# Removing negative weights in Monte Carlo event samples

Andreas Maier

22 April 2024

J. R. Andersen, A. Maier, D. Maître Eur.Phys.J.C 83 (2023) 9, 835
J. R. Andersen, A. Maier Eur.Phys.J.C 82 (2022) 5, 433
J. R. Andersen, C. Gütschow, A. Maier, S. Prestel Eur.Phys.J.C 80 (2020) 11, 1007
+ ongoing work with Ana Cueto, Ella Cole, Stephen Jones

# What are event weights?

## Leading-order cross sections

Example: prediction for dijet production cross section

1. Relate to partonic cross section

$$\sigma_{2\text{ jets}} \stackrel{\text{LO}}{=} \sigma_{2\text{ partons}}$$

2. Simulate partonic scattering events with weights $w_i$
   - Computed from scattering matrix elements + PDF + phase space factor
   - Weights proportional to probability: $w_i > 0$
   - Sum of weights gives the cross section:

$$\sigma_{2\text{ partons}} = \sum_i w_i$$

# What are negative event weights?

## Next-to-leading-order cross sections

Example: prediction for dijet production cross section

1. Relate to partonic cross section

$$\sigma_{2\text{ jets}} \overset{\text{NLO}}{=} \sigma_{2\text{ partons}} + \sigma_{3\text{ partons}}$$

2. Simulate partonic scattering events

$$\sigma_{2\text{ partons}} = \sum_i w_i$$

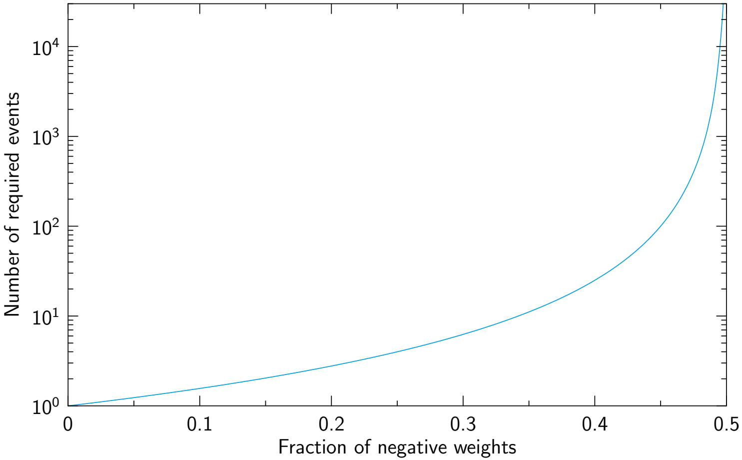
$$\sigma_{3\text{ partons}} = \sum_j w_j$$

$\sigma_{2\text{ partons}}, \sigma_{3\text{ partons}}$ not separately observable:

Events weights can be either positive or negative

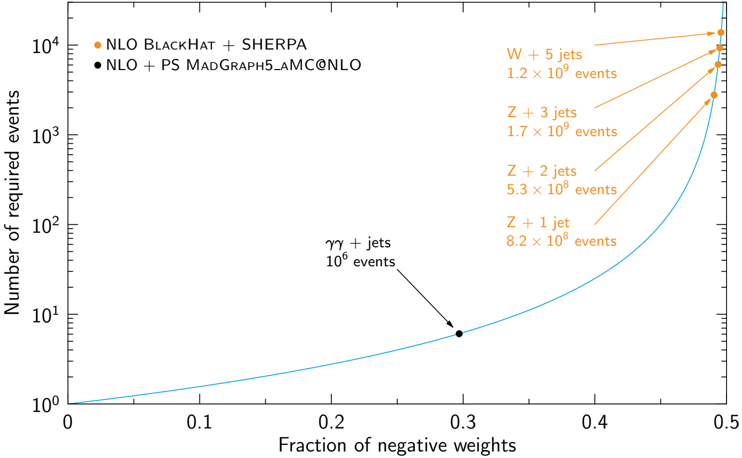
# Why are negative event weights a problem?

Number of unweighted events to reach given accuracy:



# Why are negative event weights a problem?

Number of unweighted events to reach given accuracy:

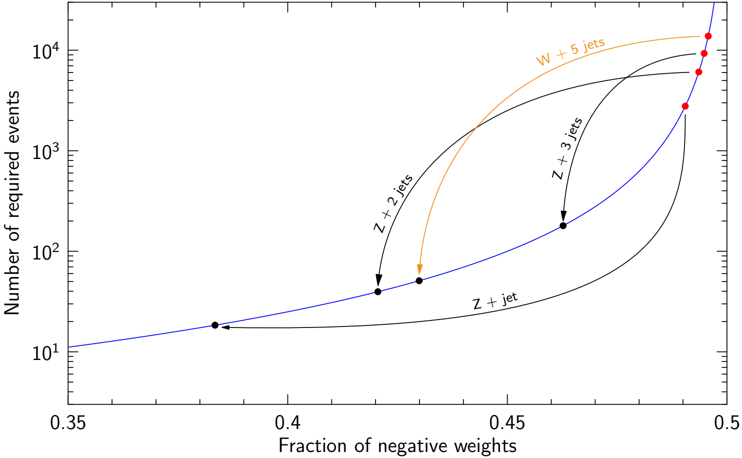



V + jets: Phys. Rev. D 88 (2013) 014025, Phys. Rev. D 97 (2018) 096010

$\gamma\gamma$ + jets: parameters from background modelling for ATLAS $H \to \gamma\gamma$ measurement arXiv:2306.11379

# Cell resampling for V + jets at NLO

## Negative weights

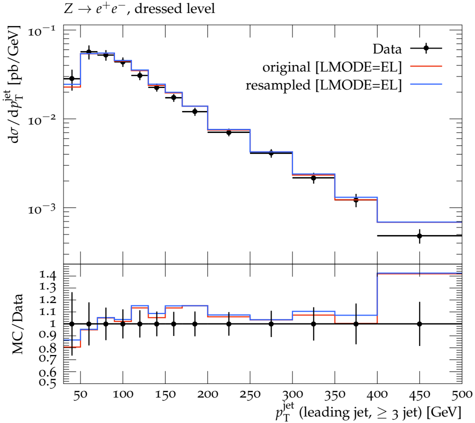


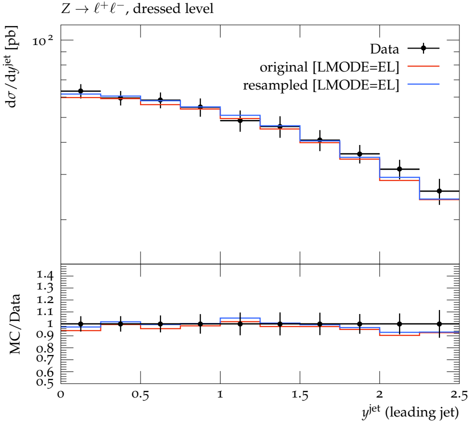

Cell resampling drastically reduces the number of required events

# Cell resampling for V + jets at NLO

## Predictions

Analysis from ATLAS, Eur. Phys. J. C77 (2017) 361:

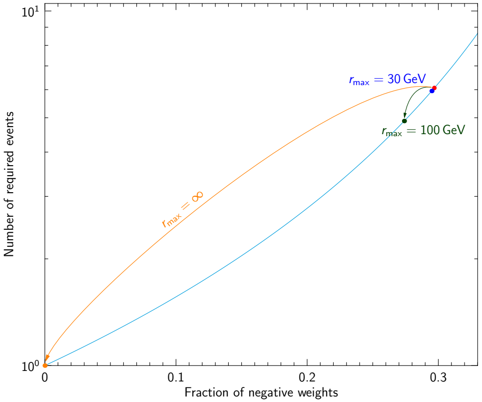


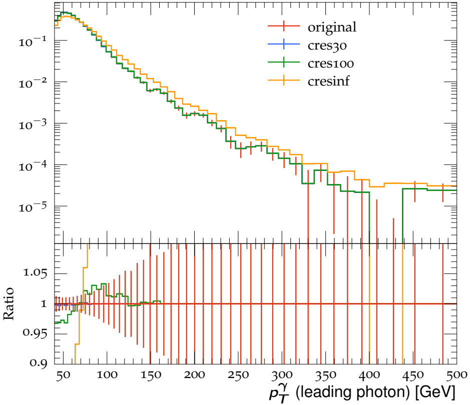




Cell resampling preserves predictions within a few per cent

# Work in progress: showered samples

$pp \to \gamma\gamma +$ jets, $10^6$ events:

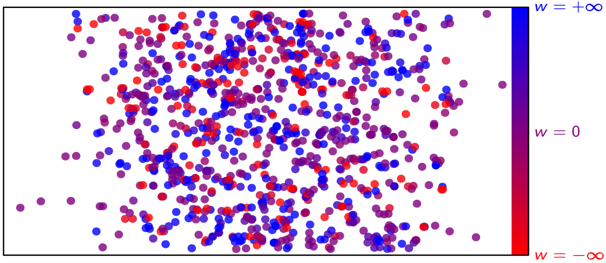



Negative-weight reduction more efficient for large weighted samples

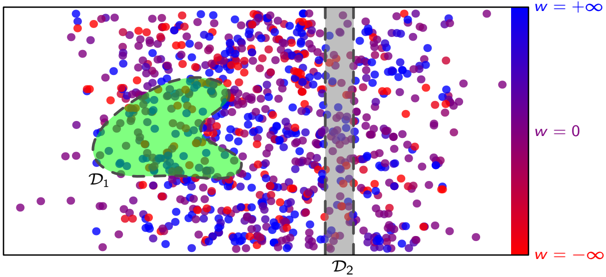
# Observables

Weighted events in 2D projection of phase space:



# Observables

Weighted events in 2D projection of phase space:




Observables $\mathcal{O}$:

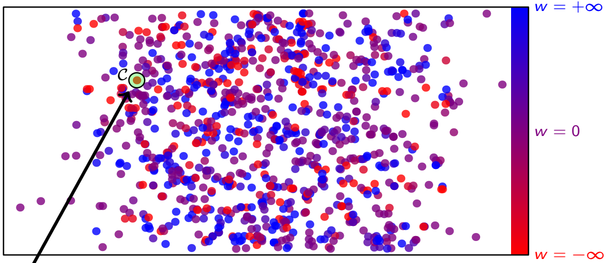
- Select region $\mathcal{D}$ in phase space $\geq$ experimental resolution
- $\mathcal{O} = \sum_{i \in \mathcal{D}} w_i \geq 0$ with sufficient statistics

e.g. histogram bins

Redistribute weights without affecting any observable

# Cell resampling

[Andersen, Maier 2021]

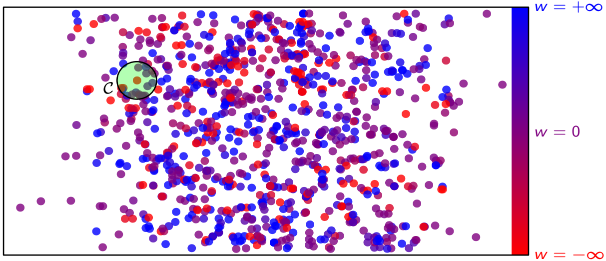



Cell resampling:

1. Choose seed event with negative weight for cell $\mathcal{C}$

# Cell resampling

[Andersen, Maier 2021]



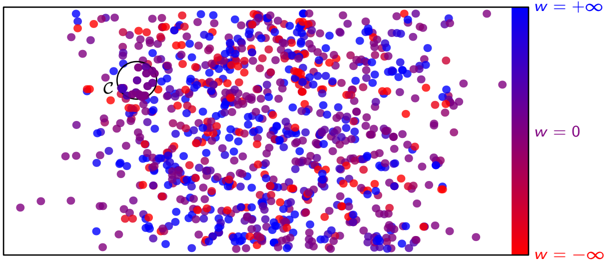

Cell resampling:

1. Choose seed event with negative weight for cell $\mathcal{C}$
2. Iteratively add nearest event to cell until $\sum_{i \in \mathcal{C}} w_i \geq 0$ or radius exceeds $r_{\text{max}}$

Cells get systematically smaller with increasing statistics

# Cell resampling

**[Andersen, Maier 2021]**

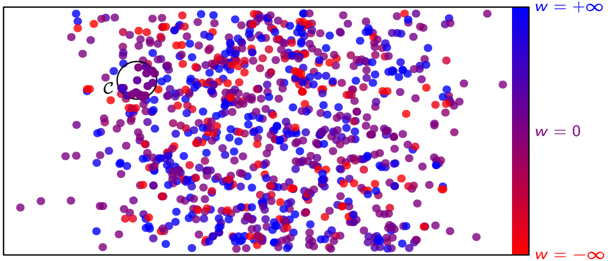



## Cell resampling:

1. Choose seed event with negative weight for cell $\mathcal{C}$
2. Iteratively add nearest event to cell until $\sum_{i \in \mathcal{C}} w_i \geq 0$ or radius exceeds $r_{\text{max}}$
3. Redistribute weights, e. g. average over cell: $w_i \to w = \frac{\sum_{j \in \mathcal{C}} w_j}{\text{\# events in } \mathcal{C}}$
4. Repeat

# Cell resampling

[Andersen, Maier 2021]




## Cell resampling:

1. Choose seed event with negative weight for cell $\mathcal{C}$
2. Iteratively add nearest event to cell until $\sum_{i \in \mathcal{C}} w_i \geq 0$ or radius exceeds $r_{\text{max}}$
   What does "nearest" mean?
3. Redistribute weights, e. g. average over cell: $w_i \to w = \frac{\sum_{j \in \mathcal{C}} w_j}{\text{\# events in } \mathcal{C}}$
4. Repeat

# Distances in phase space

Criteria for distance function:

- Small distance between events that look similar in detector or differ only in properties the event generator can't predict
- Large distance between events that look different in detector

Define distance in terms of infrared & collinear safe objects, e.g. jets

# Distances in phase space

Criteria for distance function:

- Small distance between events that look similar in detector
  or differ only in properties the event generator can't predict
- Large distance between events that look different in detector

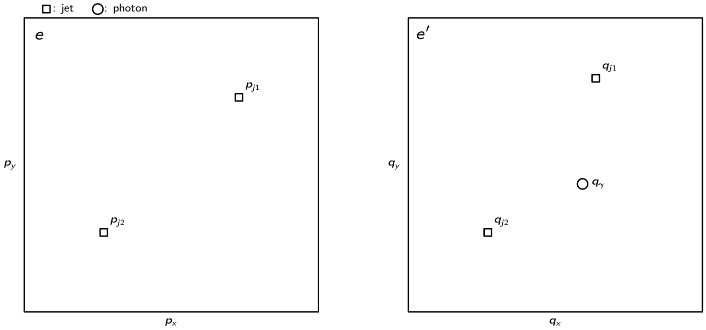
Define distance in terms of infrared & collinear safe objects, e.g. jets

Current choice:

1. Find optimal pairing between observable objects in both events
2. Sum up spatial momentum differences

# Distances in phase space

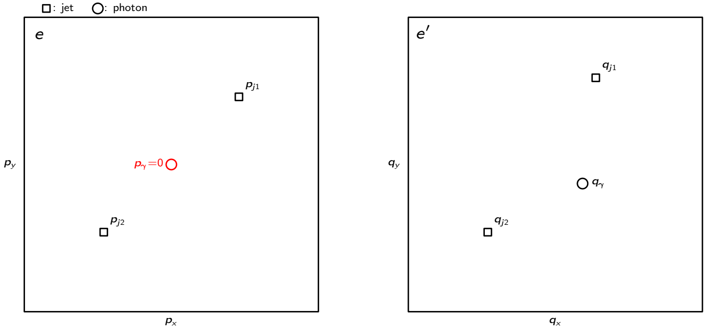
## Example



# Distances in phase space

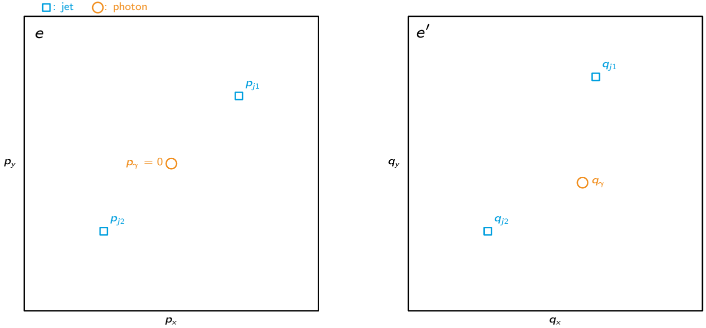
## Example

Ensure same multiplicities



# Distances in phase space
## Example

Compare physics objects of same type

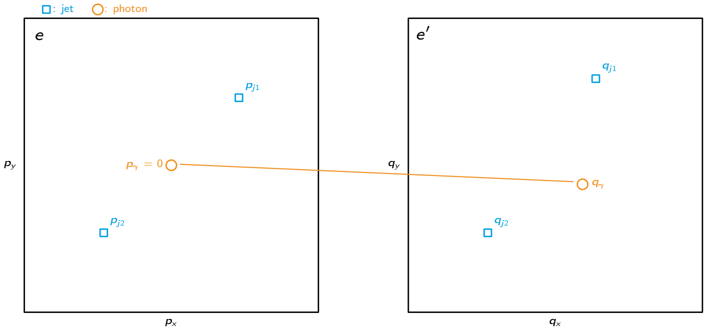



$$d(e, e') = d(s_j, s'_j) + d(s_\gamma, s'_\gamma)$$

# Distances in phase space

## Example

Compare photons

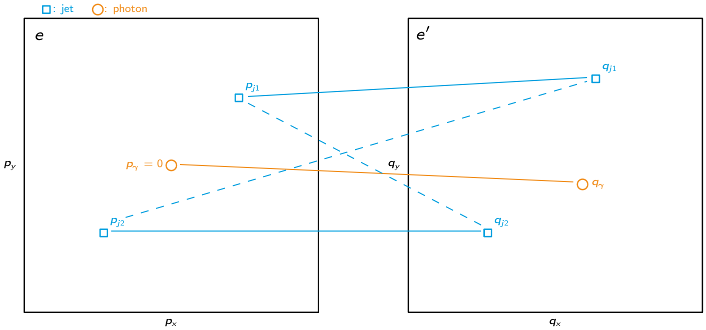



$$
\begin{aligned}
d(e, e') &= d(s_j, s_j') + d(s_\gamma, s_\gamma') \\
&= d(s_j, s_j') + d(p_\gamma, q_\gamma)
\end{aligned}
$$

# Distances in phase space

## Example

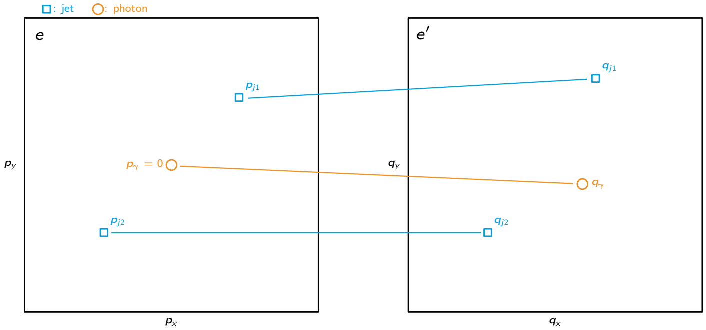
Compare jets: find pairs of most similar jets




$$
\begin{aligned}
d(e, e') &= d(s_j, s'_j) + d(s_\gamma, s'_\gamma) \\
&= \min\left[d(p_{j1}, q_{j1}) + d(p_{j2}, q_{j2}), d(p_{j1}, q_{j2}) + d(p_{j2}, q_{j1})\right] + d(p_\gamma, q_\gamma)
\end{aligned}
$$

# Distances in phase space

## Example

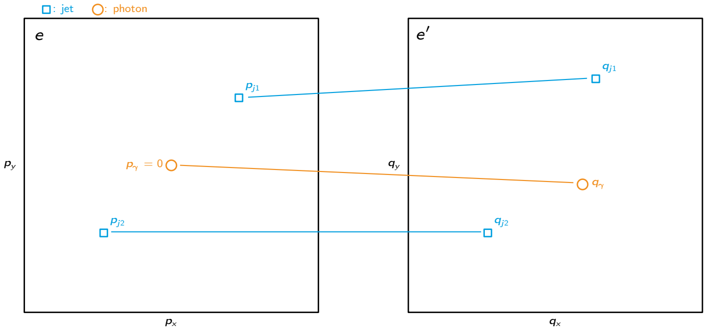



$$
\begin{aligned}
d(e, e') &= d(s_j, s'_j) + d(s_\gamma, s'_\gamma) \\
&= d(p_{j1}, q_{j1}) + d(p_{j2}, q_{j2}) + d(p_\gamma, q_\gamma)
\end{aligned}
$$

# Distances in phase space

## Example

Compare momenta




$$d(e, e') = d(s_j, s_j') + d(s_\gamma, s_\gamma')$$
$$= |\vec{p}_{j1} - \vec{q}_{j1}| + |\vec{p}_{j2} - \vec{q}_{j2}| + |\vec{p}_\gamma - \vec{q}_\gamma|$$

# Distances in phase space

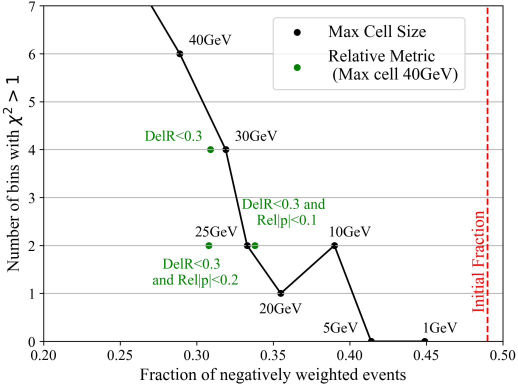
## Work in progress: relative distance

Alternative metric based on relative momentum differences:

- Closer to experimental sensitivity
- Amount of cancellation better aligned with statistical uncertainty

Initial exploration for W $p_\perp$ distribution in W + 1 jet:

[Plot by Ella Cole]



# Runtime scaling

- Runtime for generating $N$ events: $\mathcal{O}(N)$
- Naive cell resampling: $\mathcal{O}(N^2)$
  - Expect: $\#\text{cells} \propto (\#\text{events with } w < 0) \propto \#\text{events}$
  - Naive (k) nearest neighbour search: $\mathcal{O}(N)$

Need faster nearest-neighbour search

# Runtime scaling

- Runtime for generating $N$ events: $\mathcal{O}(N)$
- Naive cell resampling: $\mathcal{O}(N^2)$
  - Expect: $\#$cells $\propto$ ($\#$events with $w < 0$) $\propto$ $\#$events
  - Naive (k) nearest neighbour search: $\mathcal{O}(N)$

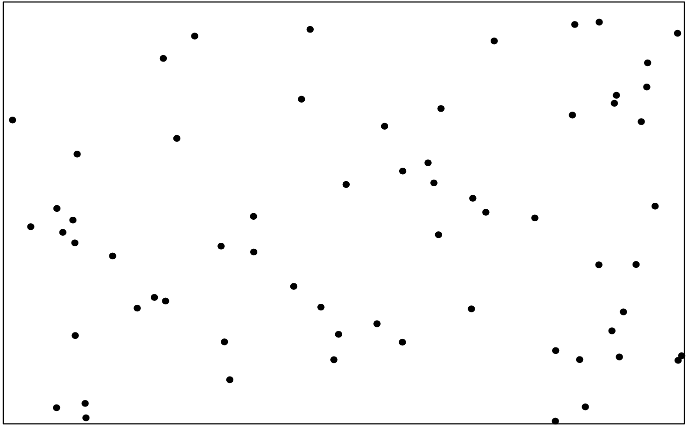
Need faster nearest-neighbour search

Small number of dimensions:

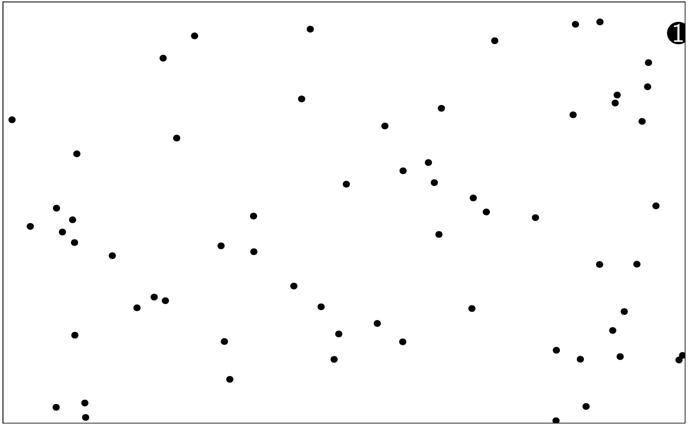
- Histograms
- Voronoi cells ($\to$ jet clustering in `fastjet`)

Here: vantage-point tree search

# Nearest-neighbour search



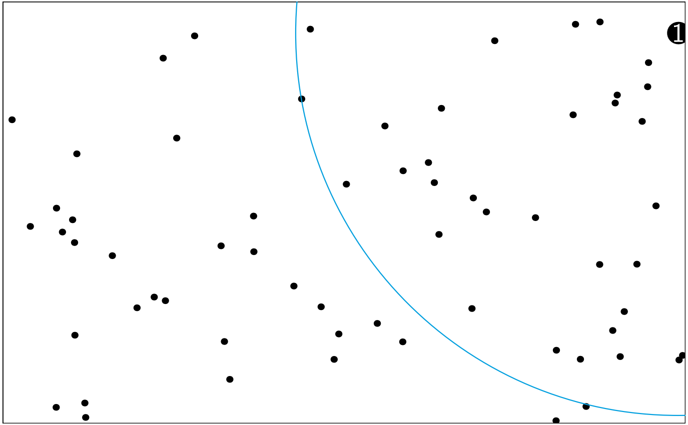
# Nearest-neighbour search



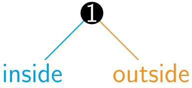

Vantage-point tree:

➊

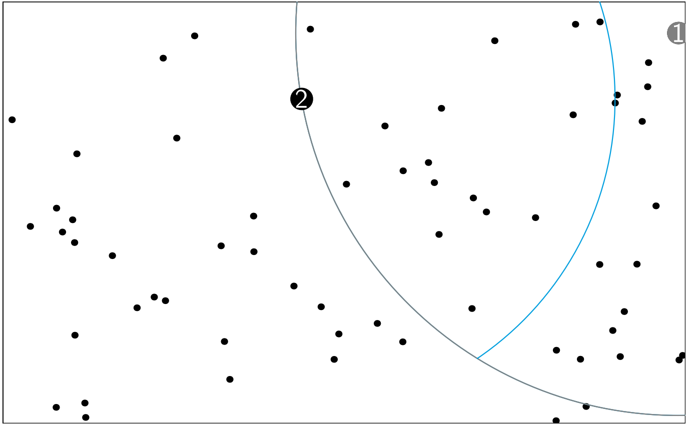
# Nearest-neighbour search



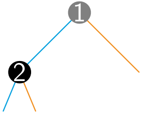

Vantage-point tree:



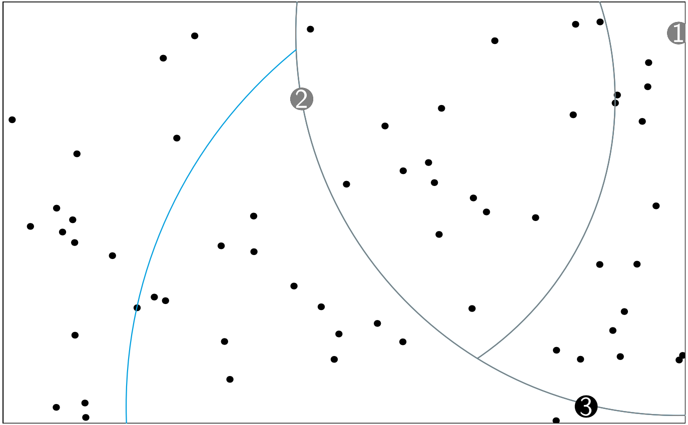
# Nearest-neighbour search



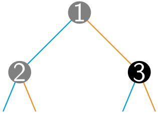

Vantage-point tree:



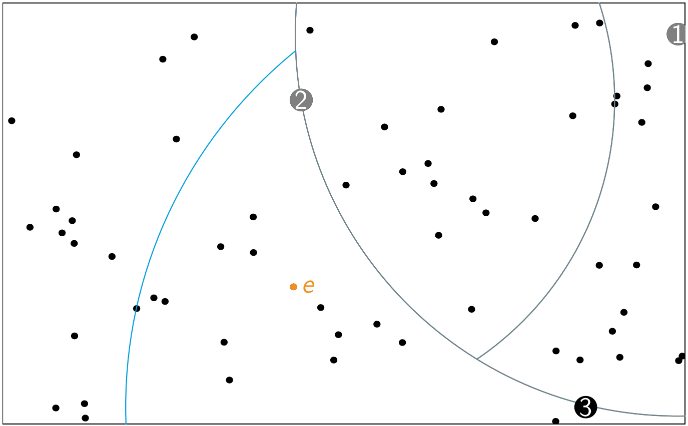
# Nearest-neighbour search



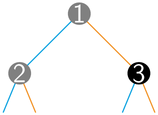

Vantage-point tree:



# Nearest-neighbour search




Vantage-point tree:




Search nearest neighbour for *e*:

- Find candidate in region containing *e*
- Search neighbouring regions only if better candidate may be found

# Summary

- Negative event weights lead to slow statistical convergence
- Idea: remove negative weights by smearing over small phase space regions
  - Potential to reduce the number of required events by orders of magnitude
  - Preserves predictions of observables
  - Agnostic with respect to process and observables
  - Automatic improvement with increasing statistics
  - Computationally efficient: $\sim 55$ CPU hours for one billion events (W + 5 jets)

# Summary

- Negative event weights lead to slow statistical convergence
- Idea: remove negative weights by smearing over small phase space regions
  - Potential to reduce the number of required events by orders of magnitude
  - Preserves predictions of observables
  - Agnostic with respect to process and observables
  - Automatic improvement with increasing statistics
  - Computationally efficient: $\sim$ 55 CPU hours for one billion events (W + 5 jets)

Outlook:

- Application to parton showered samples ✓ [Andersen, Cueto, Maier, Jones]
- IRC safety with electroweak corrections [Andersen, Maier, Schönherr]
- Systematic estimate of uncertainties
- Integrate into existing workflows
- Guide Monte Carlo event generation? [Andersen, Maier, Maître, Schönherr]

# Backup

# Why do we need many more events?

Consider $N$ uncorrelated unweighted events:

- $w_i = -w$ for $i \leq N_-$
- $w_i = +w$ for $i > N_-$

$\Rightarrow$ Negative weight fraction: $r_- = \frac{N_-}{N}$

Then

- $\sigma = \sum_i w_i = -N_- w + (N - N_-)w = (1 - 2r_-)Nw$
- $\Delta\sigma = \sqrt{\sum_i w_i^2} = \sqrt{N}w$

To reach given relative uncertainty $\frac{\Delta\sigma}{\sigma} = \frac{1}{(1-2r_-)\sqrt{N(r_-)}} = \frac{1}{\sqrt{N(0)}}$:

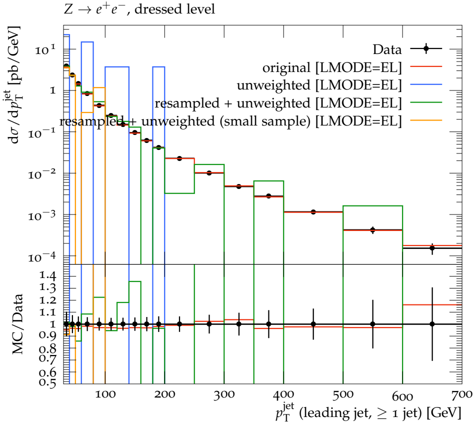
$$N(r_-) = \frac{N(0)}{(1 - 2r_-)^2}$$

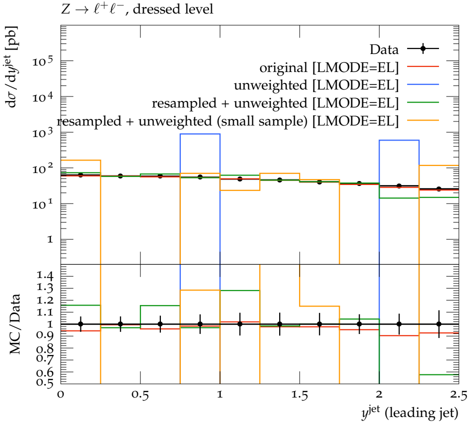
# Event samples

[BlackHat + Sherpa 2013 + 2017]

| Sample | Process | Centre-of-mass energy | # events |
|---|---|---|---|
| `Z1` | $pp \to (Z \to e^+e^-) + \text{jet}$ | 13 TeV | $8.21 \times 10^8$ |
| `Z2` | $pp \to (Z \to e^+e^-) + 2\ \text{jets}$ | 13 TeV | $5.30 \times 10^8$ |
| `Z3` | $pp \to (Z \to e^+e^-) + 3\ \text{jets}$ | 13 TeV | $1.65 \times 10^9$ |
| `W5` | $pp \to (W^- \to e^-\nu_e) + 5\ \text{jets}$ | 7 TeV | $1.17 \times 10^9$ |

# Unweighting for Z + jet



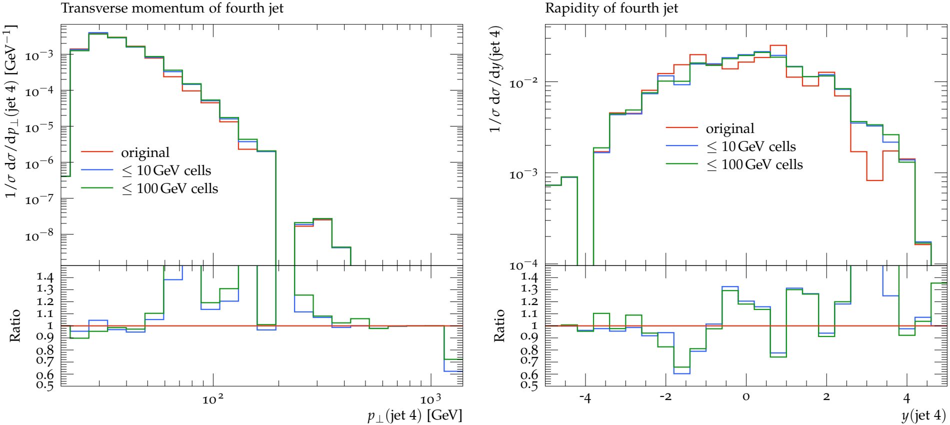

original: $8.21 \times 10^8$ events

unweighted: 320 events

resampled + unweighted: 11574 events

resampled + unweighted (small sample): 320 events

# Resampling for W + 5 jets



# Distances in phase space

Need distance function $d(e, e')$ between events $e, e'$

- Essential: $d(e, e')$ small $\Rightarrow$ $e, e'$ look similar in detector or differ only in properties the event generator can't predict
- Desirable: $d(e, e')$ large $\Rightarrow$ $e, e'$ look different in detector

# Distances in phase space

Need distance function $d(e, e')$ between events $e, e'$

- Essential: $d(e, e')$ small $\Rightarrow$ $e, e'$ look similar in detector or differ only in properties the event generator can't predict
- Desirable: $d(e, e')$ large $\Rightarrow$ $e, e'$ look different in detector

Example: infrared safety

- $d(e, e')$ unaffected by collinear splittings with $\Theta \to 0$
- $d(e, e')$ unaffected by soft particles with $p \to 0$

$\Rightarrow$ define distance in terms of infrared-safe physics objects, e.g. jets

Here: Example for fixed-order (QCD) event generator

# Distances in phase space

## Concrete implementation

1. Collect all infrared-safe objects in event $e$ into sets $\{\overset{\text{jets}}{s_1}, \overset{\text{electrons}}{s_2}, \ldots, s_T\}$

$$d(e, e') = \sum_{t=1}^{T} d(s_t, s'_t)$$

# Distances in phase space

## Concrete implementation

jets electrons

1. Collect all infrared-safe objects in event $e$ into sets $\{ \overset{\text{jets}}{s_1}, \overset{\text{electrons}}{s_2}, \ldots, s_T \}$

$$d(e, e') = \sum_{t=1}^{T} d(s_t, s_t')$$

2. Objects in $s_t$ have four-momenta $( p_1, \ldots \ldots\ldots, p_P )$

Objects in $s_t'$ have four-momenta $( q_1, \ldots, q_Q, 0, \ldots, 0 )$

$$d(s_t, s_t') = \min_{\sigma \in S_P} \sum_{i=1}^{P} d_t(p_i, q_{\sigma(i)})$$

# Distances in phase space

## Concrete implementation

1. Collect all infrared-safe objects in event $e$ into sets $\{\underset{\text{jets}}{s_1}, \underset{\text{electrons}}{s_2}, \ldots, s_T\}$

$$d(e, e') = \sum_{t=1}^{T} d(s_t, s_t')$$

2. Objects in $s_t$ have four-momenta $(p_1, \ldots \ldots\ldots\ldots, p_P)$

   Objects in $s_t'$ have four-momenta $(q_1, \ldots, q_Q, 0, \ldots, 0)$

$$d(s_t, s_t') = \min_{\sigma \in S_P} \sum_{i=1}^{P} d_t(p_i, q_{\sigma(i)})$$

# Distances in phase space

## Concrete implementation

1. Collect all infrared-safe objects in event $e$ into sets $\{\overset{\text{jets}}{s_1}, \overset{\text{electrons}}{s_2}, \dots, s_T\}$

$$d(e, e') = \sum_{t=1}^{T} d(s_t, s'_t)$$

2. Objects in $s_t$ have four-momenta $(p_1, \dots \dots\dots\dots, p_P)$

   Objects in $s'_t$ have four-momenta $(q_1, \dots, q_Q, 0, \dots, 0)$

$$d(s_t, s'_t) = \min_{\sigma \in S_P} \sum_{i=1}^{P} d_t(p_i, q_{\sigma(i)})$$

Efficient minimisation: Hungarian algorithm [Jacobi 1890]

# Distances in phase space

## Concrete implementation

1. Collect all infrared-safe objects in event $e$ into sets $\{\overset{\text{jets}}{s_1}, \overset{\text{electrons}}{s_2}, \ldots, s_T\}$

$$d(e, e') = \sum_{t=1}^{T} d(s_t, s'_t)$$

2. Objects in $s_t$ have four-momenta $(\, p_1 \,, \ldots \ldots\ldots\ldots, p_P \,)$
   Objects in $s'_t$ have four-momenta $(\, q_1 \,, \ldots \,, q_Q, 0, \ldots, 0 \,)$

$$d(s_t, s'_t) = \min_{\sigma \in S_P} \sum_{i=1}^{P} d_t(p_i, q_{\sigma(i)})$$

3. Choose distance function between particle momenta
   Here: independent of particle type $t$, do not consider internal structure

$$d_t(p, q) = \sqrt{(\vec{p} - \vec{q})^2 + \tau^2 (p_\perp - q_\perp)^2} \qquad \tau\text{: tunable parameter}$$

# Computing requirements

## Memory

Fast + exact nearest-neighbour search: keep all events in memory

Need $\sim$ (byte size of event) GB for $\sim 10^9$ events

# Computing requirements

## Memory

Fast + exact nearest-neighbour search: keep all events in memory

Need $\sim$ (byte size of event) GB for $\sim 10^9$ events

Only store relevant event data: weights + momenta of outgoing analysis objects

Read + convert events $\longrightarrow$ Cell resampling $\longrightarrow$ Read + write events

# Computing requirements

## Memory

Fast + exact nearest-neighbour search: keep all events in memory

Need $\sim$ (byte size of event) GB for $\sim 10^9$ events

Only store relevant event data: weights + momenta of outgoing analysis objects

Read + convert events → Cell resampling → Read + write events

Current requirements:

- Persistent event samples with reasonably fast sequential access
- 300 GB to 400 GB of memory per $10^9$ events,
  no huge increase from showering expected

# Computing requirements

## Memory

Fast + exact nearest-neighbour search: keep all events in memory

Need $\sim$ (byte size of event) GB for $\sim 10^9$ events

Only store relevant event data: weights + momenta of outgoing analysis objects

Read + convert events → Cell resampling → Read + write events

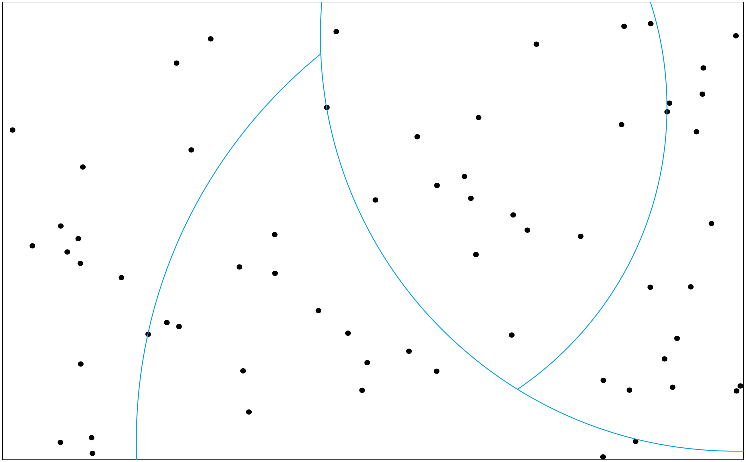
Current requirements:

- Persistent event samples with reasonably fast sequential access
- 300 GB to 400 GB of memory per $10^9$ events,
  no huge increase from showering expected

Can we go beyond $\sim 10^9$ events?

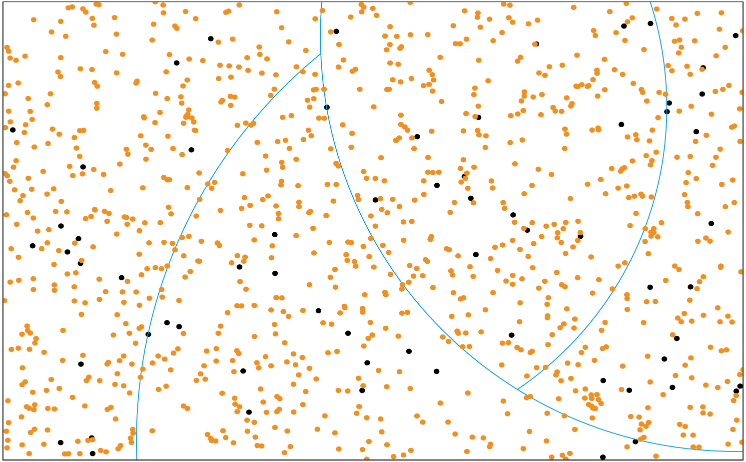
# Work in progress: memory efficiency

1. Partition phase space using vantage-point tree from small event sample



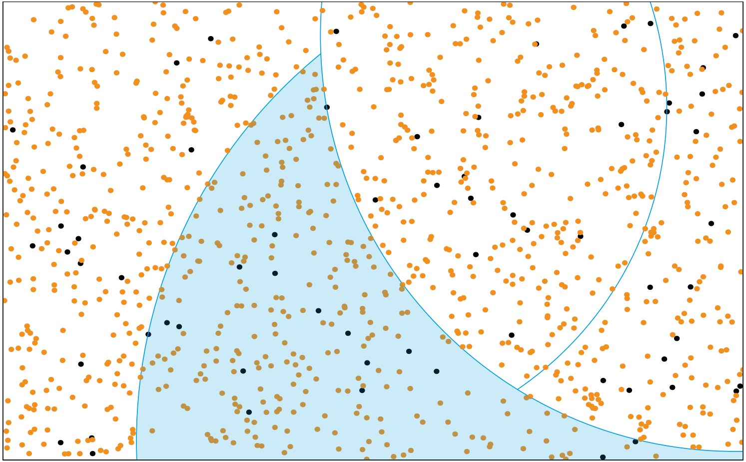
# Work in progress: memory efficiency

2. Identify region for each event in large sample



# Work in progress: memory efficiency

3. Independent cell resampling for each region



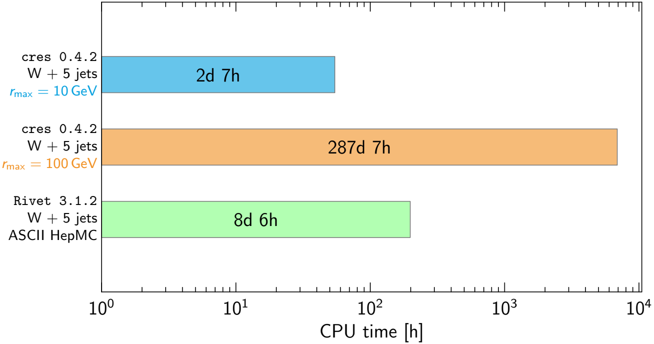
# Computing requirements

## CPU time

Benchmark machines:

| # Cores | CPU model | Memory | Age |
|---|---|---|---|
| 20 | XEON E5-2640 @ 2.40GHz | 400GB | ∼7 years |
| 12 | XEON E5-2643 @ 3.40GHz | 800GB | ∼6 years |

Local rotating disks, RAID 6



# Computing requirements

## Wall-clock time

Benchmark machines:

| # Cores | CPU model | Memory | Age |
|---|---|---|---|
| 20 | XEON E5-2640 @ 2.40GHz | 400GB | ∼7 years |
| 12 | XEON E5-2643 @ 3.40GHz | 800GB | ∼6 years |

Local rotating disks, RAID 6