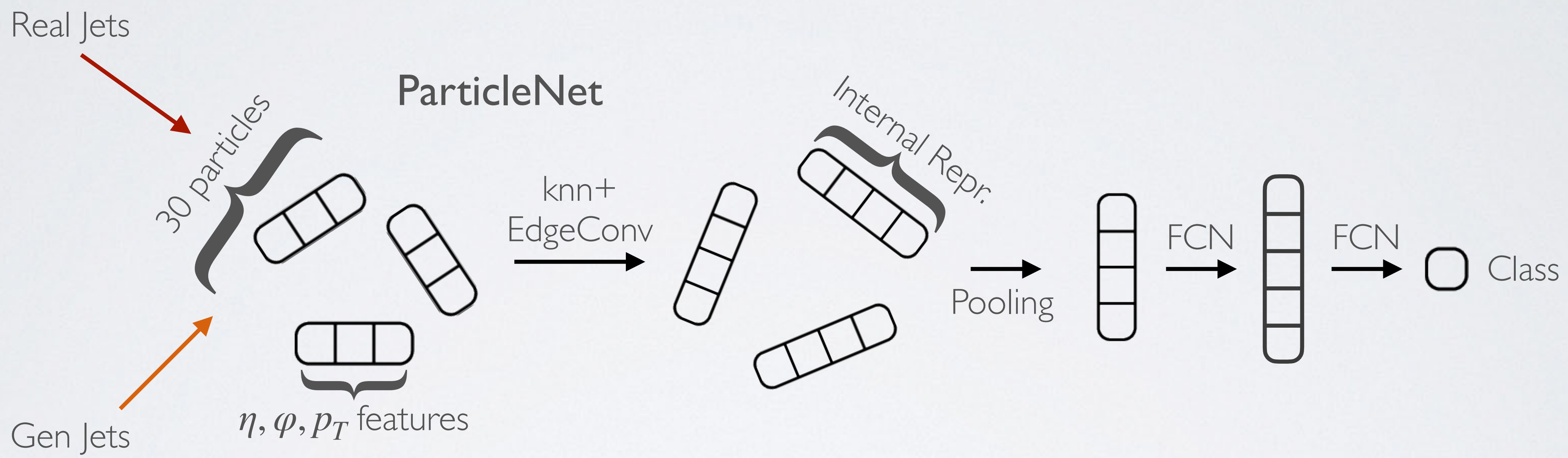
- Machine learning version of this: use classifier hidden features instead! Kansal et al., NeurIPS 2021

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



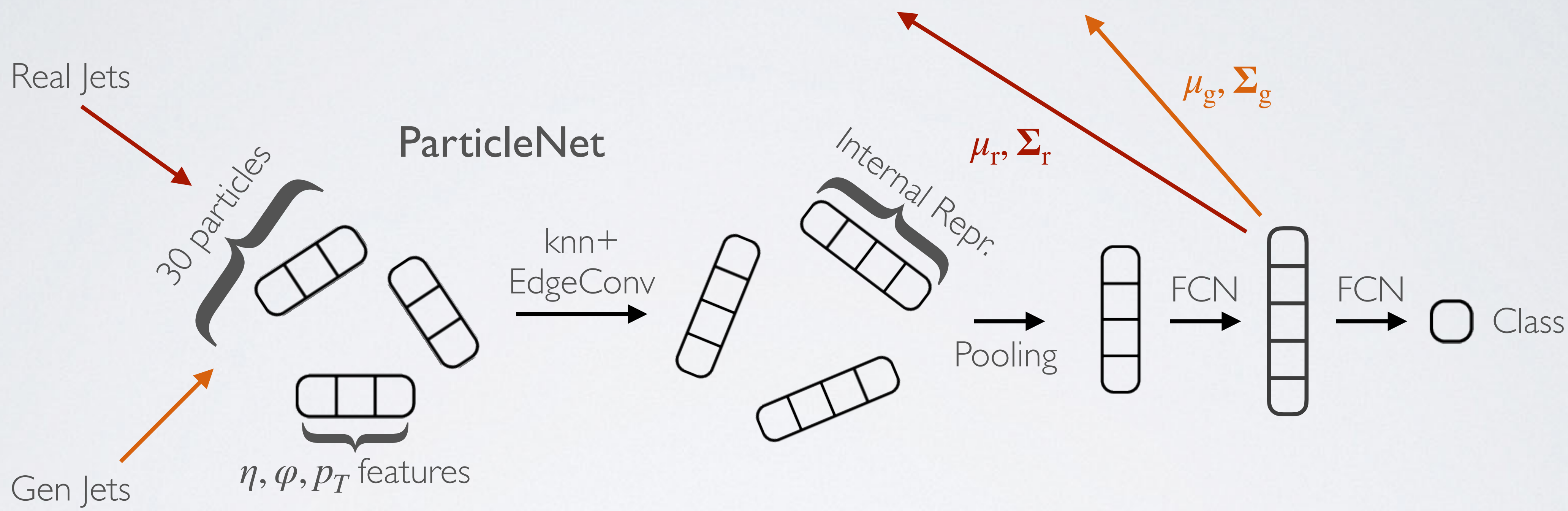
FRÉCHET <CLASSIFIER> DISTANCES

- 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:



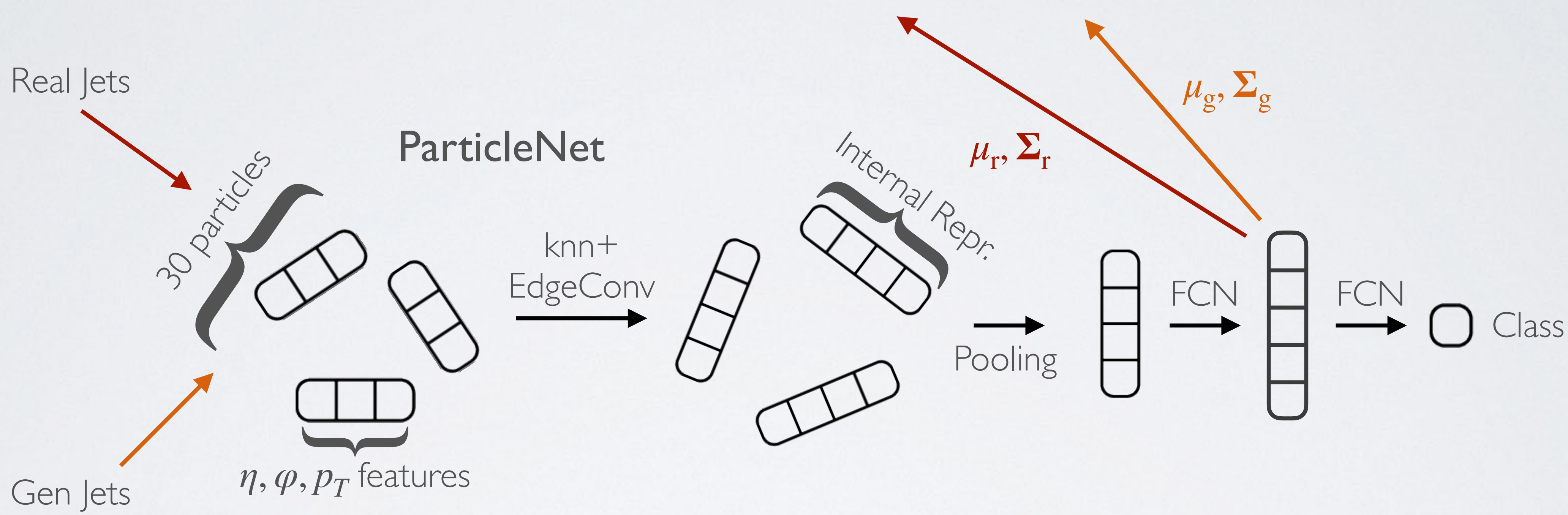
FRÉCHET <CLASSIFIER> DISTANCES

- 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:



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:

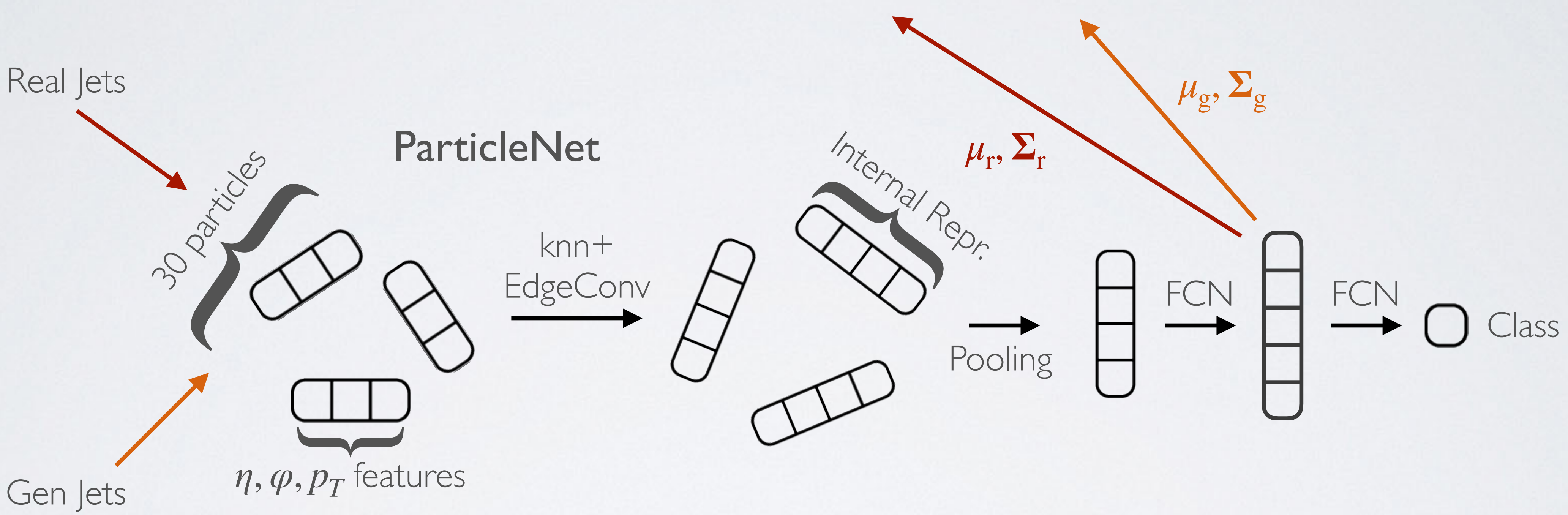


- High-performing classifier learns salient hidden features from data

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:



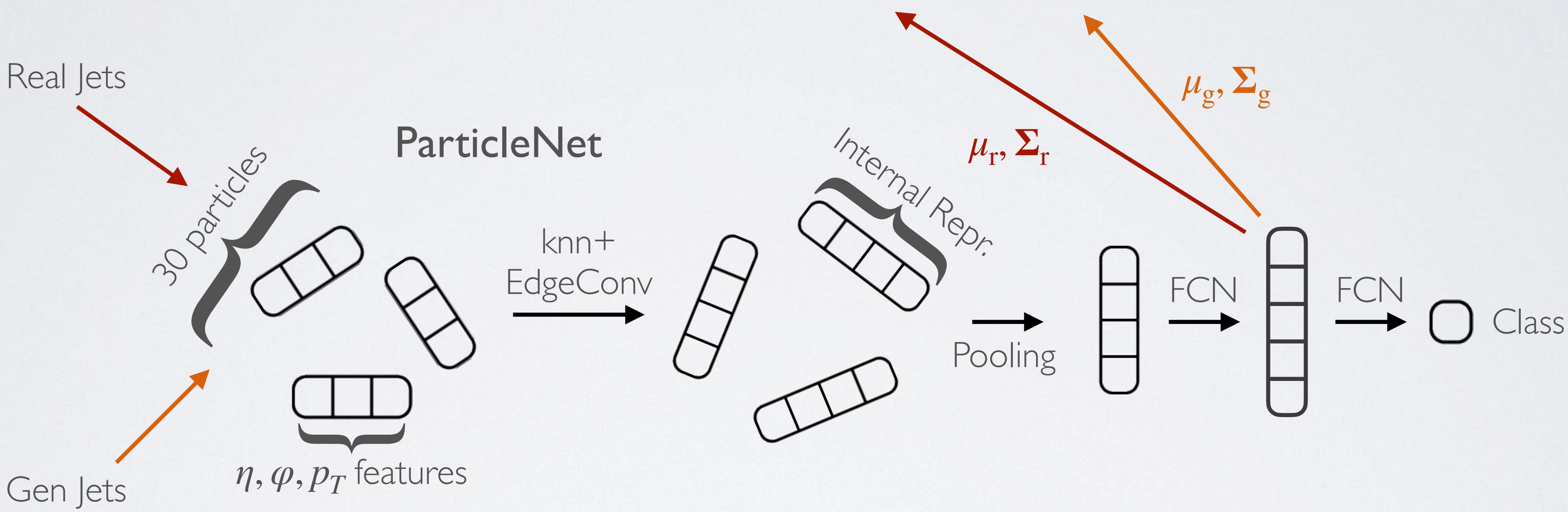
- High-performing classifier learns salient hidden features from data

- Retain sensitivity to **quality, diversity** from W_1, **reproducible** and **efficient** plus:

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:

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



- 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

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

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 $k(x, y)$

MAXIMUM MEAN DISCREPANCY

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

- IPM where $\mathcal{F}$ is unit ball in the reproducing kernel Hilbert space (RKHS) for kernel $k(x, y)$
  - RKHS $\Leftrightarrow f(x) = \langle f, \varphi(x) \rangle_{\mathcal{F}}$, where $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{F}}$

MAXIMUM MEAN DISCREPANCY

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

- IPM where $\mathcal{F}$ is unit ball in the reproducing kernel Hilbert space (RKHS) for kernel $k(x, y)$
  - RKHS $\Leftrightarrow f(x) = \langle f, \varphi(x) \rangle_{\mathcal{F}}$, where $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{F}}$
  - $\mathbb{E}_{x \sim p} f(x) = \langle f, \mathbb{E}_{x \sim p} \varphi(x) \rangle_{\mathcal{F}} = \langle f, \mu_p \rangle_{\mathcal{F}}$

MAXIMUM MEAN DISCREPANCY

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

- IPM where $\mathcal{F}$ is unit ball in the reproducing kernel Hilbert space (RKHS) for kernel $k(x, y)$
  - RKHS $\Leftrightarrow f(x) = \langle f, \varphi(x) \rangle_{\mathcal{F}}$, where $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{F}}$
  - $\mathbb{E}_{x \sim p} f(x) = \langle f, \mathbb{E}_{x \sim p} \varphi(x) \rangle_{\mathcal{F}} = \langle f, \mu_p \rangle_{\mathcal{F}}$
  - μ_p is the embedding of distribution p in $\mathcal{F}$

MAXIMUM MEAN DISCREPANCY

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

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

MAXIMUM MEAN DISCREPANCY

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

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

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

MAXIMUM MEAN DISCREPANCY

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

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

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

- MMD: distance between means in embedding space

MAXIMUM MEAN DISCREPANCY

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

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

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

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

TESTS FOR QUALITY / DIVERSITY

TESTS FOR QUALITY / DIVERSITY

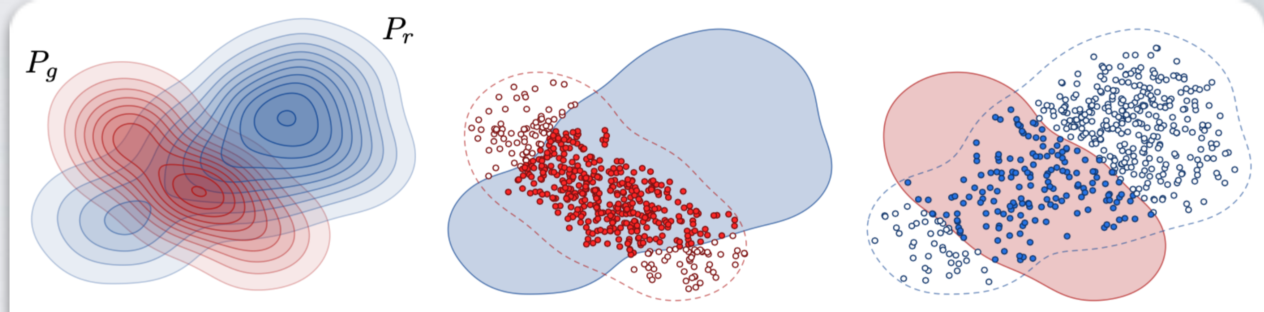
- Can be valuable to disentangle these

TESTS FOR QUALITY / DIVERSITY

- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)

TESTS FOR QUALITY / DIVERSITY

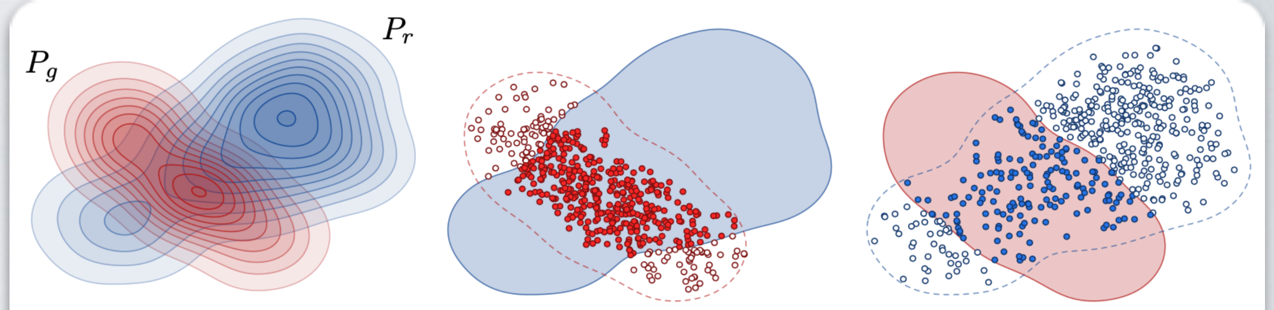
- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)
  - Estimate real and generated manifold using k-nearest-neighbours



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

TESTS FOR QUALITY / DIVERSITY

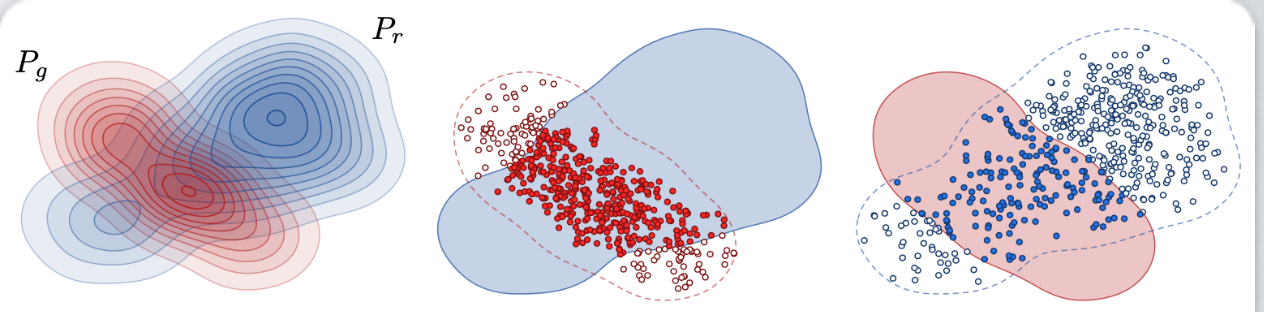
- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)
    - Estimate real and generated manifold using k-nearest-neighbours
    - Precision: fraction of generated samples lying within real manifold (quality)



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

TESTS FOR QUALITY / DIVERSITY

- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)

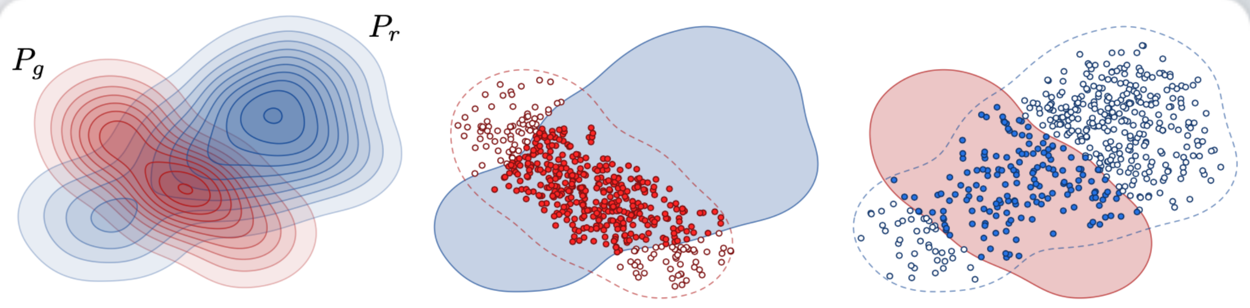


(a) Example distributions (b) Precision (c) Recall

  - 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)

TESTS FOR QUALITY / DIVERSITY

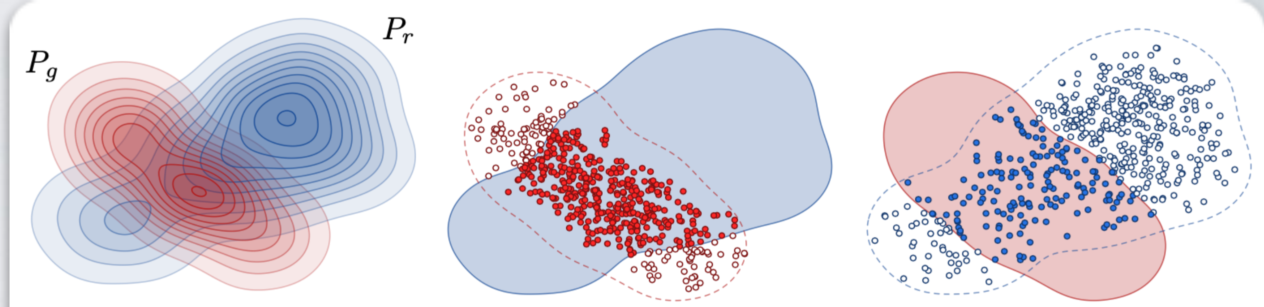
- Can be valuable to disentangle these



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

- 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)

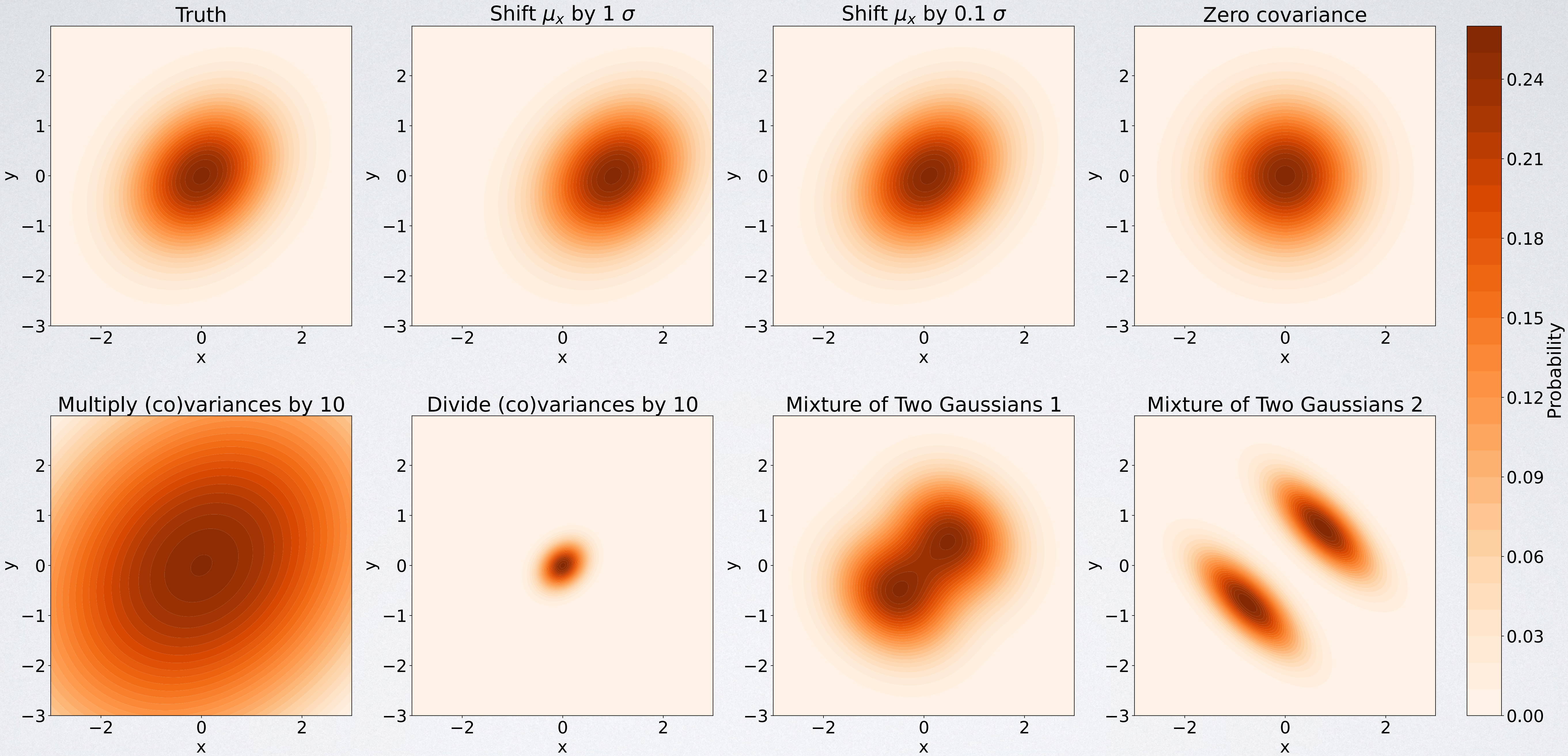
TESTS FOR QUALITY / DIVERSITY



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

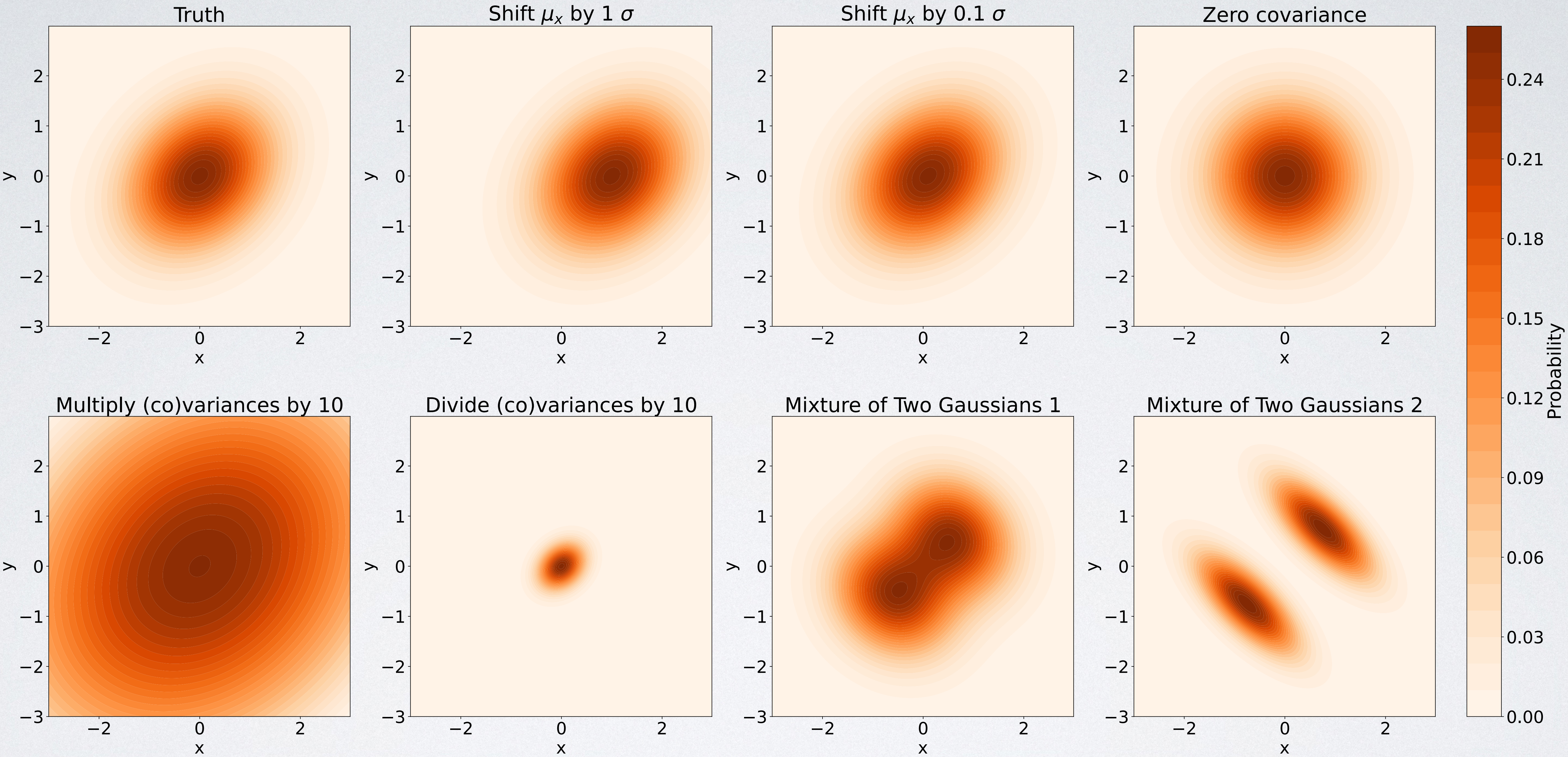
- Can be valuable to disentangle these
- Precision & Recall (Kynkäänniemi et al 2019)
  - Estimate real and generated manifold using k-nearest-neighbours
  - Precision: fraction of generated samples lying within real manifold (quality)
  - Recall: fraction of real samples which lying within gen manifold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

TOY DISTRIBUTIONS



TOY DISTRIBUTIONS

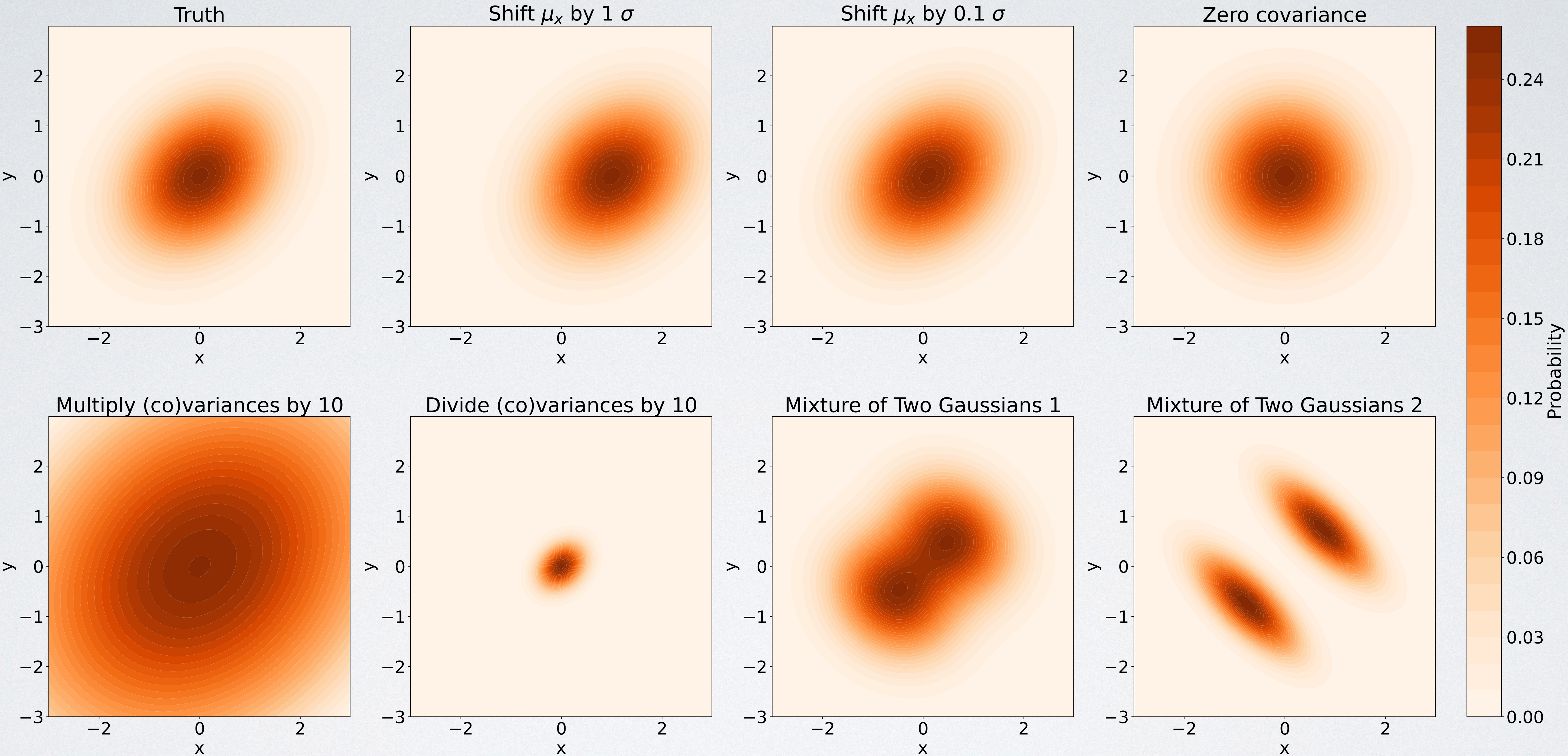
- We first test on toy Gaussian distributions



TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

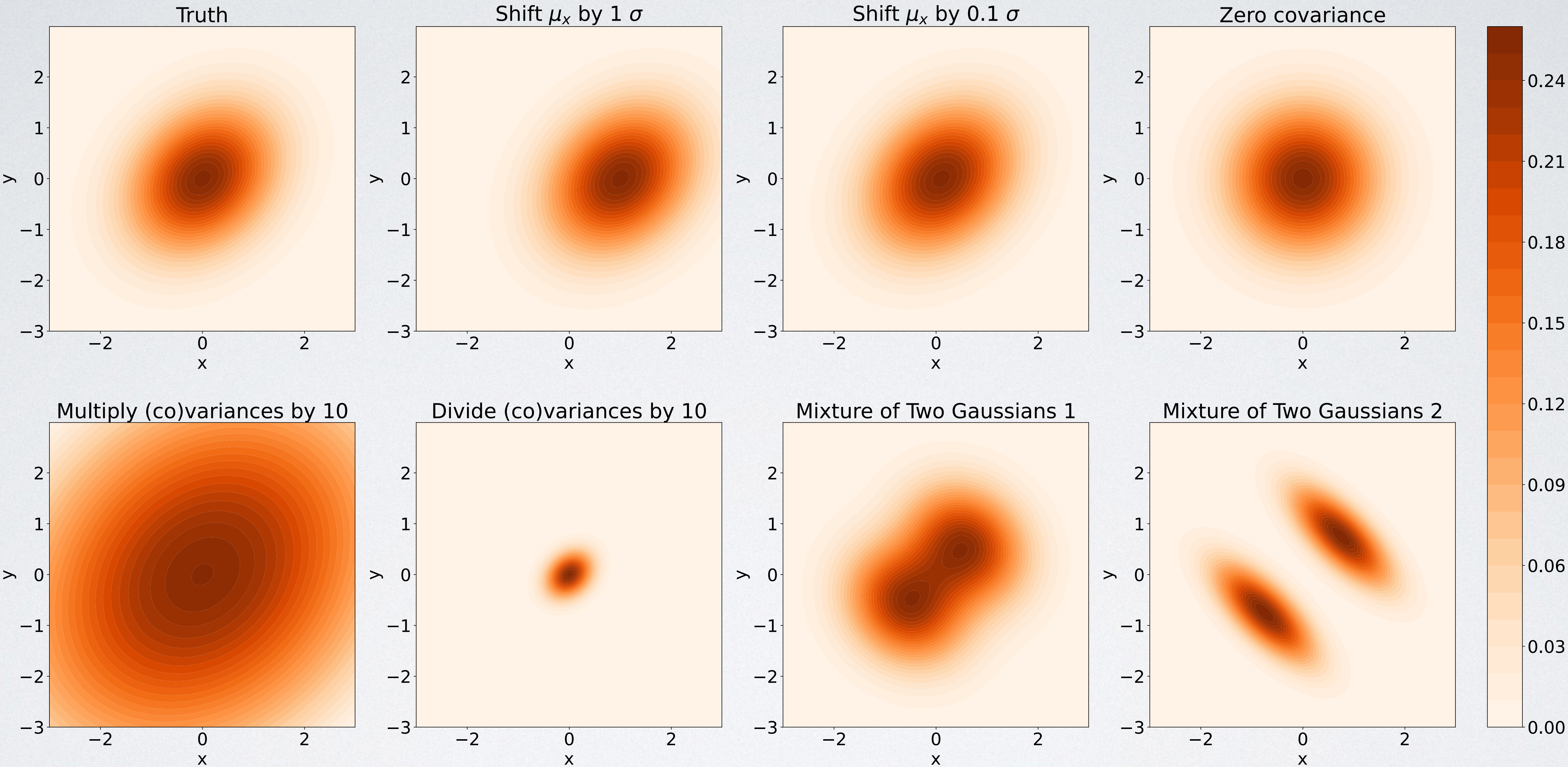
Tests if metrics are sensitive to correlations



TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

Tests if metrics are sensitive to correlations

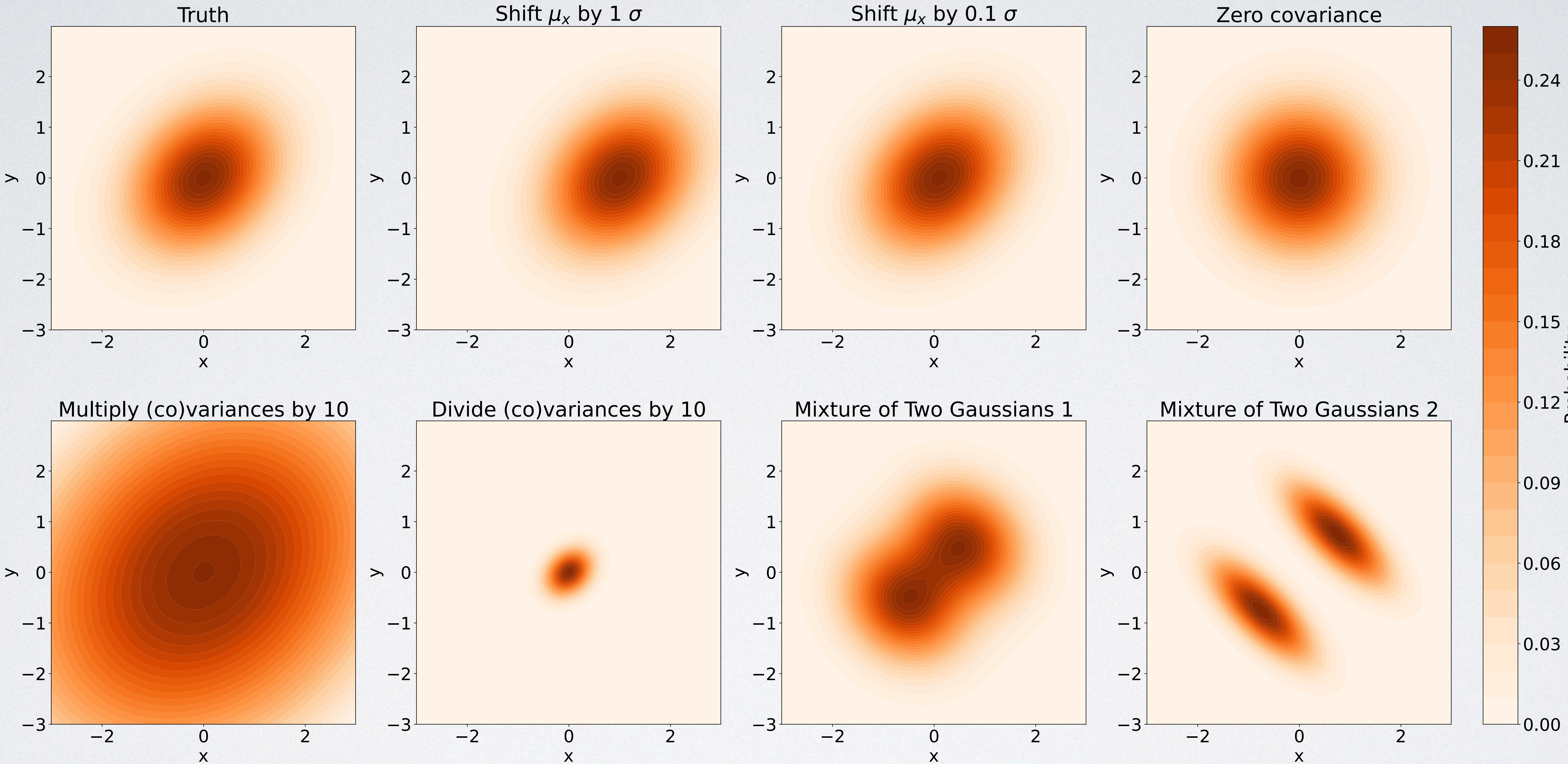


Tests sensitivity to quality

TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

Tests if metrics are sensitive to correlations



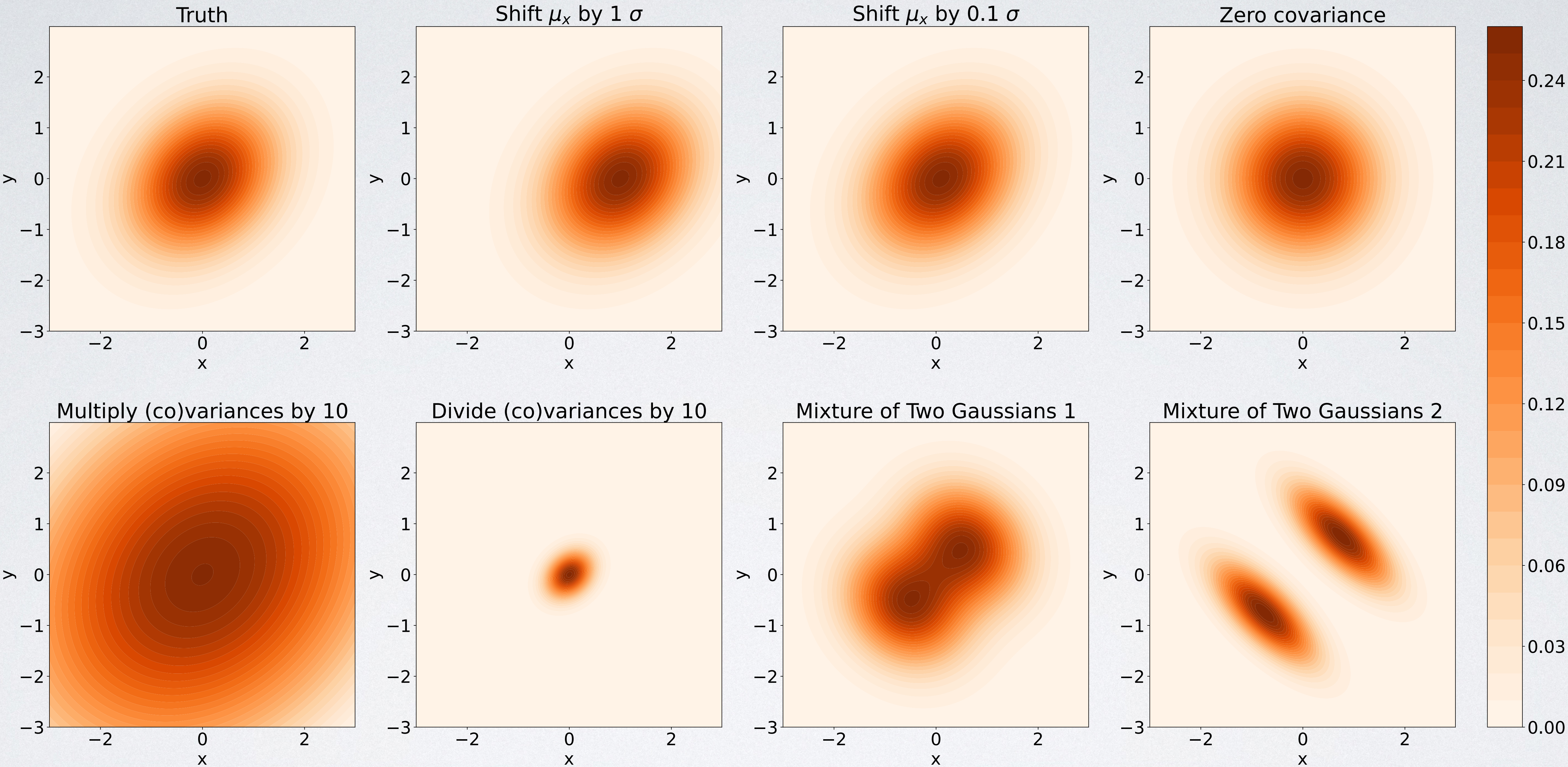
Tests sensitivity to quality

Tests sensitivity to diversity

TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

Tests if metrics are sensitive to correlations



Tests sensitivity to quality

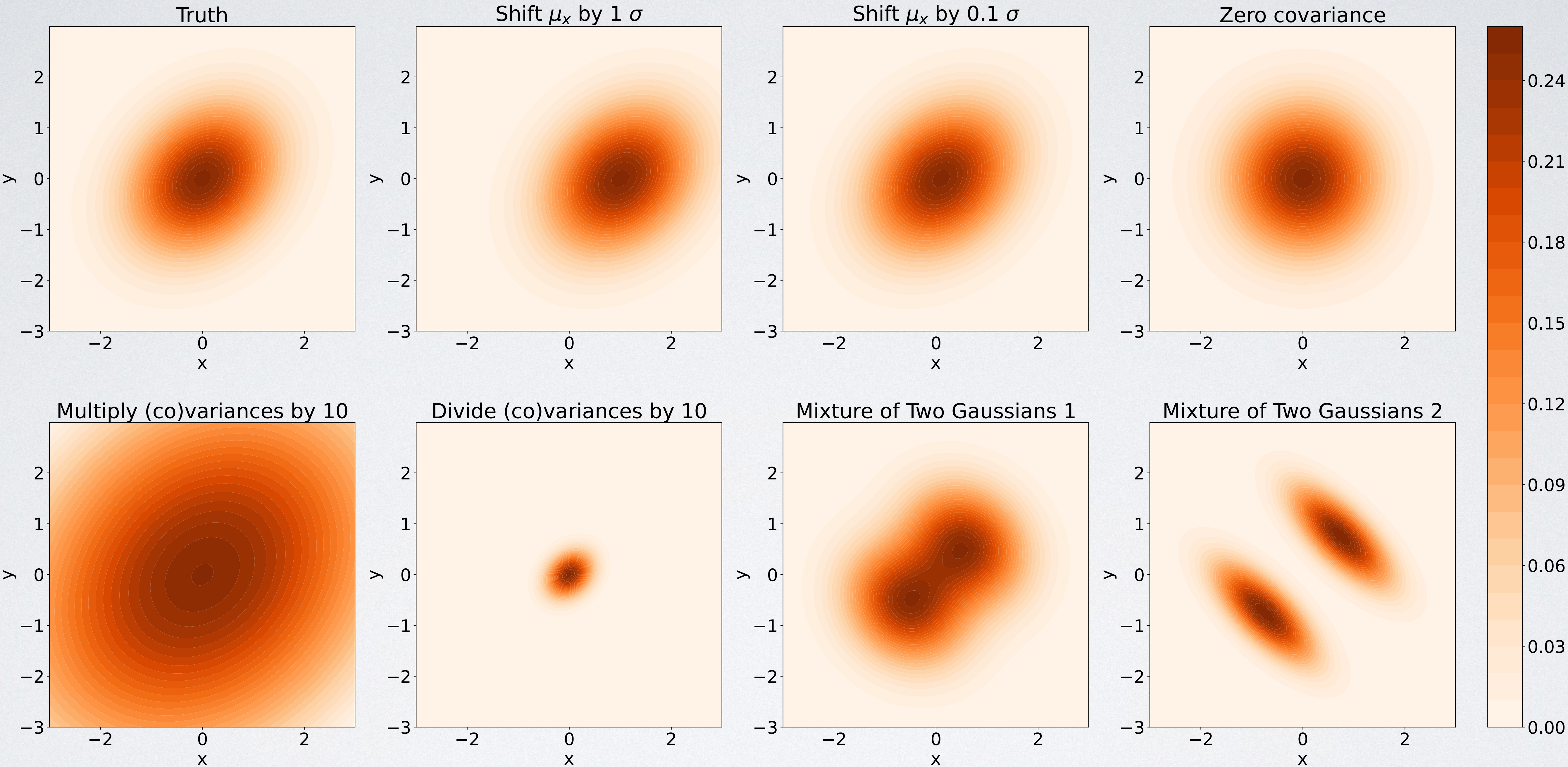
Tests sensitivity to diversity

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

TOY DISTRIBUTIONS

- We first test on toy Gaussian distributions

Tests if metrics are sensitive to correlations



Tests sensitivity to quality

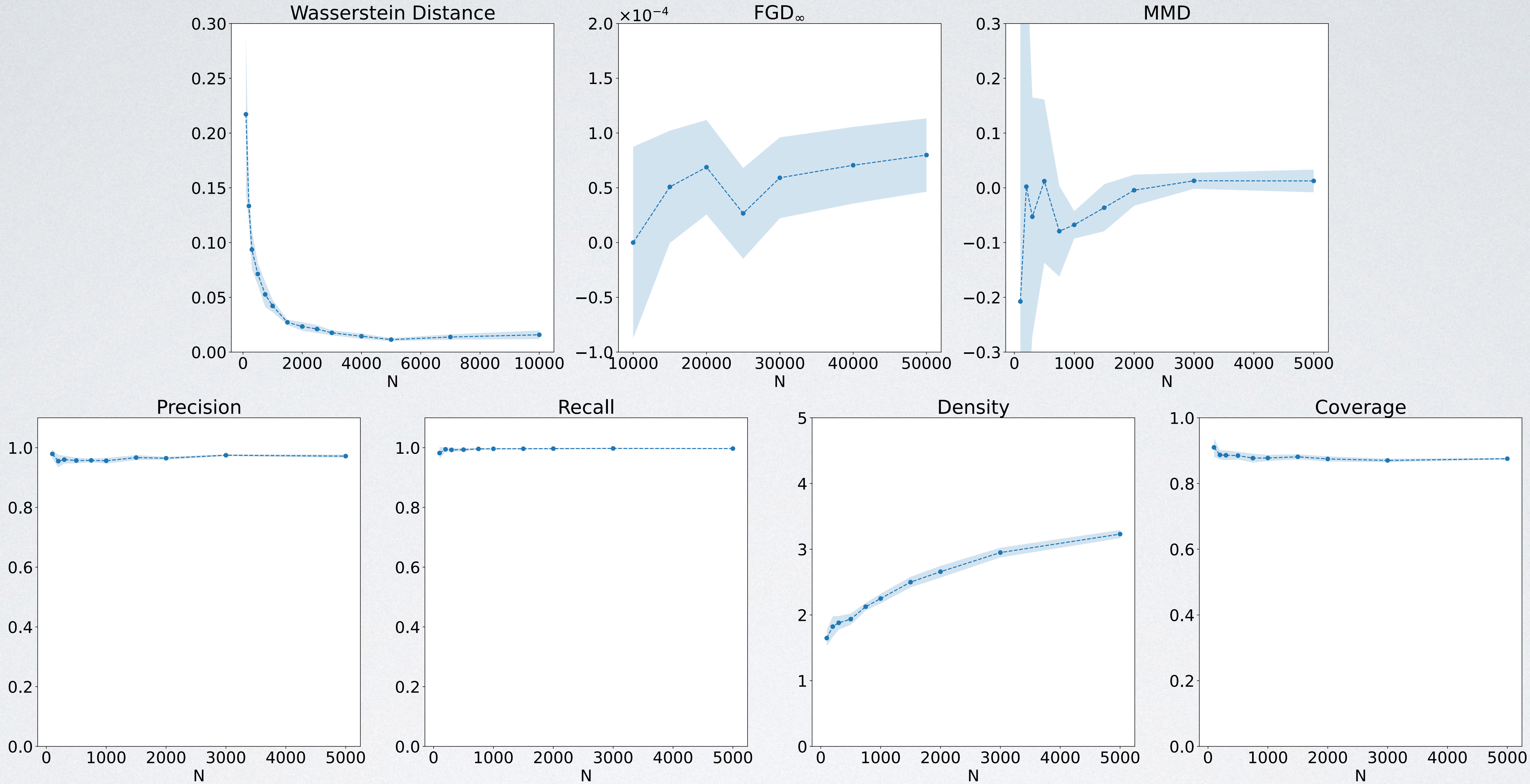
Tests sensitivity to diversity

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

Same statistics, but easier to distinguish (by eye)

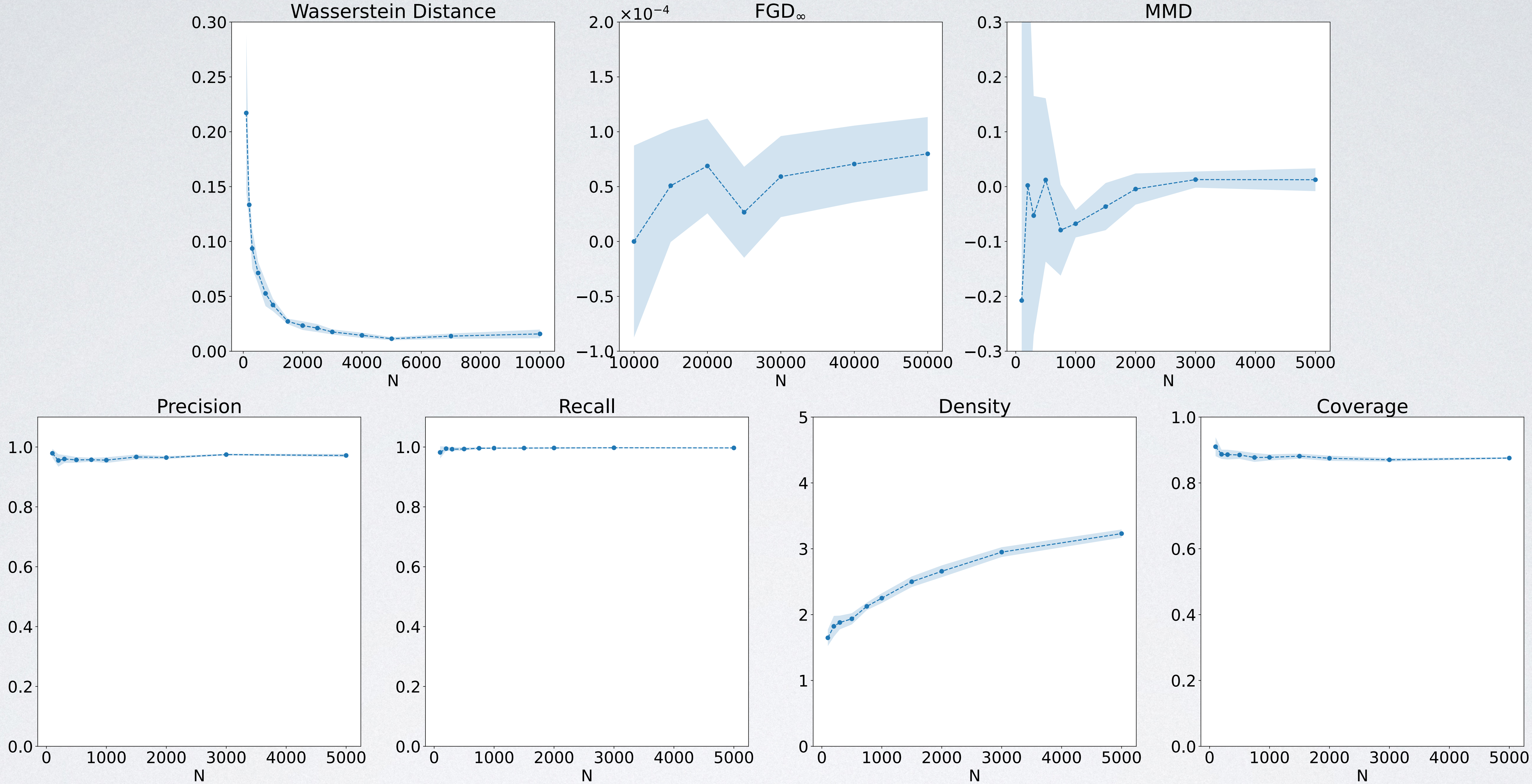
TRUTH SCORES

Scores vs sample size comparing samples of the true distribution



TRUTH SCORES

Scores vs sample size comparing samples of the true distribution



- FGD_∞ and MMD are effectively unbiased
- Wasserstein, density, and coverage very slow to converge

TRUTH SCORES

Most sensitive metric per distribution in bold

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	$\mathbf{0.036 \pm 0.003}$	$\mathbf{0.191 \pm 0.005}$
FGD_∞ $\times 10^3$	0.08 ± 0.03	$\mathbf{1011 \pm 1}$	$\mathbf{11.0 \pm 0.1}$	$\mathbf{32.3 \pm 0.2}$	9400 ± 8	$\mathbf{935.1 \pm 0.7}$	0.07 ± 0.03	0.03 ± 0.03
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
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	0.9976 ± 0.0007
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.9976 ± 0.0006	0.998 ± 0.001	0.58 ± 0.02	0.996 ± 0.001	0.9970 ± 0.0009
Density	3.23 ± 0.06	2.48 ± 0.08	3.19 ± 0.07	3.1 ± 0.1	0.60 ± 0.02	5.7 ± 0.3	2.99 ± 0.09	0.989 ± 0.009
Coverage	0.876 ± 0.002	0.780 ± 0.006	0.872 ± 0.005	0.872 ± 0.004	0.60 ± 0.01	0.406 ± 0.008	0.871 ± 0.002	0.956 ± 0.006

TRUTH SCORES

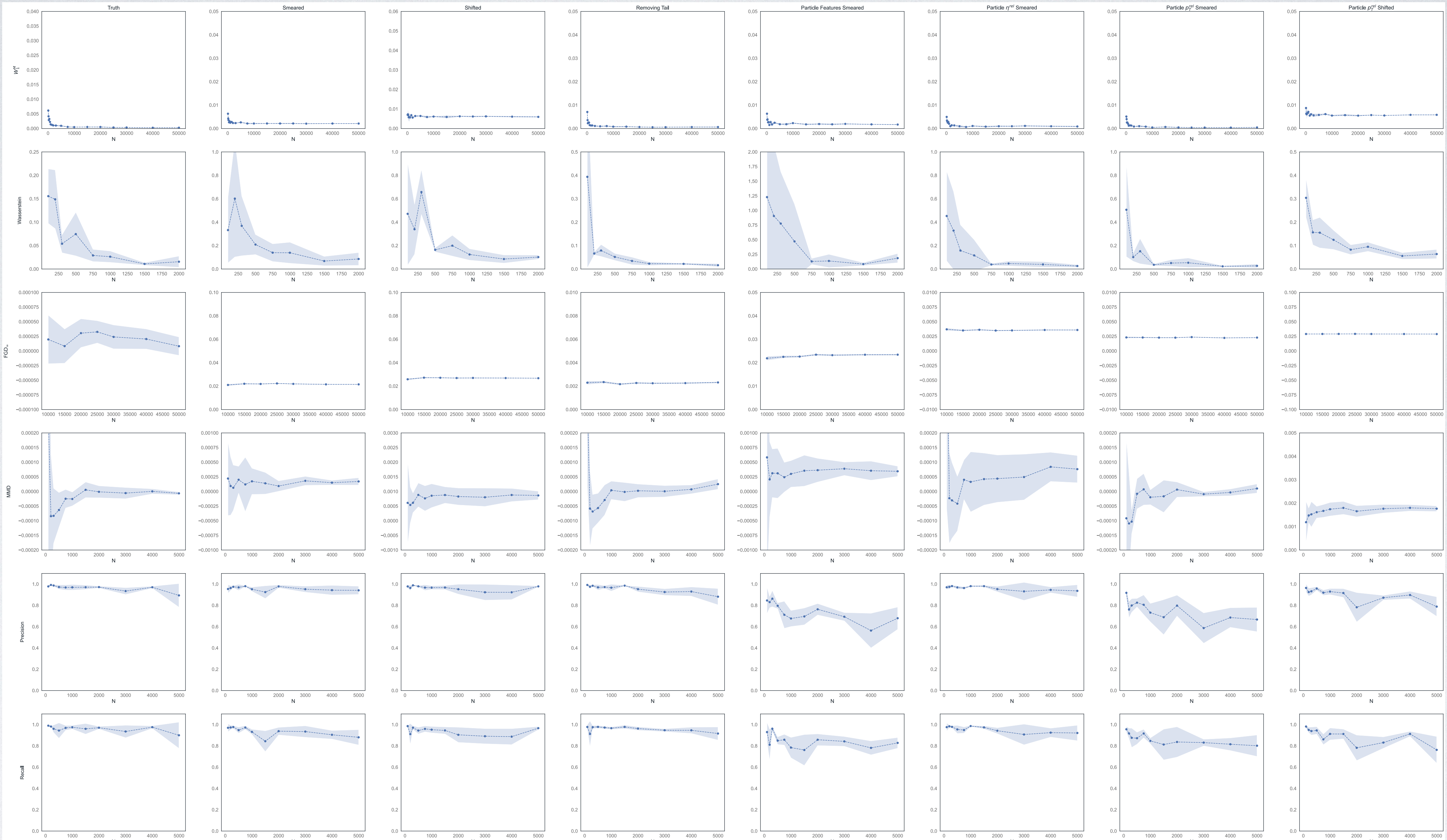
Most sensitive metric per distribution in bold

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	$\mathbf{0.036 \pm 0.003}$	$\mathbf{0.191 \pm 0.005}$
$\mathrm{FGD}_\infty \times 10^3$	0.08 ± 0.03	$\mathbf{1011 \pm 1}$	$\mathbf{11.0 \pm 0.1}$	$\mathbf{32.3 \pm 0.2}$	9400 ± 8	$\mathbf{935.1 \pm 0.7}$	0.07 ± 0.03	0.03 ± 0.03
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
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	0.9976 ± 0.0007
Recall	0.997 ± 0.001	0.992 ± 0.003	0.997 ± 0.001	0.9976 ± 0.0006	0.998 ± 0.001	0.58 ± 0.02	0.996 ± 0.001	0.9970 ± 0.0009
Density	3.23 ± 0.06	2.48 ± 0.08	3.19 ± 0.07	3.1 ± 0.1	0.60 ± 0.02	5.7 ± 0.3	2.99 ± 0.09	0.989 ± 0.009
Coverage	0.876 ± 0.002	0.780 ± 0.006	0.872 ± 0.005	0.872 ± 0.004	0.60 ± 0.01	0.406 ± 0.008	0.871 ± 0.002	0.956 ± 0.006

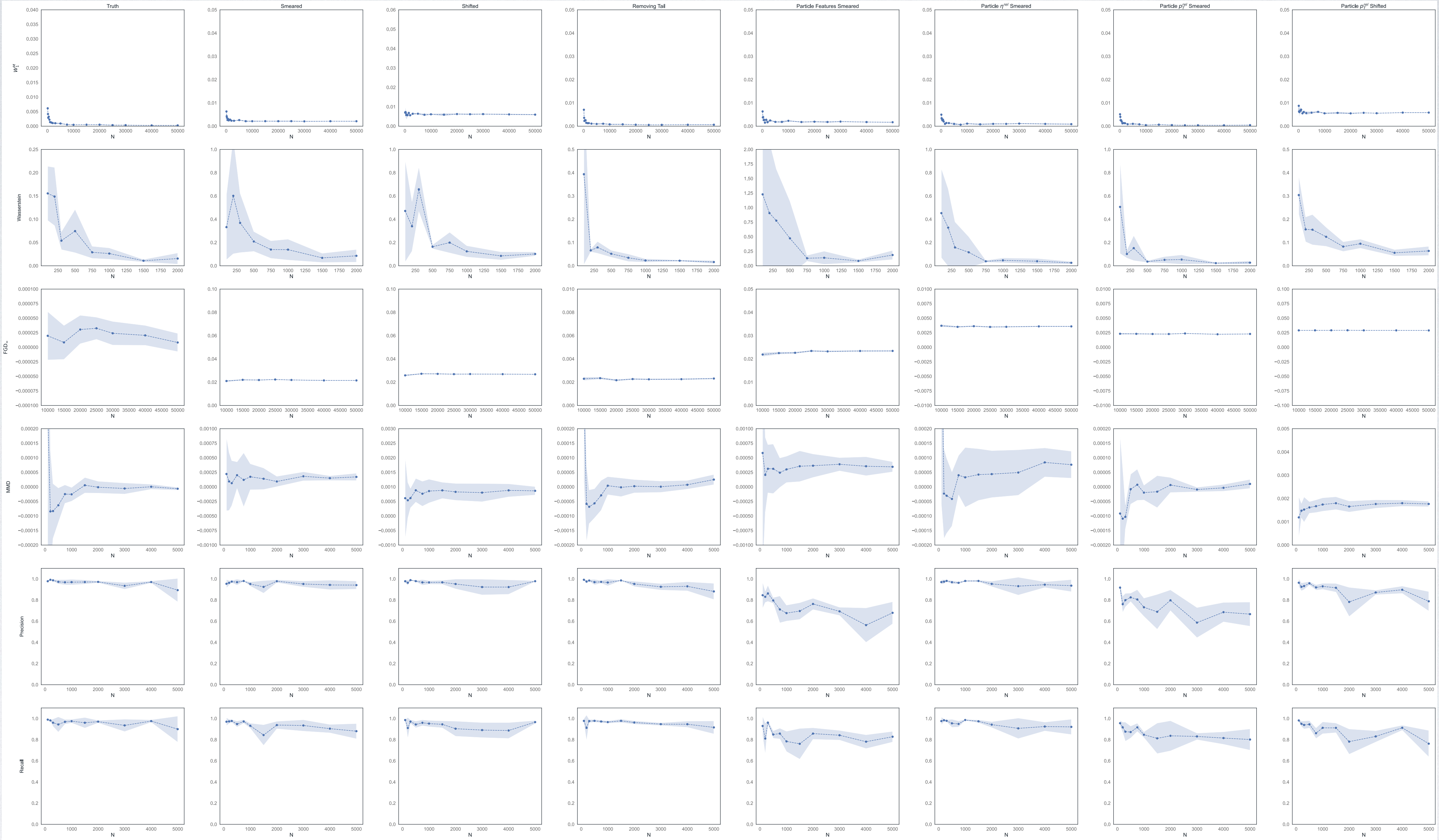
- Wasserstein, $\mathbf{FGD}_\infty$, MMD find all alternatives discrepant, except $\mathbf{FGD}_\infty$ on mixtures
- $\mathbf{FGD}_\infty$ generally the most sensitive otherwise, but misses shape distortions
- Precision and recall do their job, density and coverage give unintuitive results

EFP SCORES VS SAMPLE SIZE

EFP SCORES VS SAMPLE SIZE

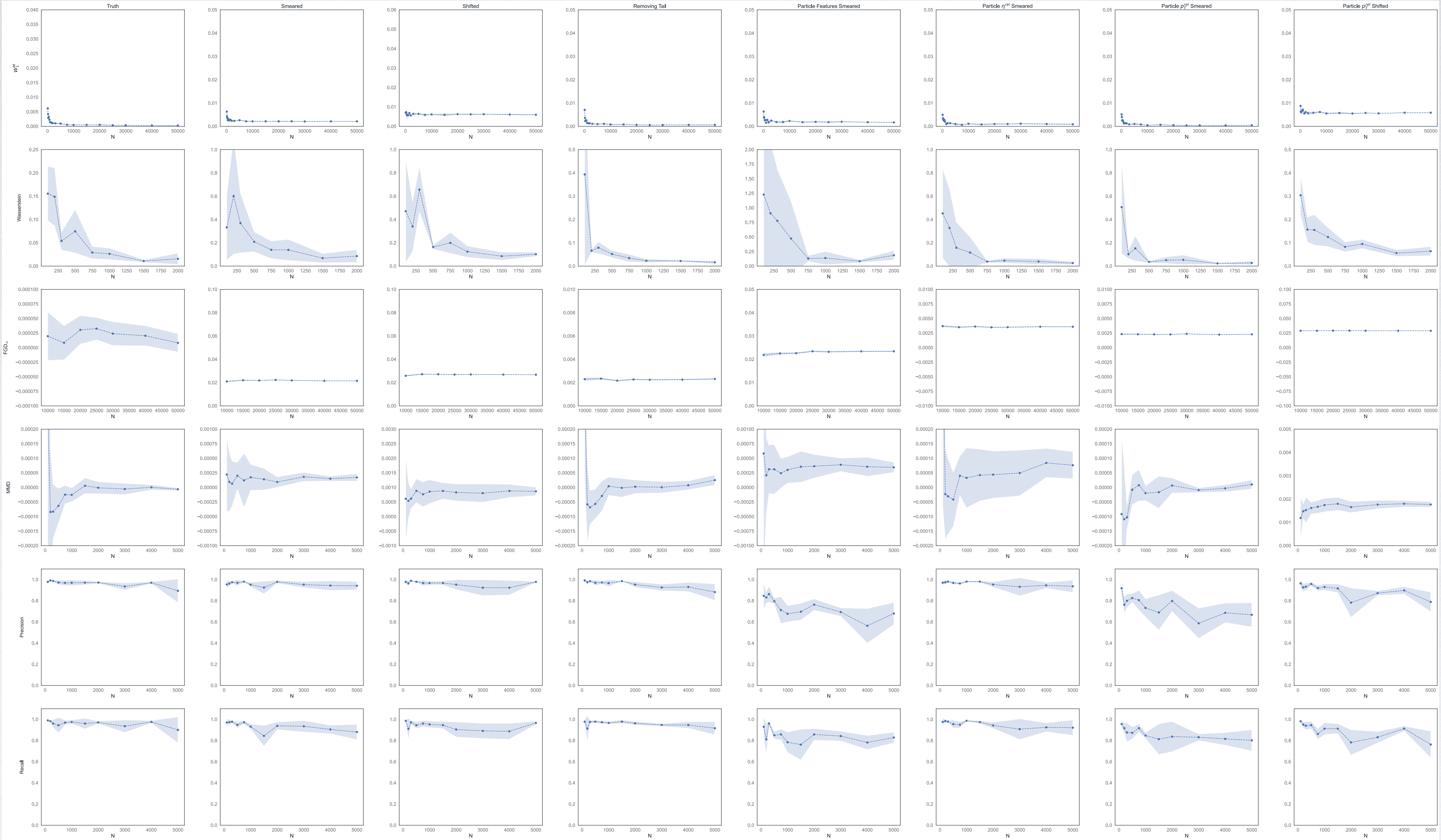


EFP SCORES VS SAMPLE SIZE



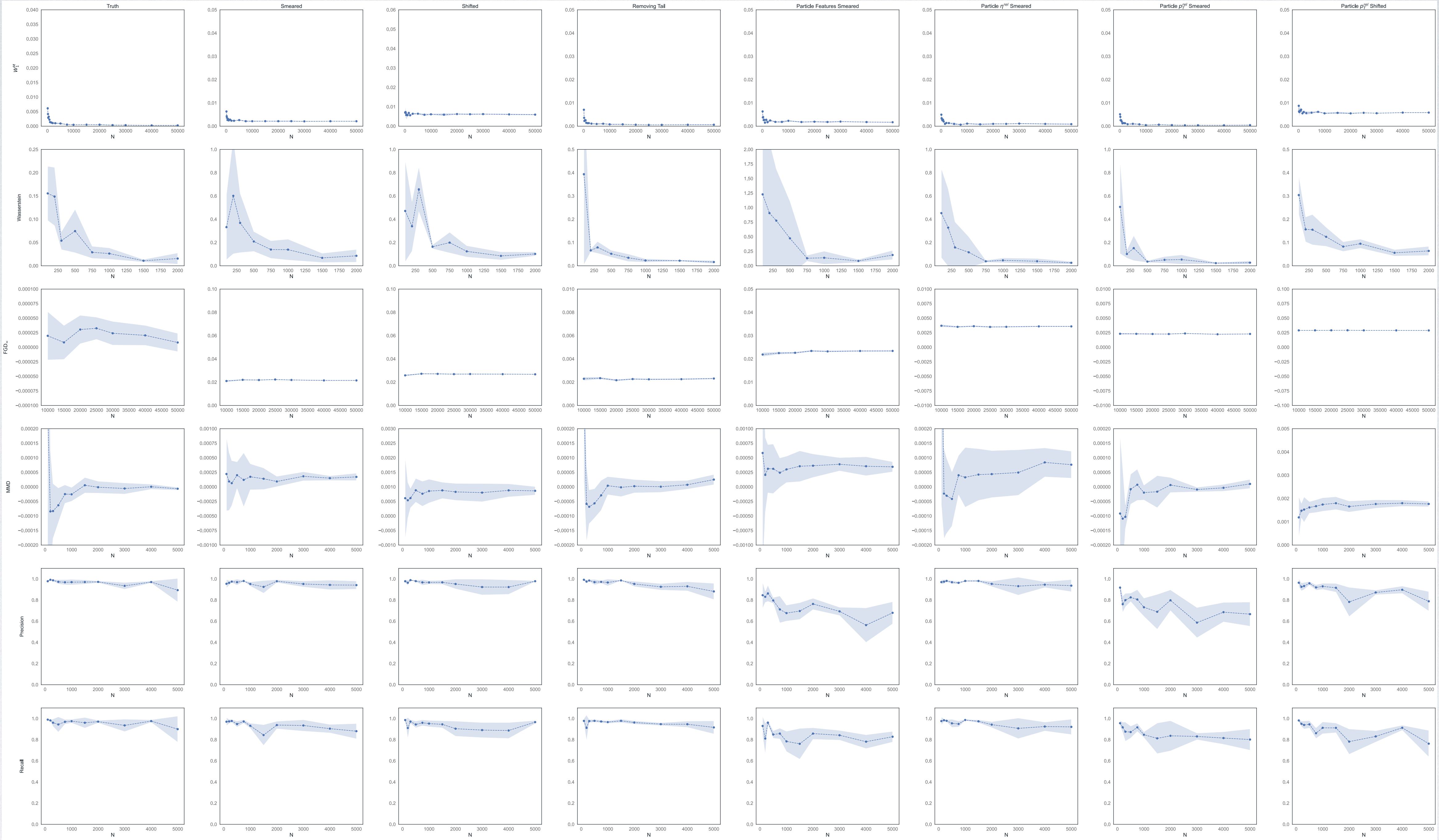
- W_1^M (looking at 1D mass distribution only) works somewhat, but not as sensitive

EFP SCORES VS SAMPLE SIZE



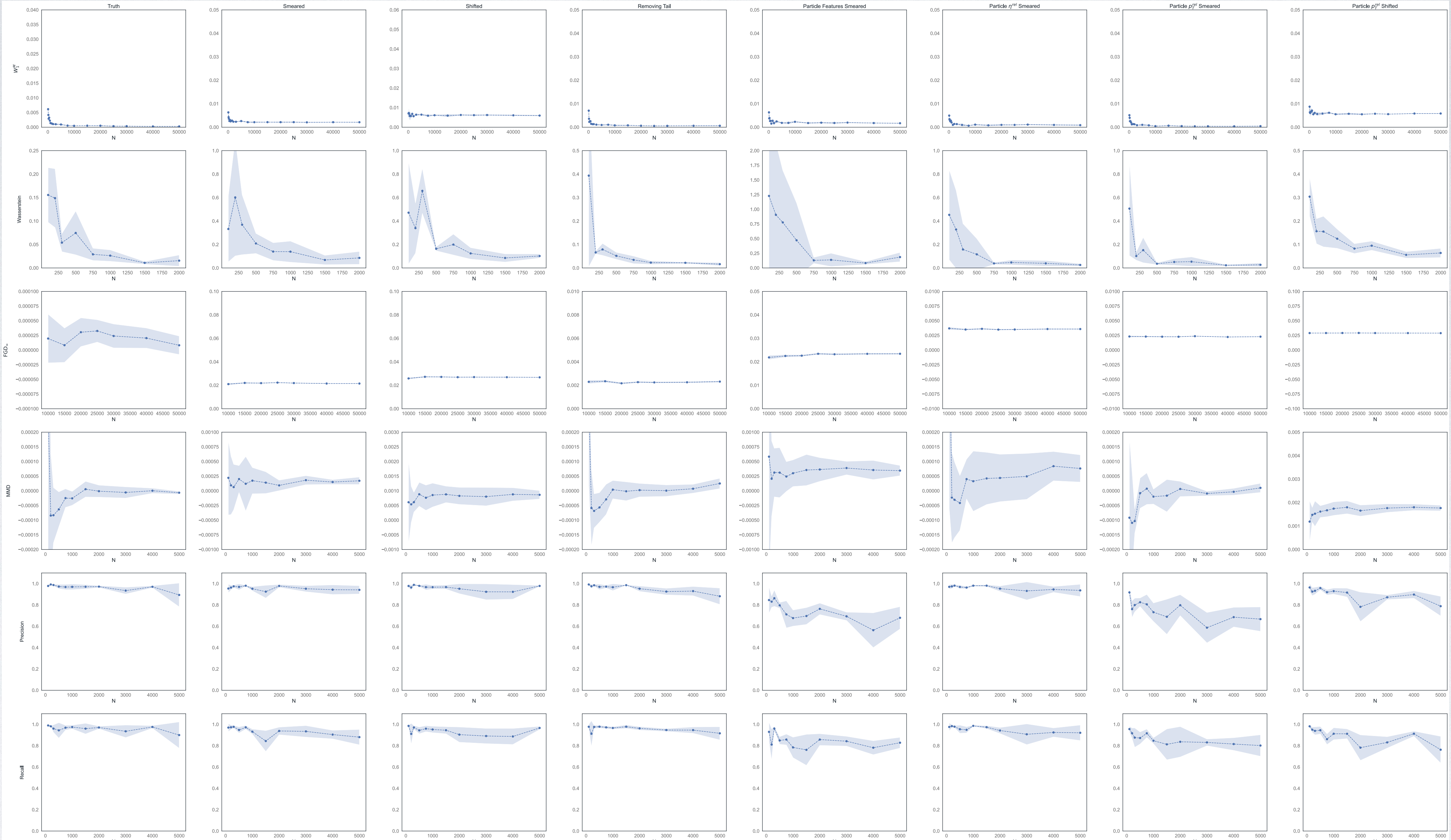
- W_1^M (looking at 1D mass distribution only) works somewhat, but not as sensitive
- Wasserstein distance is biased and slow to converge

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

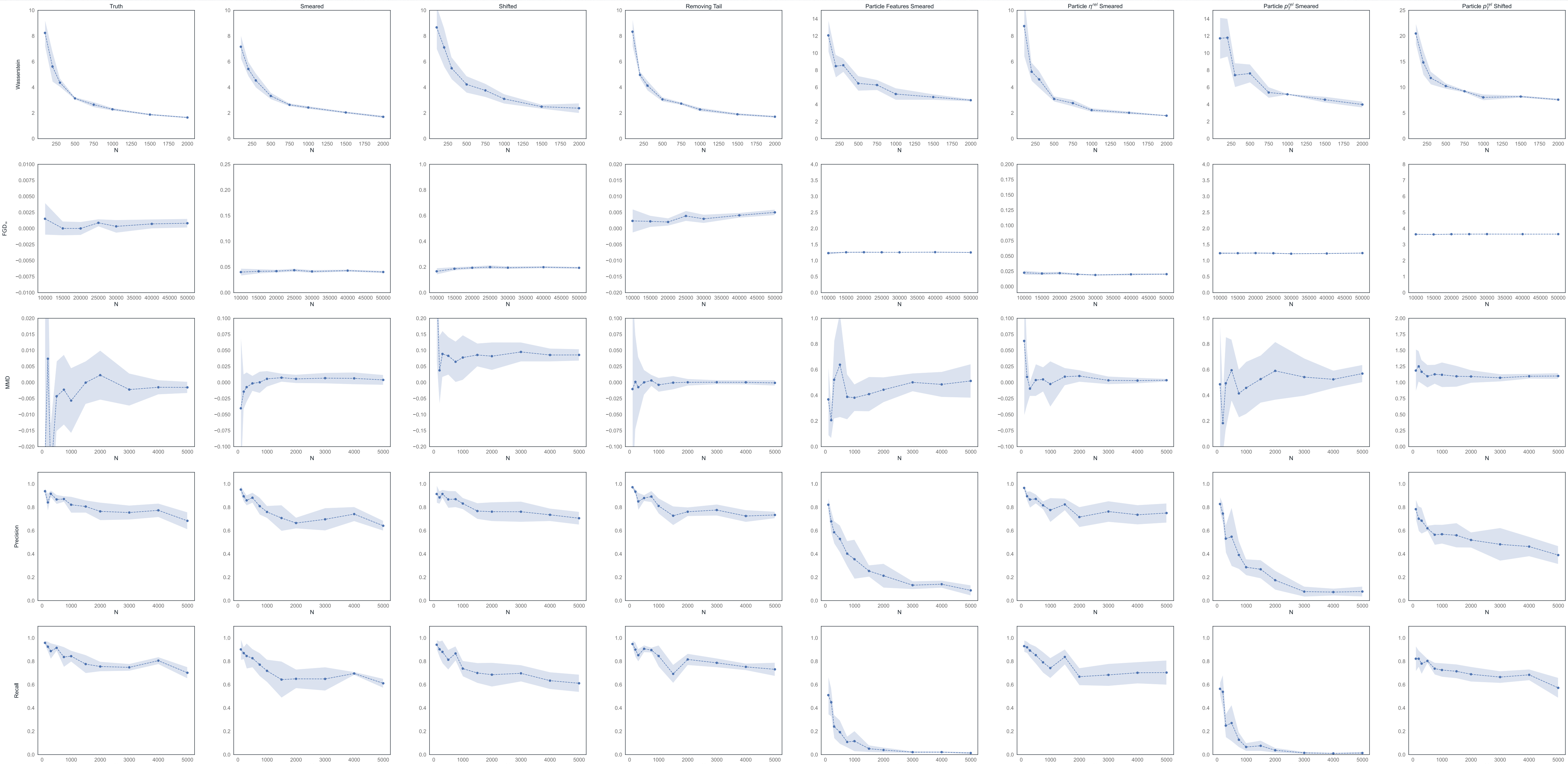
EFP SCORES VS SAMPLE SIZE



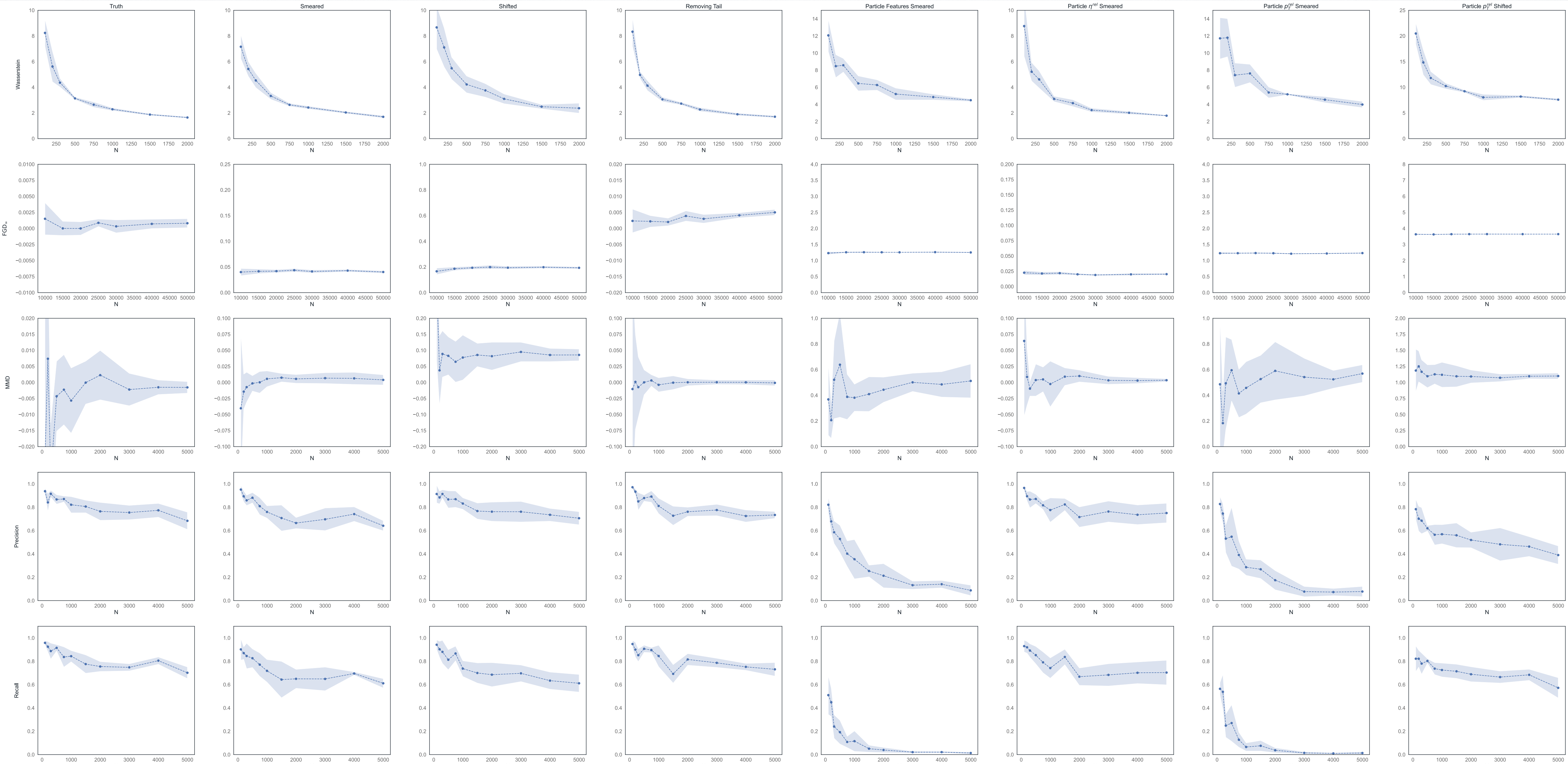
- W_1^M (looking at 1D mass distribution only) works somewhat, but not as sensitive
- Wasserstein distance is biased and slow to converge
- Precision, recall work roughly - useful for diagnosing failure modes but not for comparing

- FGD is the most sensitive
- MMD reasonable

PARTICLENET ACTIVATION SCORES



PARTICLENET ACTIVATION SCORES



- Same conclusions overall as for EFPs
- FGD the best, MMD reasonable, P&R are OK for diagnosing failure modes

RESULTS: GLUON JETS

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

RESULTS: GLUON JETS

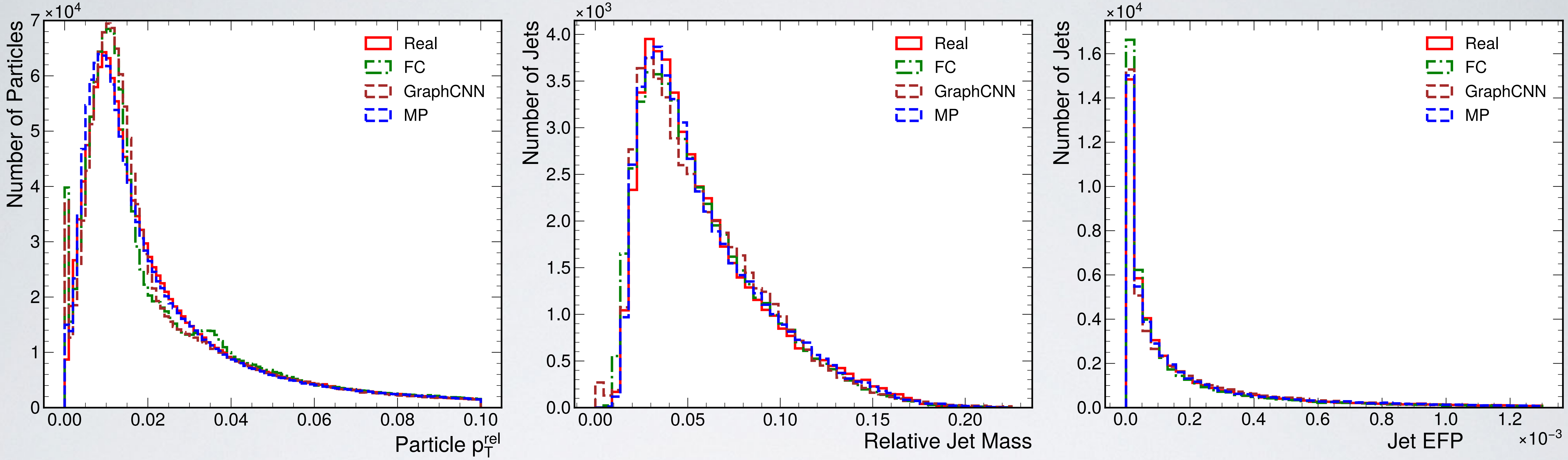
Kansal et al., ML4PS @ NeurIPS 2020
Kansal et al., NeurIPS 2021

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

RESULTS: GLUON JETS

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

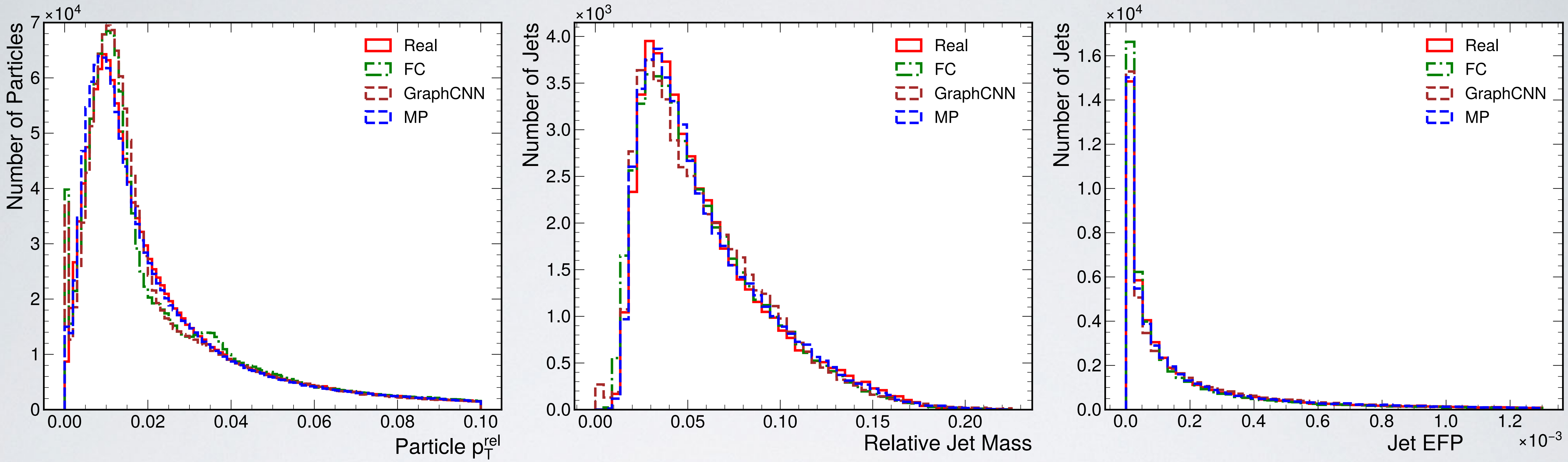
Sample feature distributions, with MPGAN compared to baseline point cloud generators



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

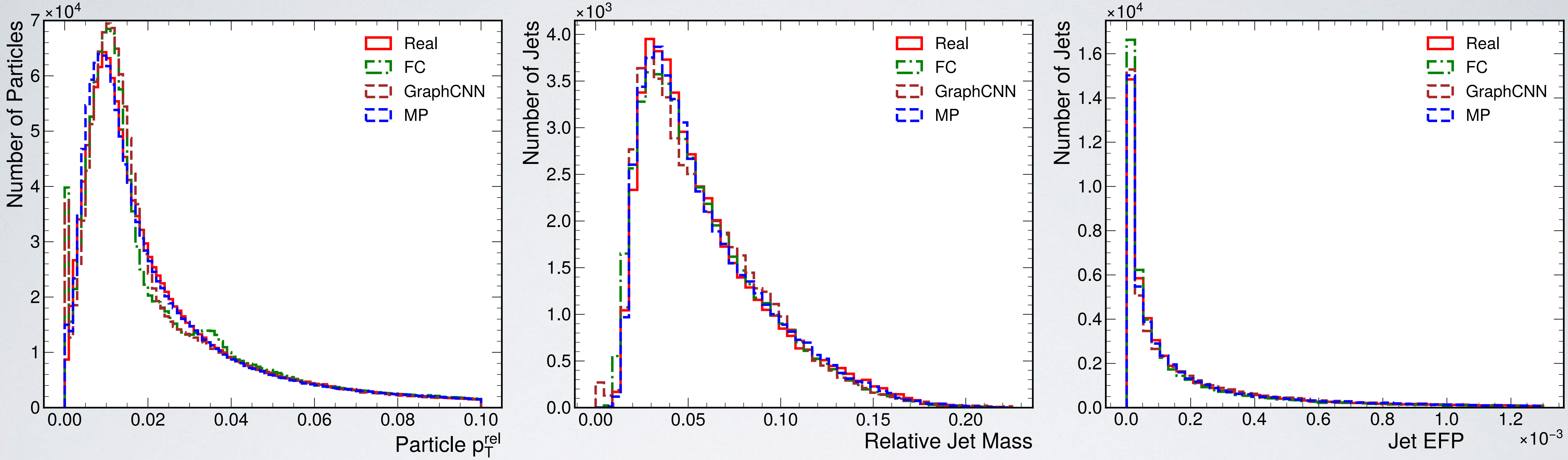


Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.3 ± 0.2	1.3 ± 0.4	1.5 ± 0.9	5.0
GraphCNN	PointNet	16 ± 6	1.9 ± 0.2	200 ± 1000	7k
MP	MP	0.9 ± 0.3	0.7 ± 0.2	0.7 ± 0.2	0.12
MP	PointNet	1.2 ± 0.4	1.3 ± 0.4	4 ± 2	18

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



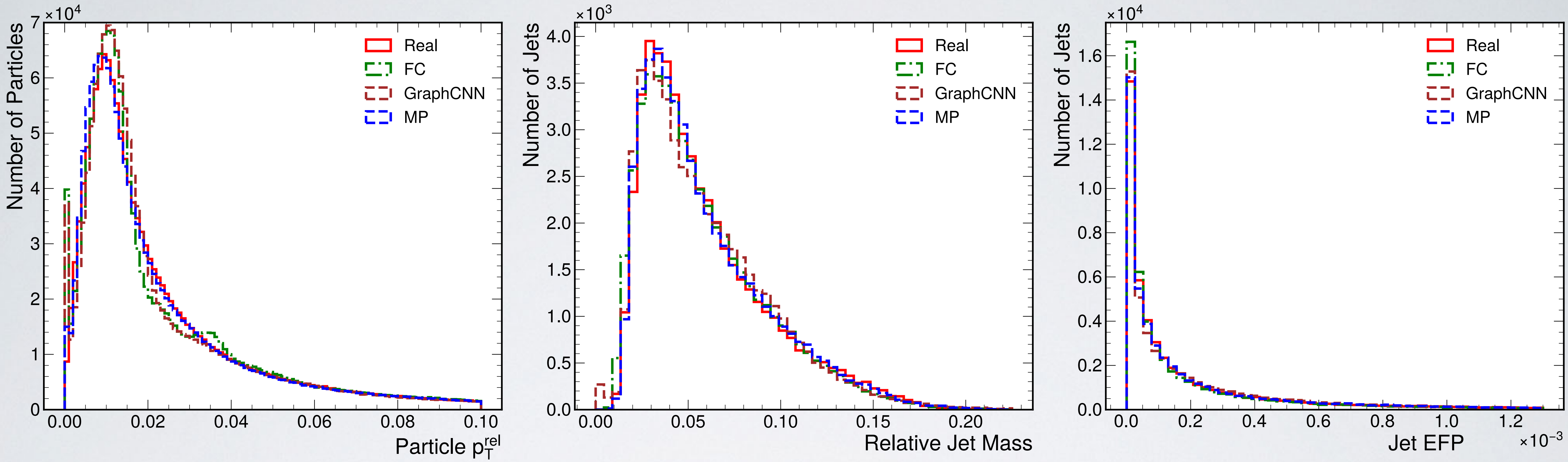
Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.3 ± 0.2	1.3 ± 0.4	1.5 ± 0.9	5.0
GraphCNN	PointNet	16 ± 6	1.9 ± 0.2	200 ± 1000	7k
MP	MP	0.9 ± 0.3	0.7 ± 0.2	0.7 ± 0.2	0.12
MP	PointNet	1.2 ± 0.4	1.3 ± 0.4	4 ± 2	18

- MPGAN generator is the best performing on every metric

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



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

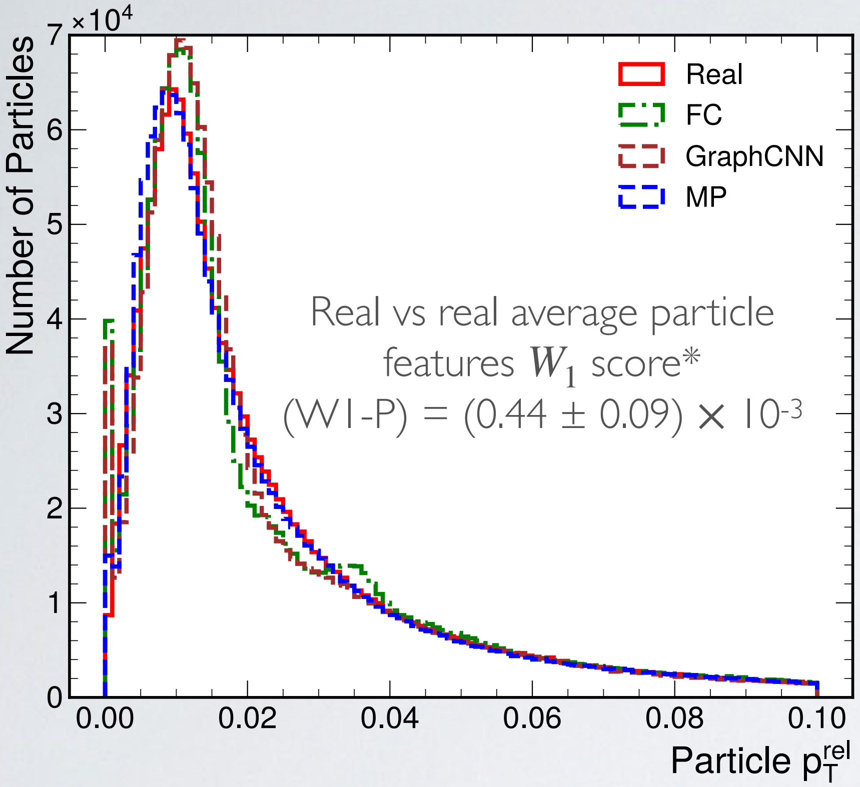
- MPGAN generator is the best performing on every metric
- Significantly outperforms alternatives on high level feature metrics (W1-M, W1-EFP, FPND)

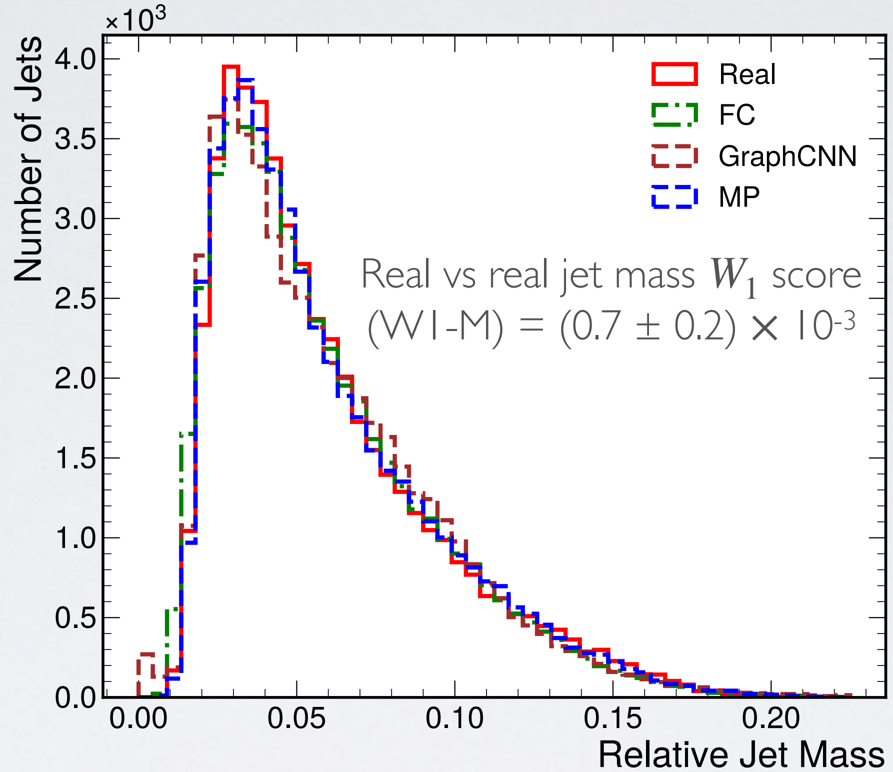
RESULTS: GLUON JETS

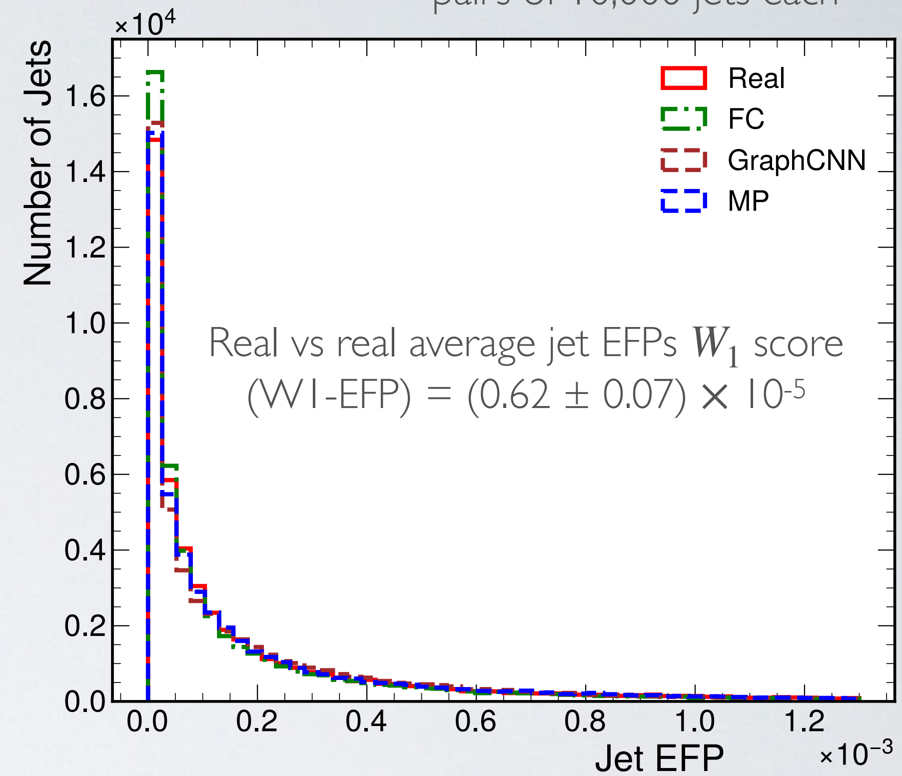
Kansal et al., ML4PS @ NeurIPS 2020
Kansal et al., NeurIPS 2021

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

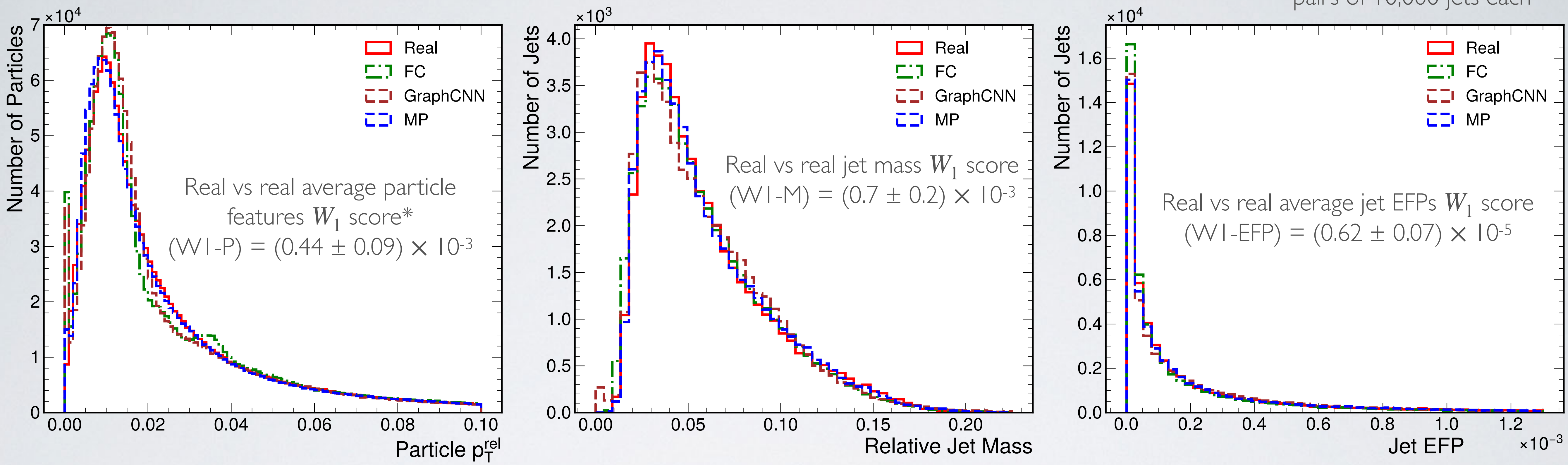
- MPGAN generator is the best performing on every metric
- Significantly outperforms alternatives on high level feature metrics (W1-M, W1-EFP, FPND)

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

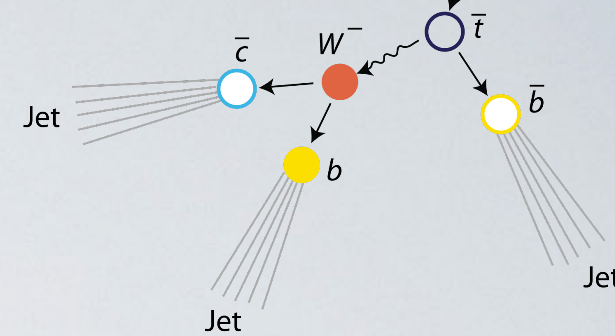
*Mean and error over 5 sets of pairs of 10,000 jets each

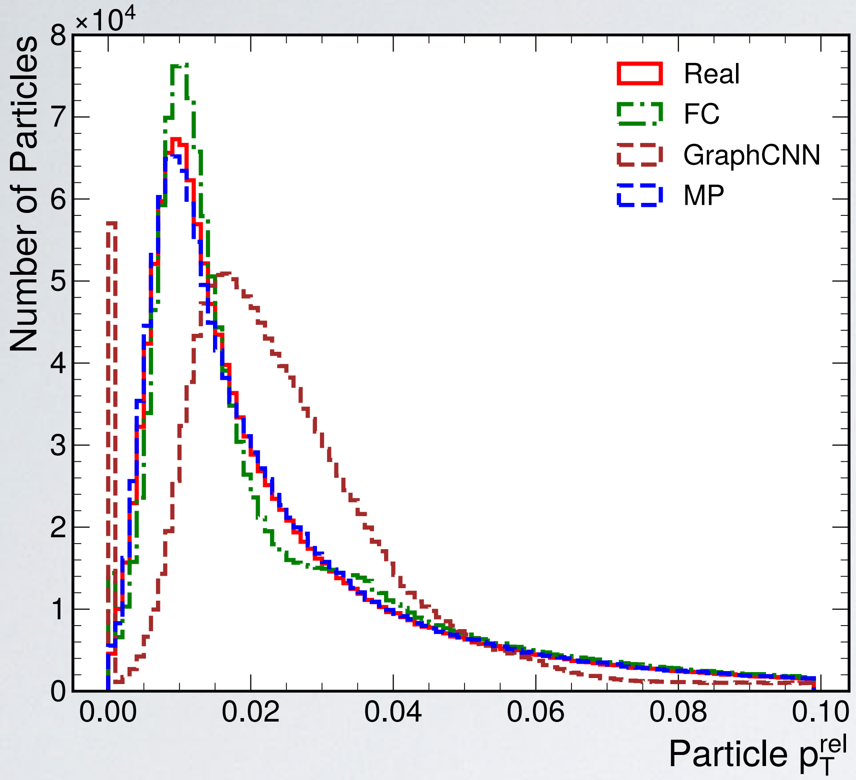


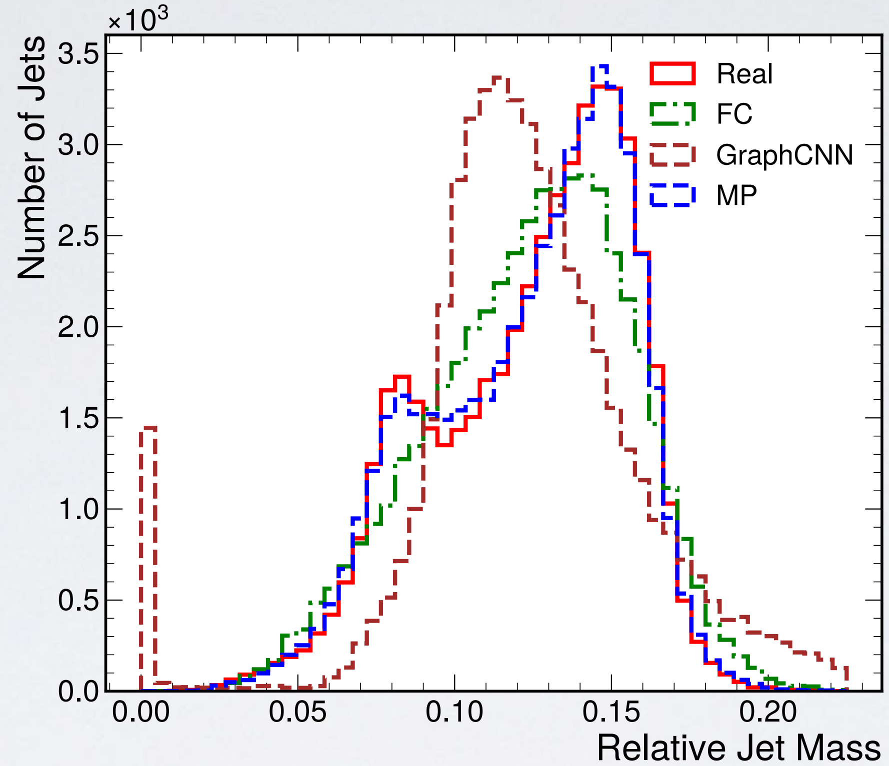
Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.3 ± 0.2	1.3 ± 0.4	1.5 ± 0.9	5.0
GraphCNN	PointNet	16 ± 6	1.9 ± 0.2	200 ± 1000	7k
MP	MP	0.9 ± 0.3	0.7 ± 0.2	0.7 ± 0.2	0.12
MP	PointNet	1.2 ± 0.4	1.3 ± 0.4	4 ± 2	18

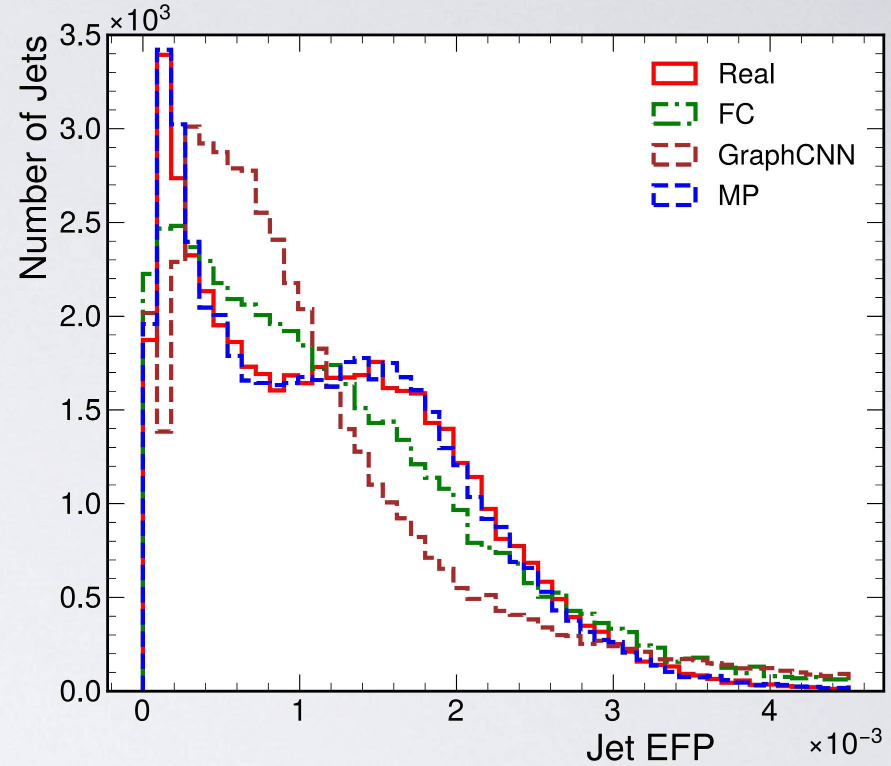
- MPGAN generator is the best performing on every metric
- Significantly outperforms alternatives on high level feature metrics (W1-M, W1-EFP, FPND)
- Mass and ave. EFP scores are within error of the real vs real baseline ⇒ learning jet substructure correctly

RESULTS: TOP QUARK JETS

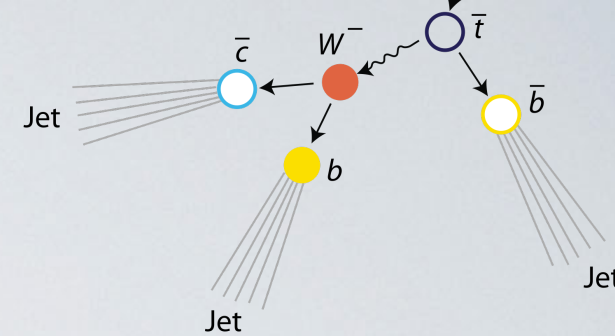


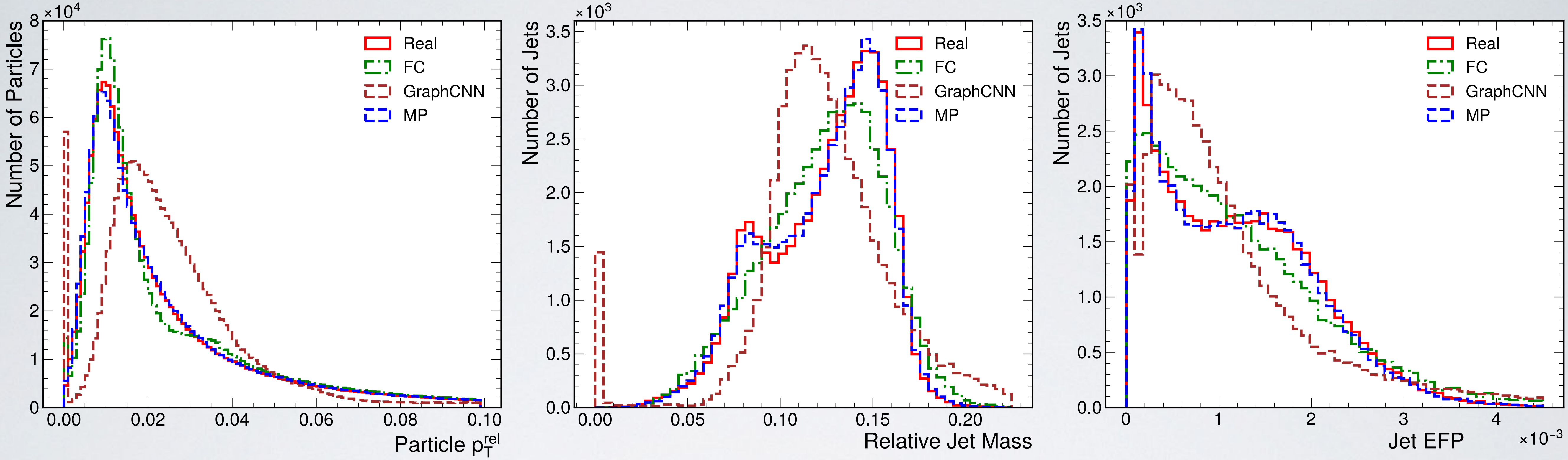






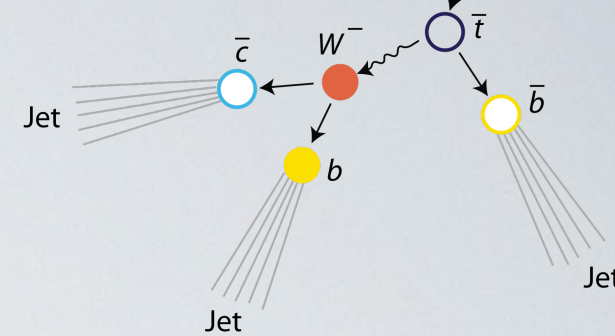
RESULTS: TOP QUARK JETS

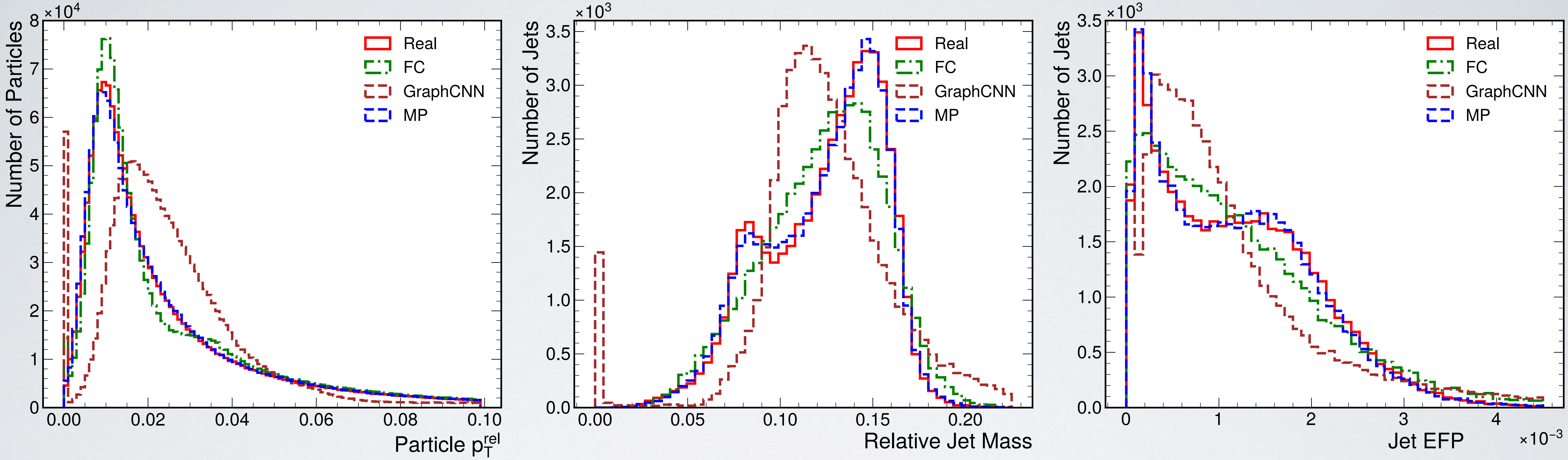




- MPGAN learns perfectly the complex bimodal jet feature distributions

RESULTS: TOP QUARK JETS

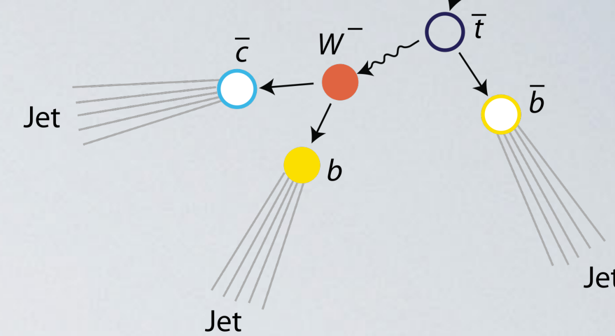


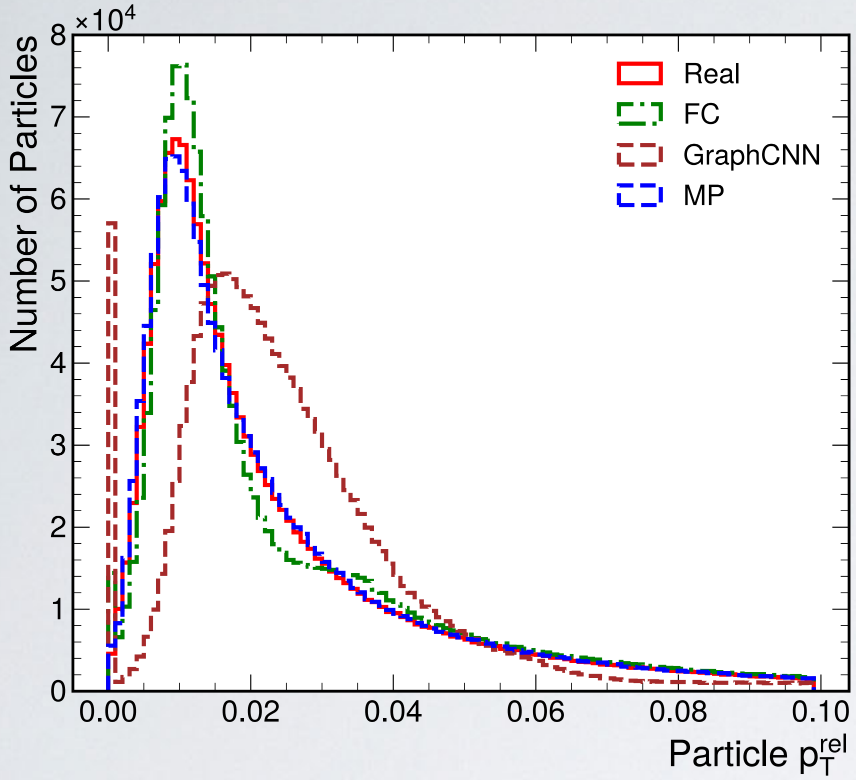


Generator	Discriminator	W1-P (10^{-3})	W1-M (10^{-3})	W1-EFP (10^{-5})	FPND
FC	PointNet	1.6 ± 0.4	2.7 ± 0.1	7.7 ± 0.5	3.9
GraphCNN	PointNet	30 ± 20	11.3 ± 0.9	37 ± 2	30k
MP	MP	2.3 ± 0.3	0.6 ± 0.2	2 ± 1	0.37
MP	PointNet	1.6 ± 0.4	0.76 ± 0.08	4 ± 1	3.7

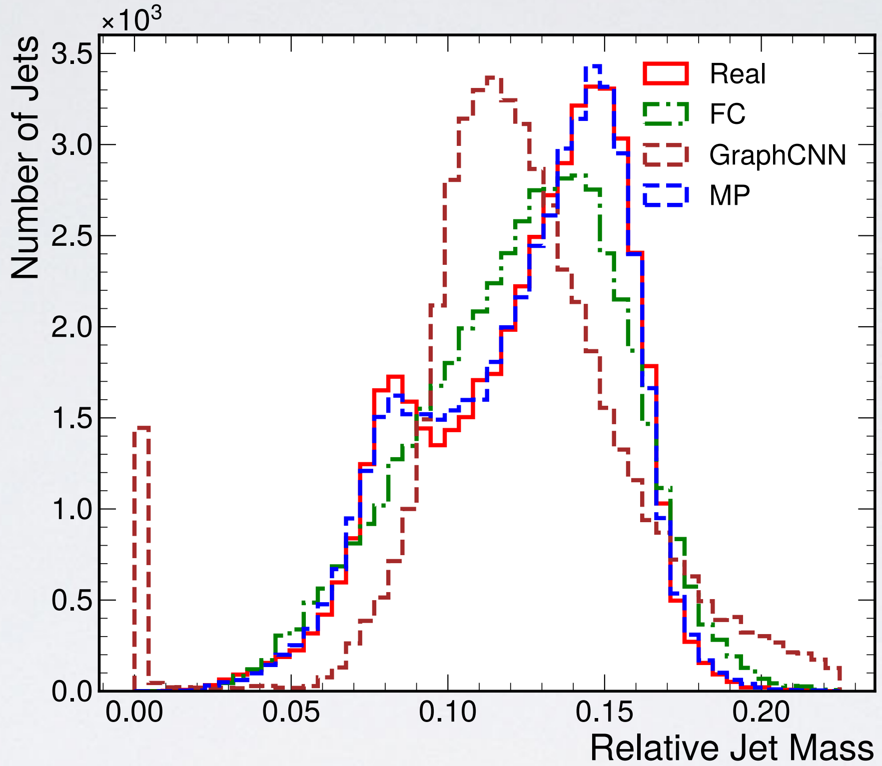
- MPGAN learns perfectly the complex bimodal jet feature distributions

RESULTS: TOP QUARK JETS

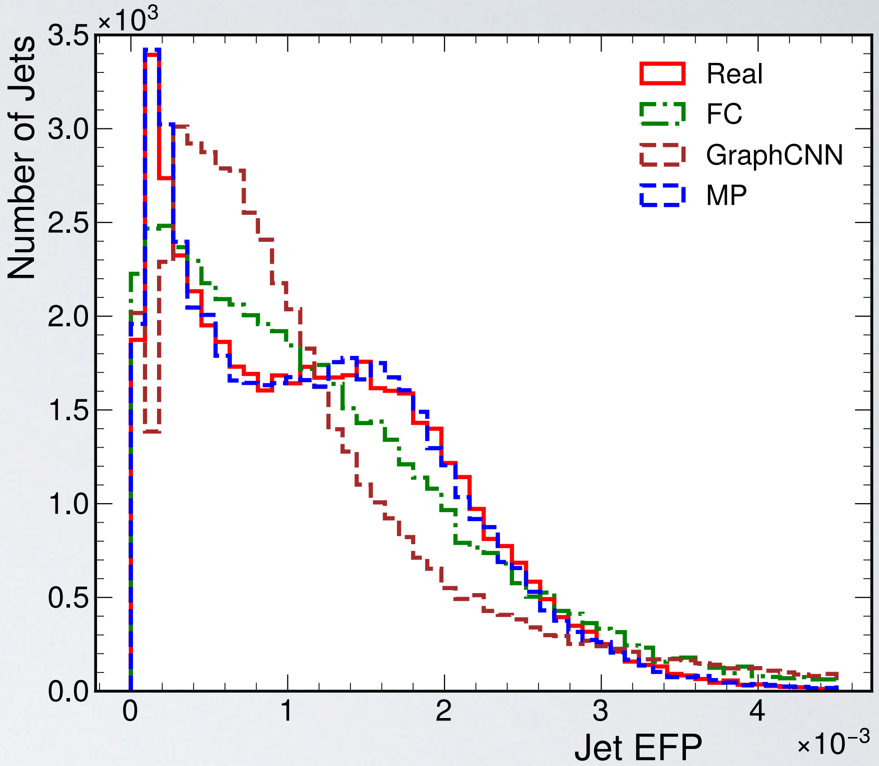




Real vs real
W1-P = (0.55 ± 0.07) × 10^{-3}



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

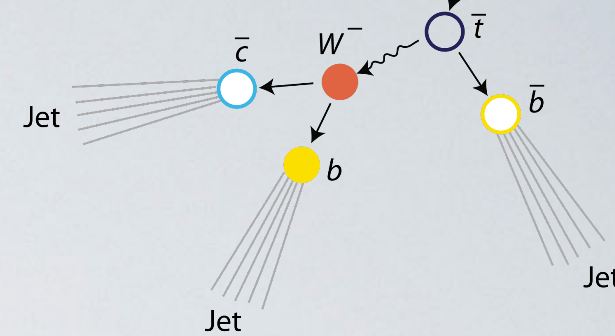


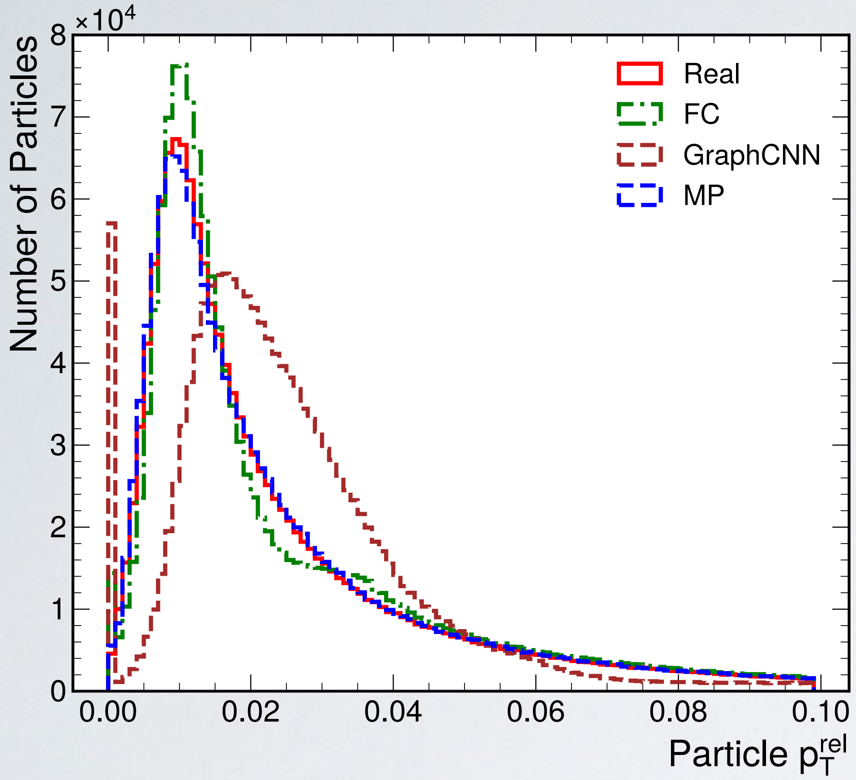
Real vs real
W1-EFP = (1.1 ± 0.1) × 10^{-5}

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

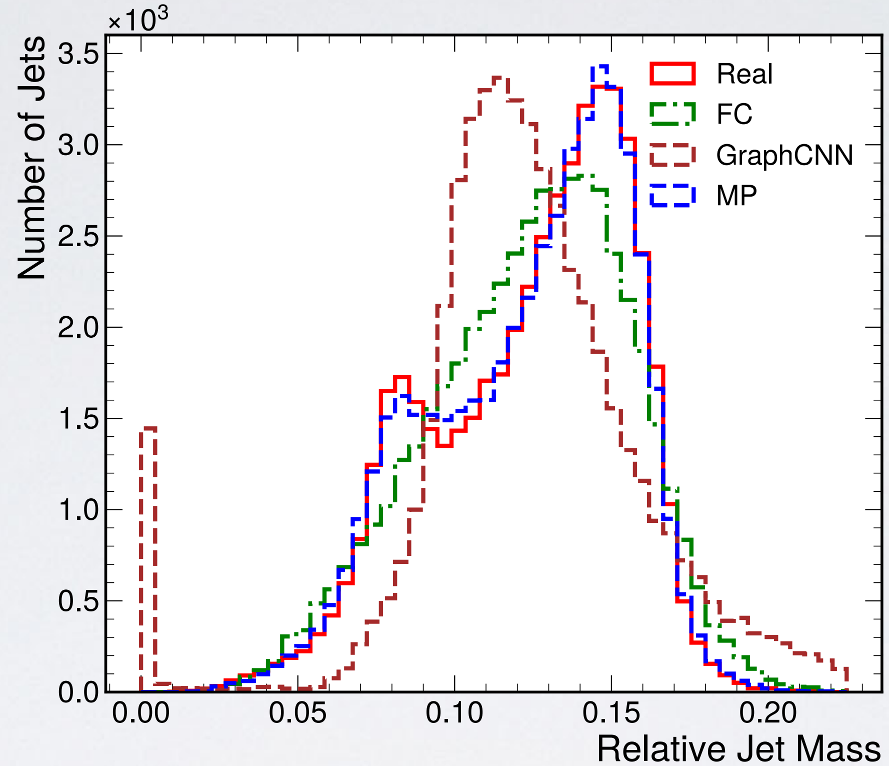
- MPGAN learns perfectly the complex bimodal jet feature distributions

RESULTS: TOP QUARK JETS

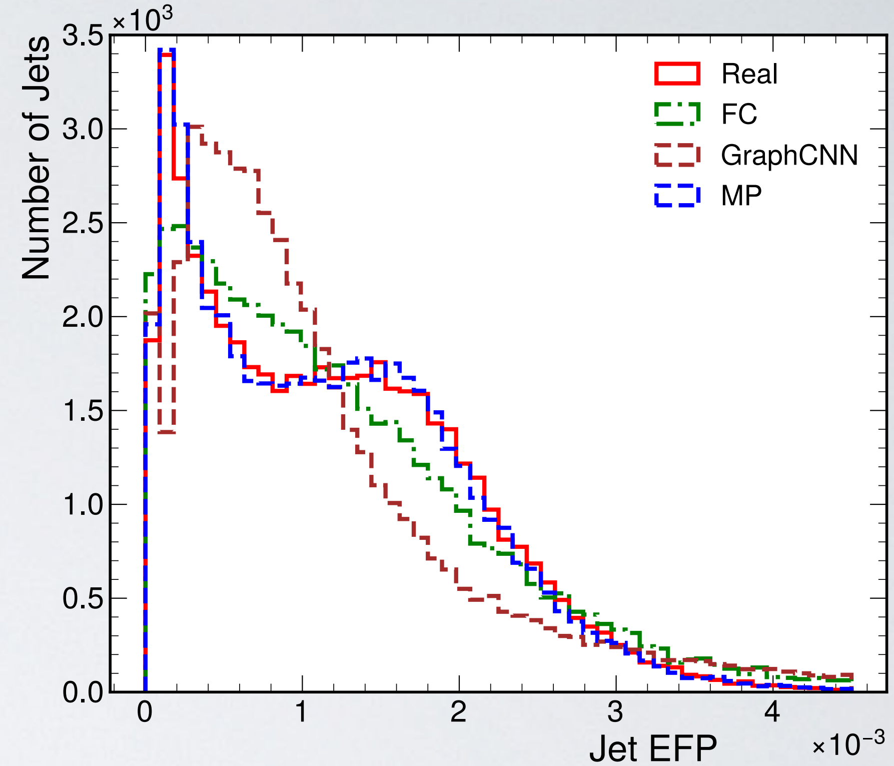




Real vs real
W1-P = (0.55 ± 0.07) × 10^{-3}



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