Differentiable Jet Clustering

ML equivalent: Differentiating stochastic programs

AK*, N. Hartmann*, B. Schölkopf, M. Kagan, L. Heinrich

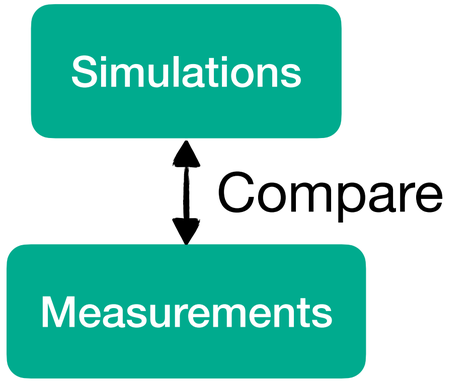
* equal contribution

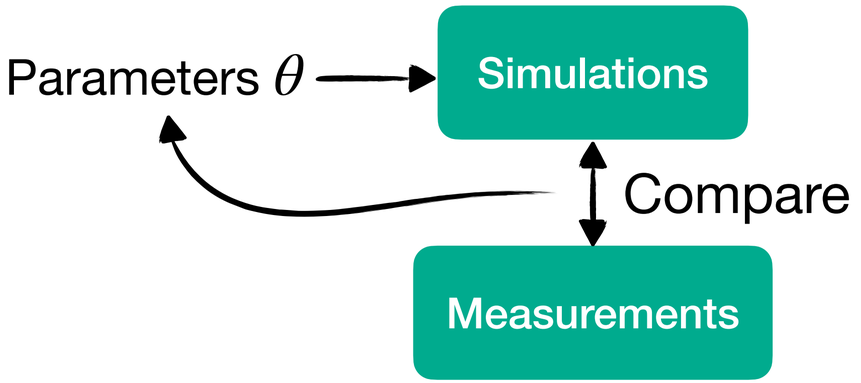
Annalena Kofler, September 12th, 2025

MIAPbP workshop ”Building big or building smart”

Physics simulations

- Simulations contain our physics knowledge

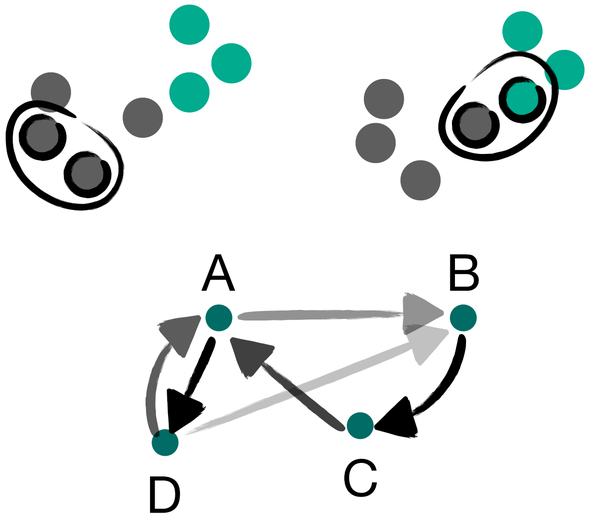




- How to find the optimal θ? $\rightarrow$ “Grad student gradient descent”
- Obvious solution: Just make your simulator differentiable!

Problem: discrete decisions

- Many physics simulators are not differentiable by definition
- Examples:
  - Clustering: Which two points to cluster together?
  - Stochastic processes: Which reaction happens next?
  - …
- Common denominator: Gradient of discrete decisions



$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}[f(k)] \text{ where } \quad p_\theta(k) = \mathrm{Ber}(k;\theta) = \begin{cases} 1-\theta & \text{if } k=0 \\ \theta & \text{if } k=1 \end{cases}$$

In this talk:

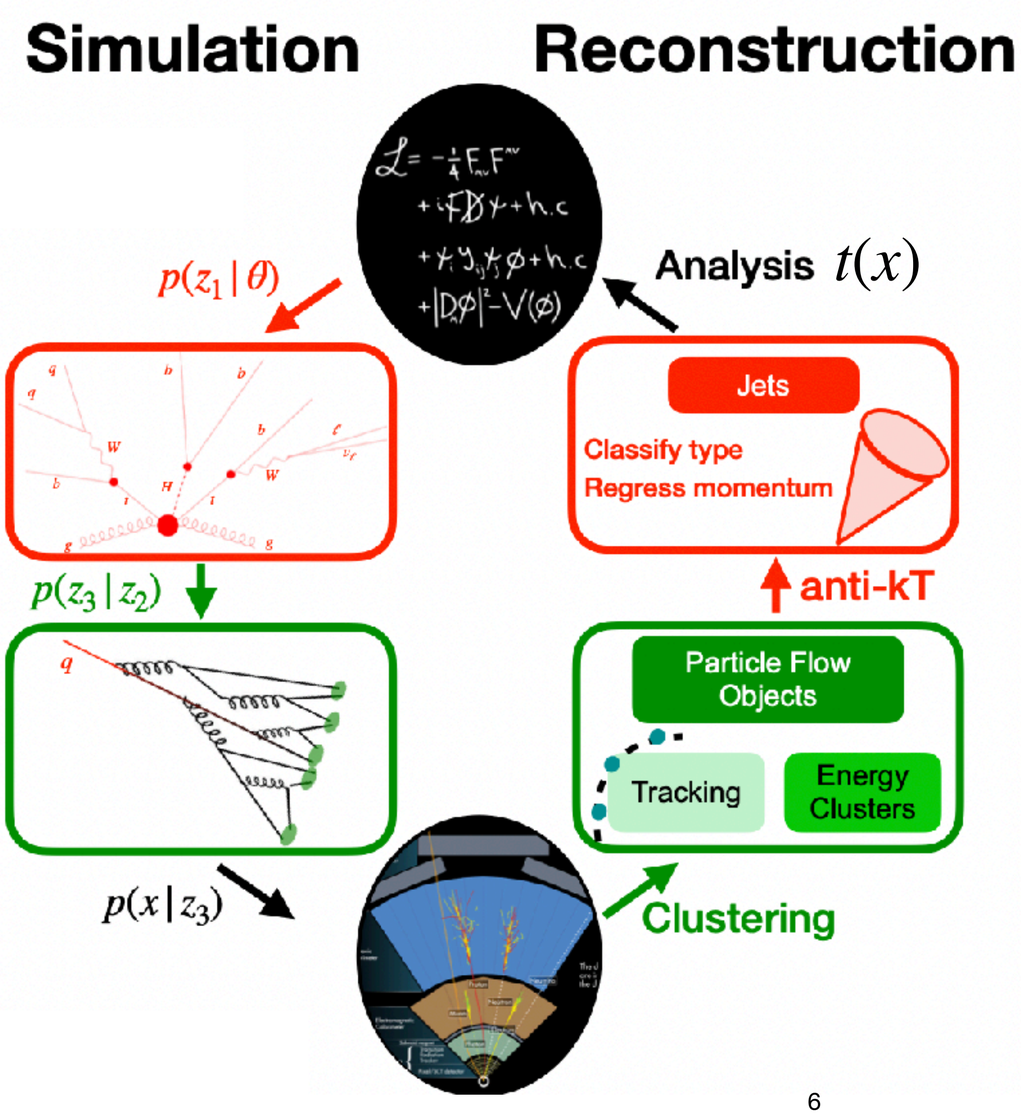
How to make discrete decisions differentiable
At the example of jet clustering

But: Applicable to many other physics problems

- Detector design in particle physics — Kagan & Heinrich, 2023, arXiv:2308.16680
- Reaction networks in biophysics — Burger, Kofler, et al., 2025, in preparation
- Heavy ion collision simulation — Megorsch, et al., 2025, in preparation

What is jet clustering and why is it important?

End-goal: Fully differentiable particle physics analysis



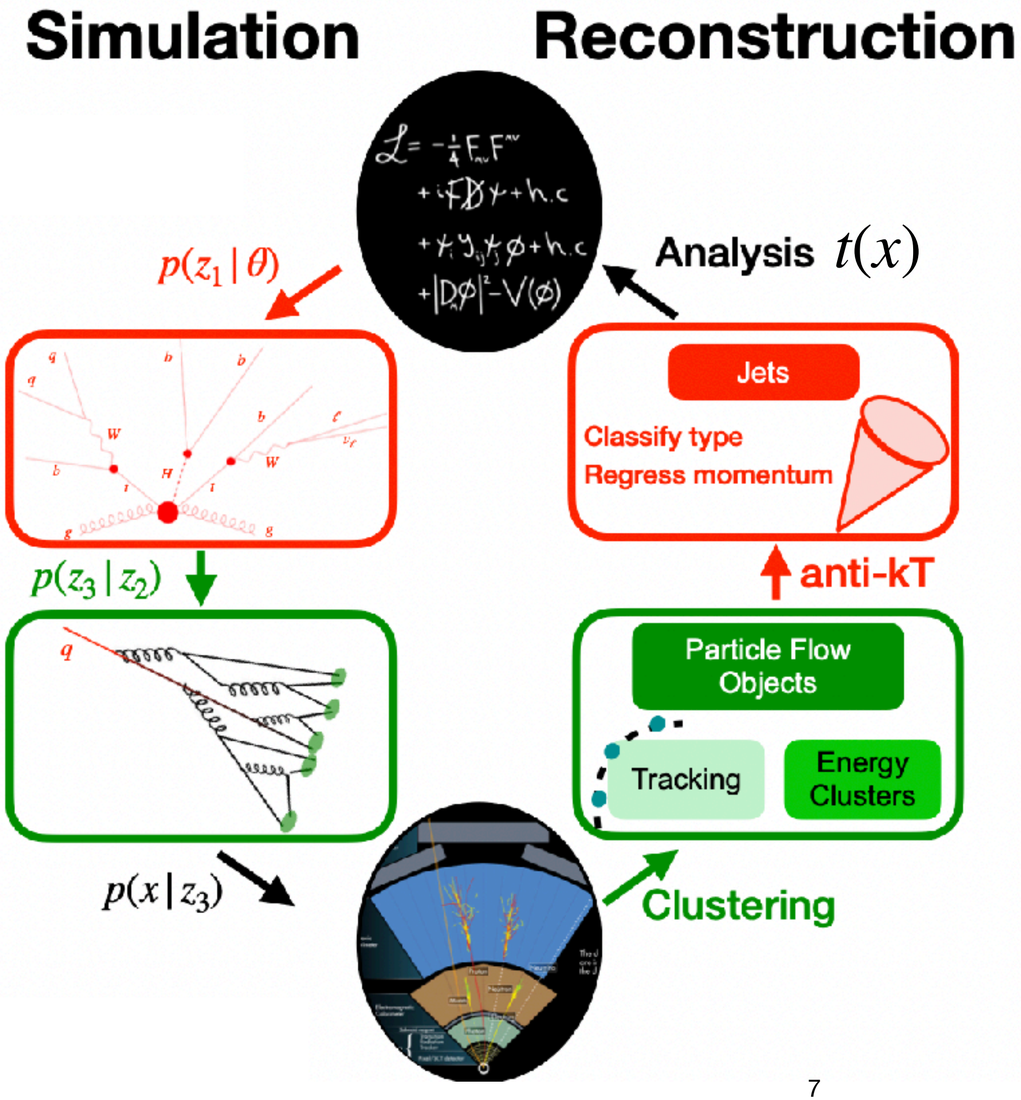
Slide adapted from K. Cranmer, H&N 2019

Slide adapted from N. Hartman, MIAPbP 2025

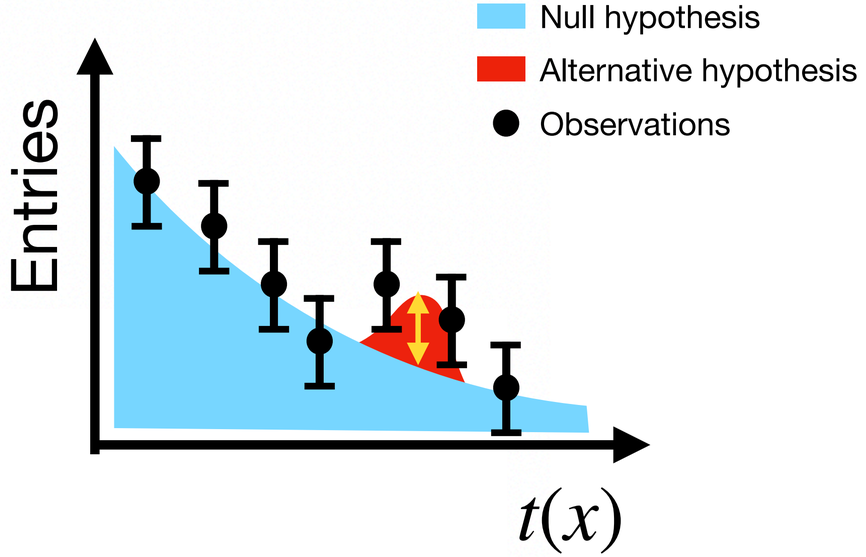
Slide adapted from G. Watts, MIAPbP 2025

Slide adapted from M. Algren, MIAPbP 2025

End-goal: Fully differentiable particle physics analysis

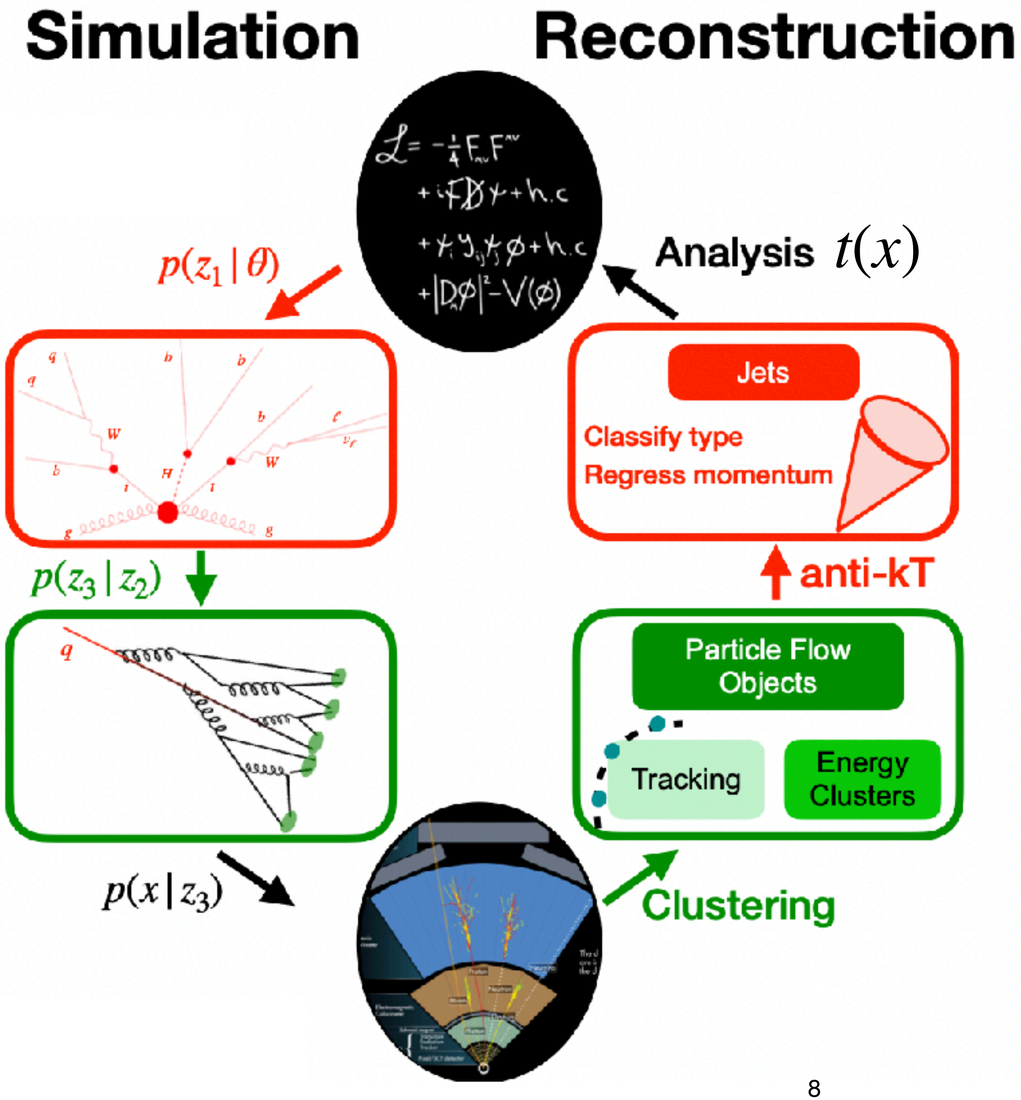


Improve?

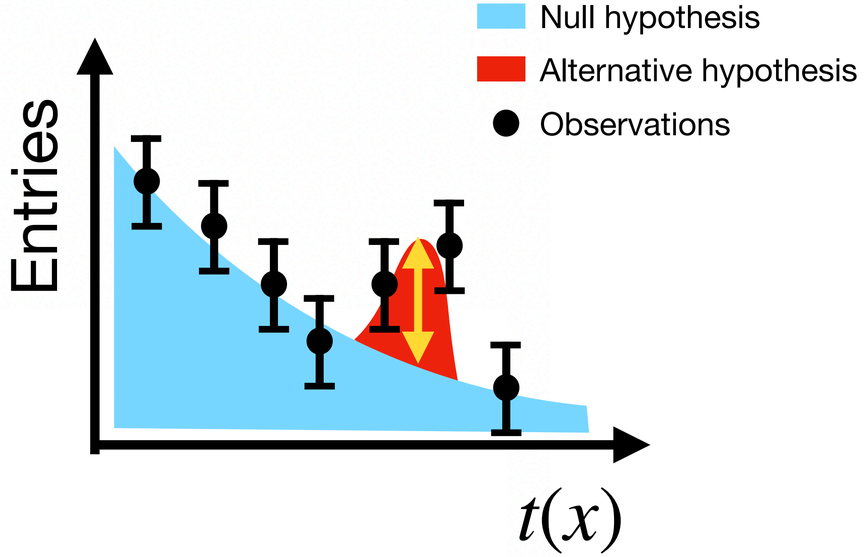


$t(x)$: smart summary statistics

End-goal: Fully differentiable particle physics analysis

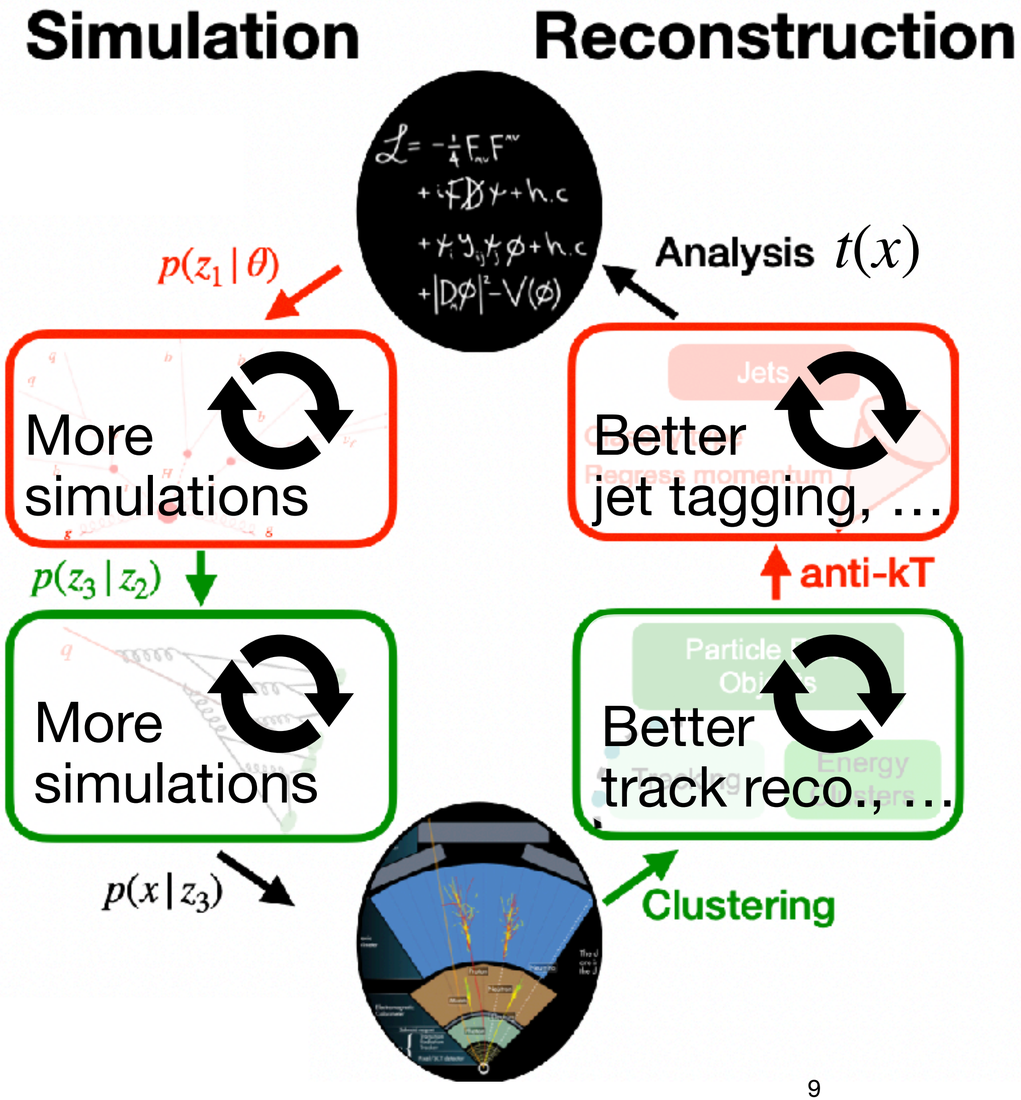


Improve?

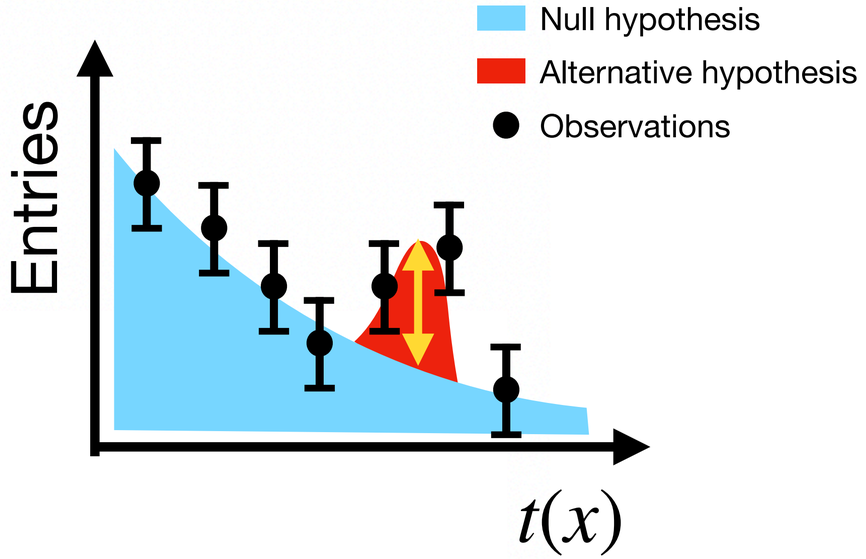


$t(x)$: smart summary statistics

End-goal: Fully differentiable particle physics analysis

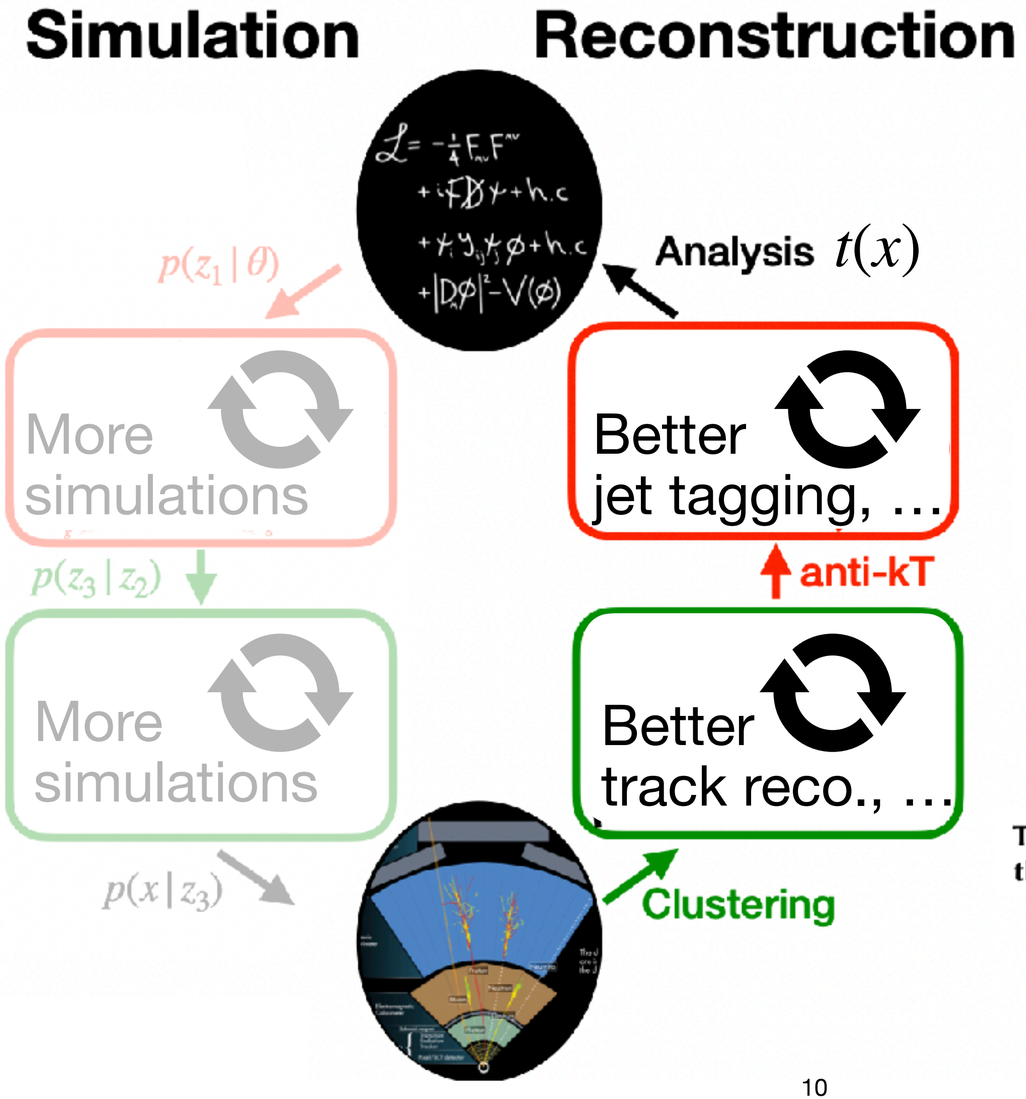


How?



$t(x)$: smart summary statistics

End-goal: Fully differentiable particle physics analysis



Train better ML models

Particle Transformer for Jet Tagging

Huilin Qu [1] Congqiao Li [2] Sitian Qian [2]

PCN: A Deep Learning Approach to Jet Tagging Utilizing Novel Graph Construction Methods and Chebyshev Graph Convolutions

Yash Semlani[a,1] Mihir Relan[b,1] Krithik Ramesh[b,2]

Is Tokenization Needed for Masked Particle Modelling?

Matthew Leigh University of Geneva matthew.leigh@unige.ch

Samuel Klein University of Geneva samuel.klein@unige.ch

François Charton Meta FAIR fcharton@meta.com

OmniJet-α: The first cross-task foundation model for particle physics

Joschka Birk,[1,*] Anna Hallin,[1,†] and Gregor Kasieczka[1]

Finetuning Foundation Models for Joint Analysis Optimization

Matthias Vigl,[1] Nicole Hartman,[1] and Lukas Heinrich[1]

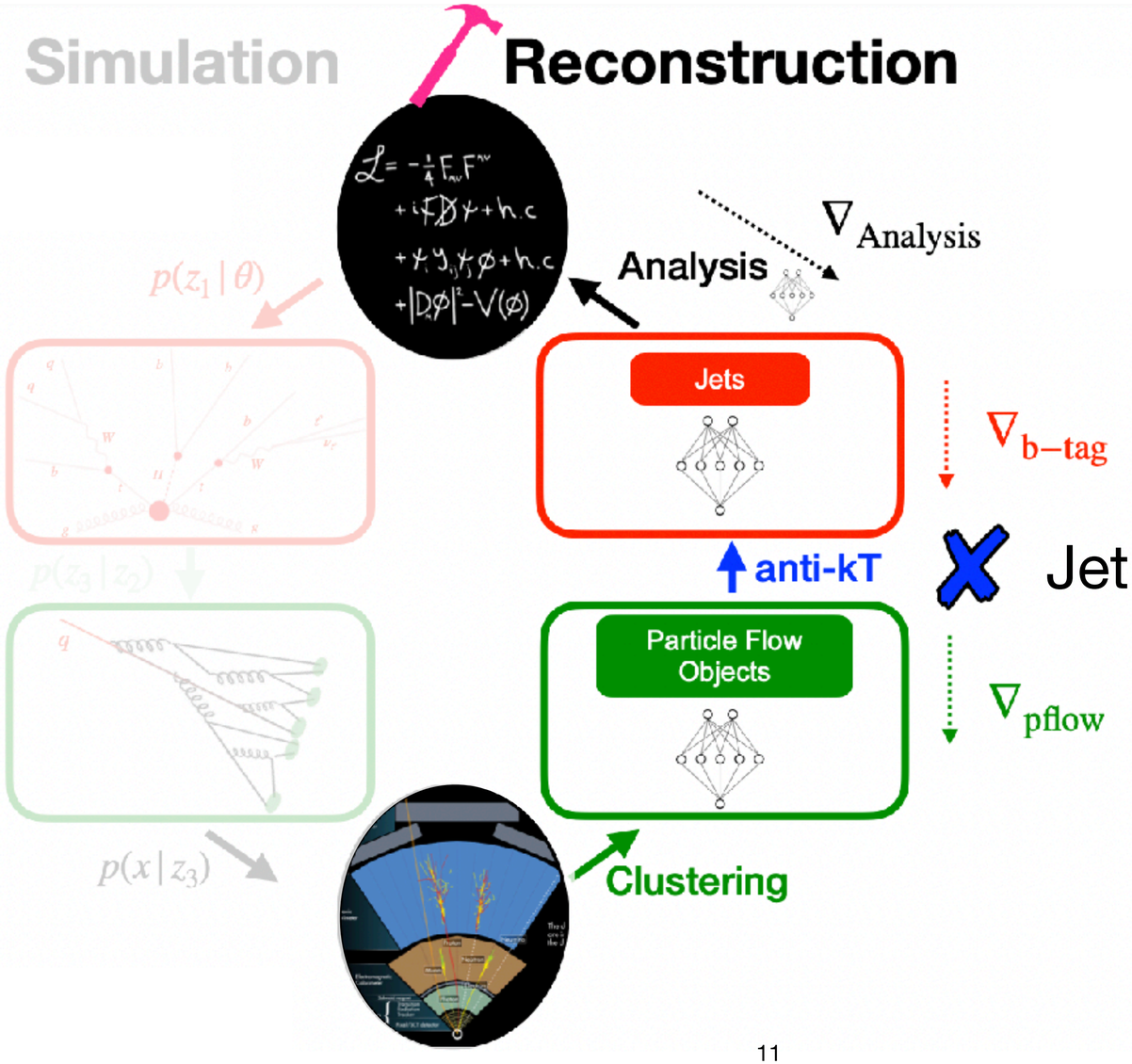
TrackFormers: In Search of Transformer-Based Particle Tracking for the High-Luminosity LHC Era

EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion

Erik Buhmann[1], Cedric Ewen[1], Darius A. Faroughy[2], Tobias Golling[3], Gregor Kasieczka[1,4], Matthew Leigh[3], Guillaume Quétant[3], John Andrew Raine[3], Debajyoti Sengupta[3], and David Shih[2]

… and MANY more

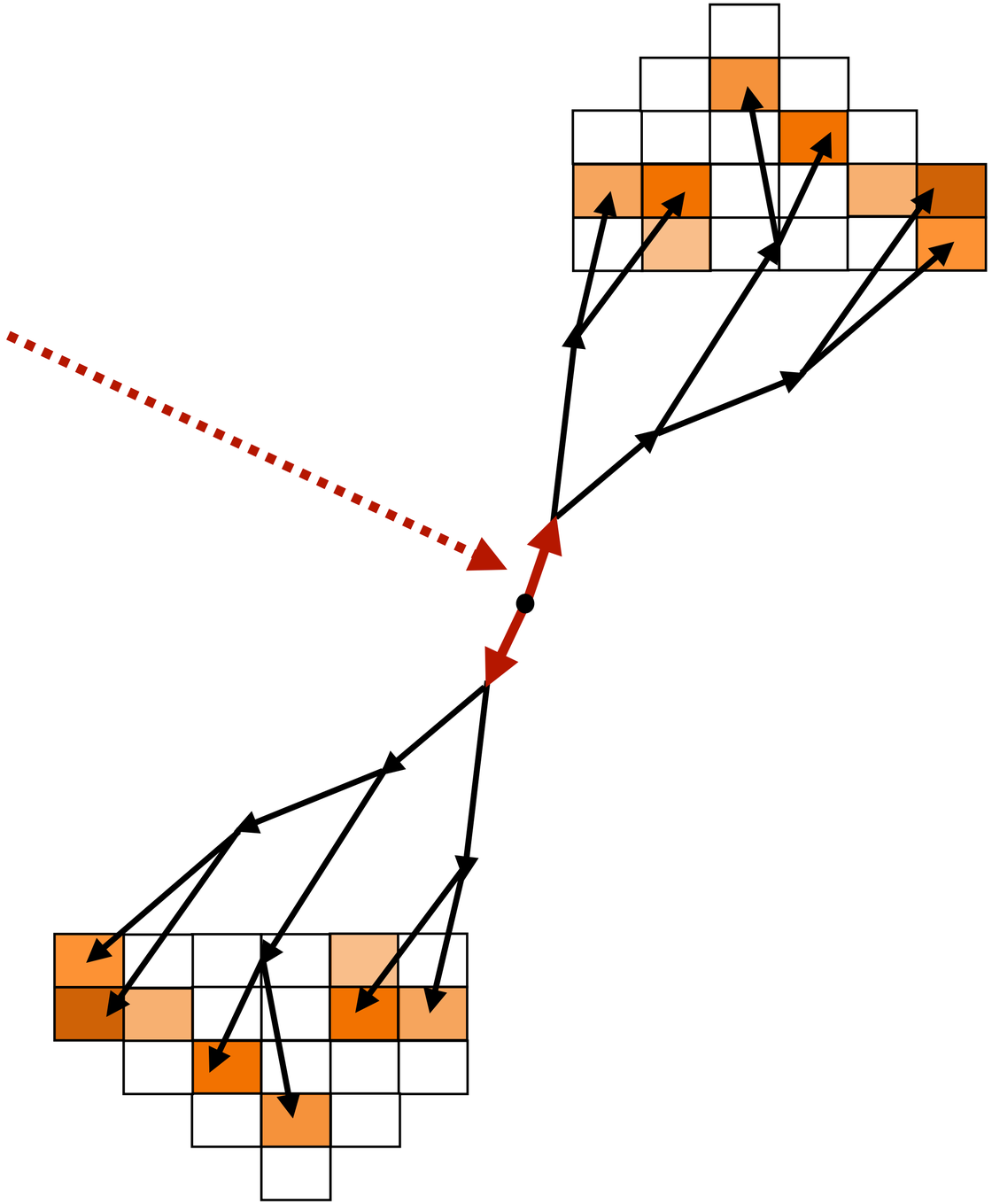
End-goal: Fully differentiable particle physics analysis



Jet clustering **not** differentiable!

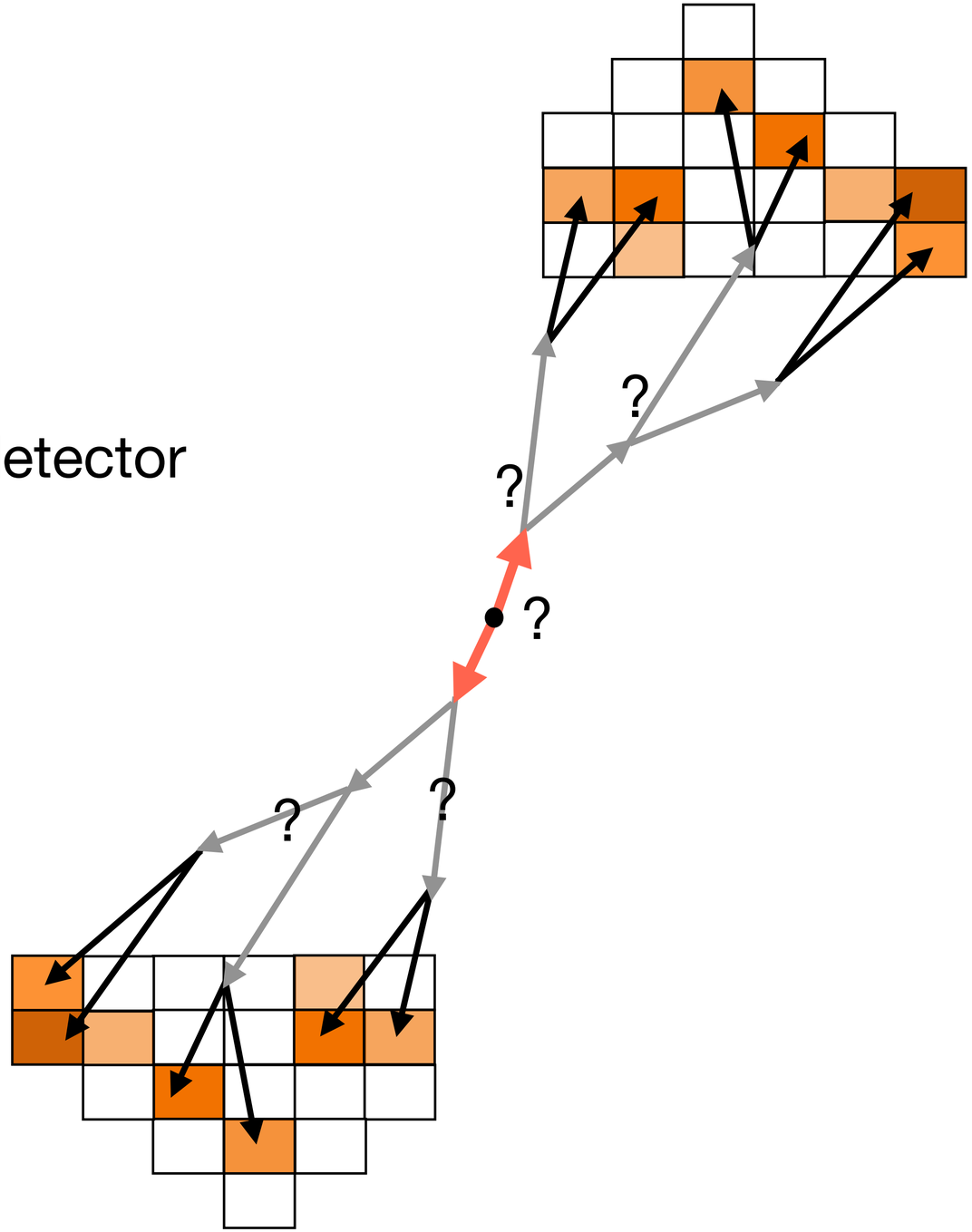
What is jet clustering?

- Goal: Information about initial particle(s)



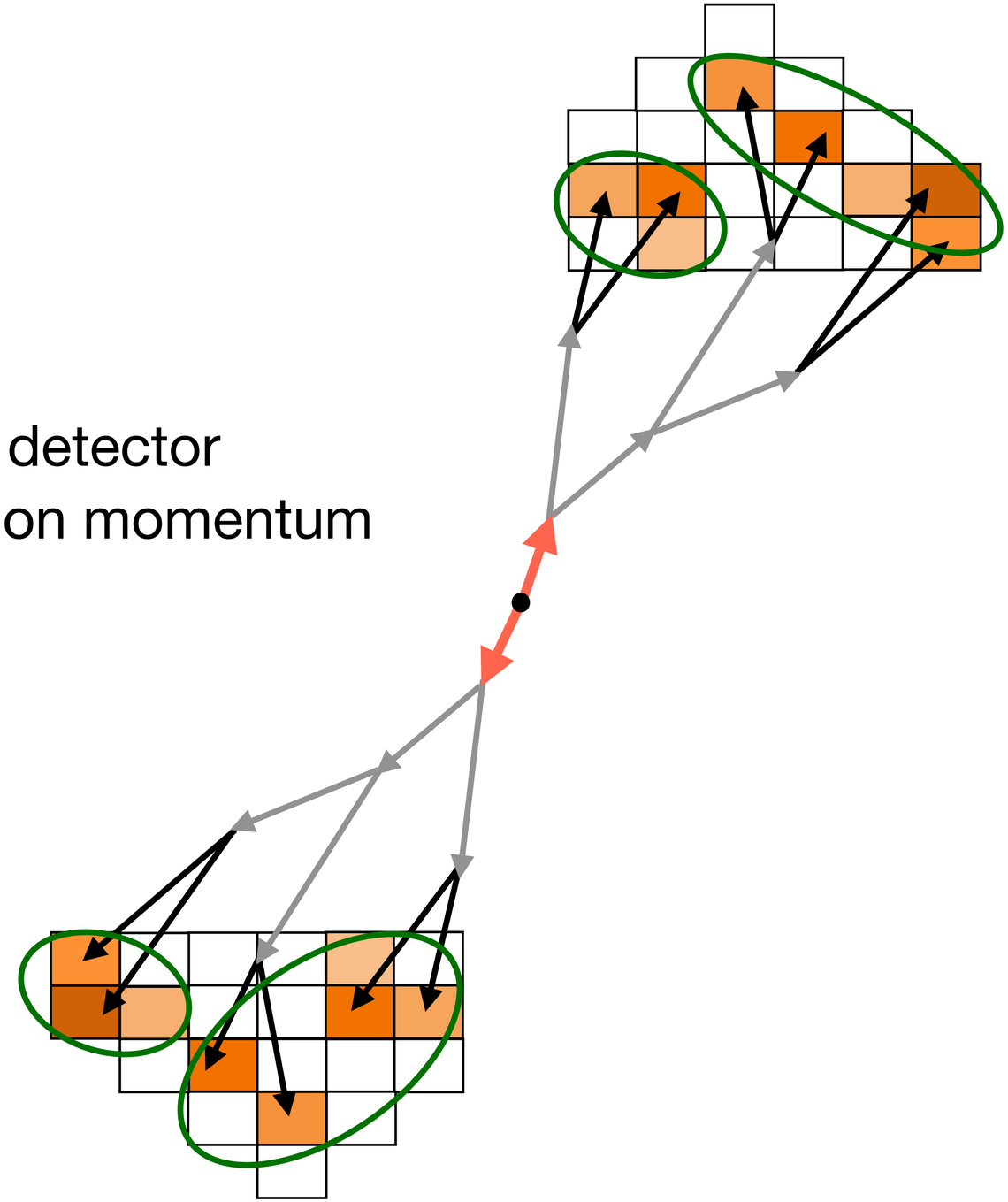
What is jet clustering?

- Goal: Information about initial particle(s)
- Problem: We only see stable particles in detector



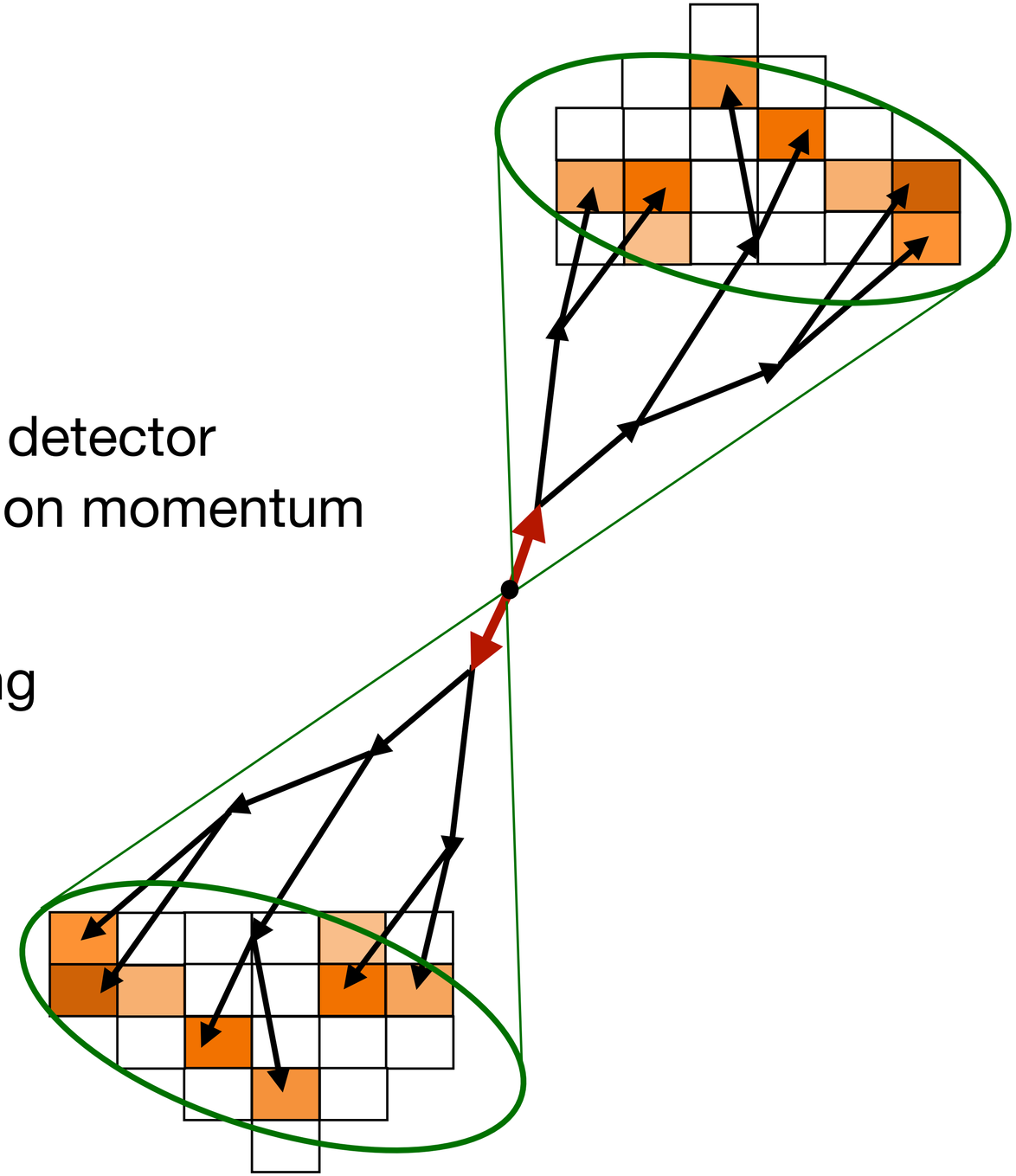
What is jet clustering?

- Goal: Information about initial particle(s)
- Problem: We only see stable particles in detector
  → iteratively cluster constituents based on momentum and distance



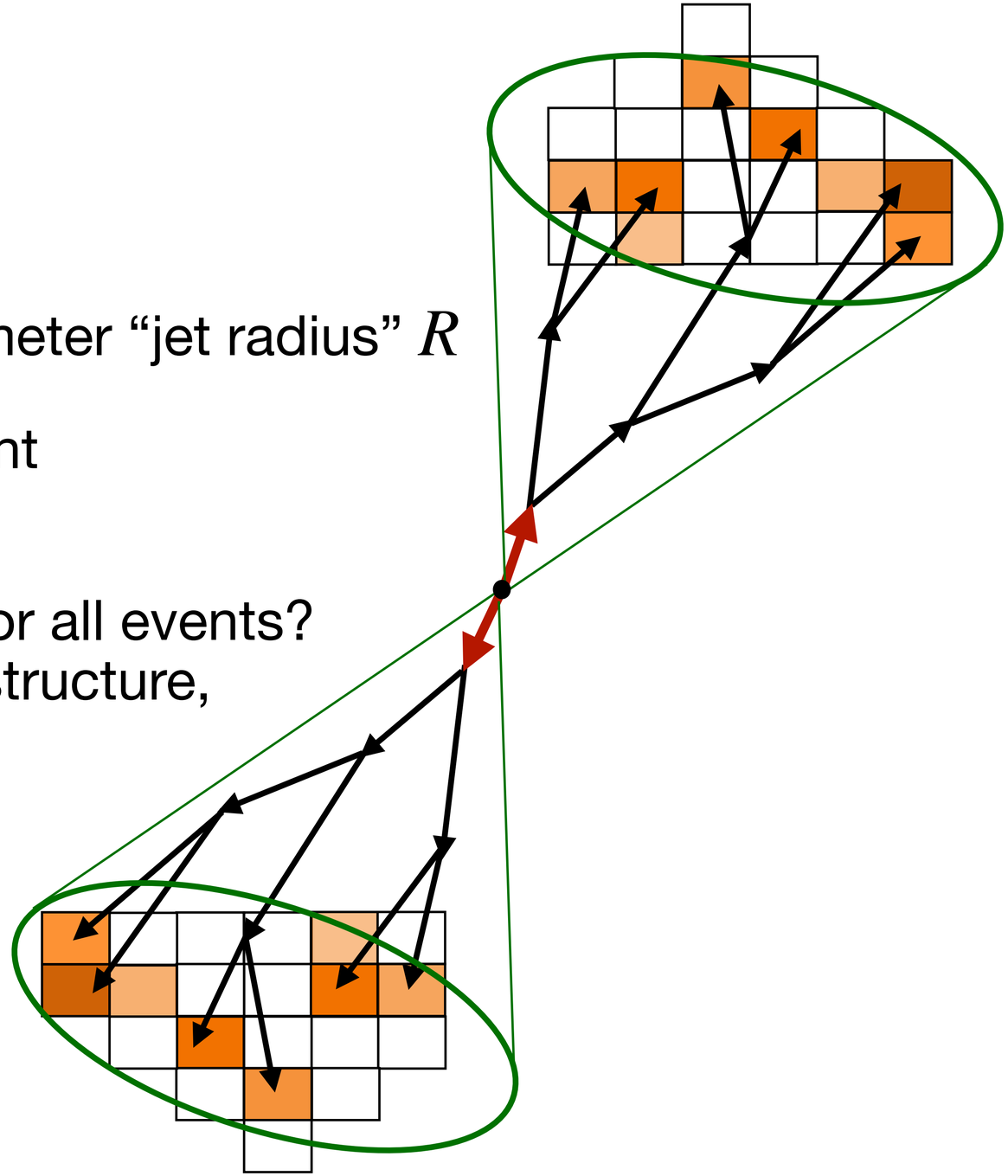
What is jet clustering?

- Goal: Information about initial particle(s)
- Problem: We only see stable particles in detector
  → iteratively cluster constituents based on momentum and distance
- Get initial momentum from final clustering



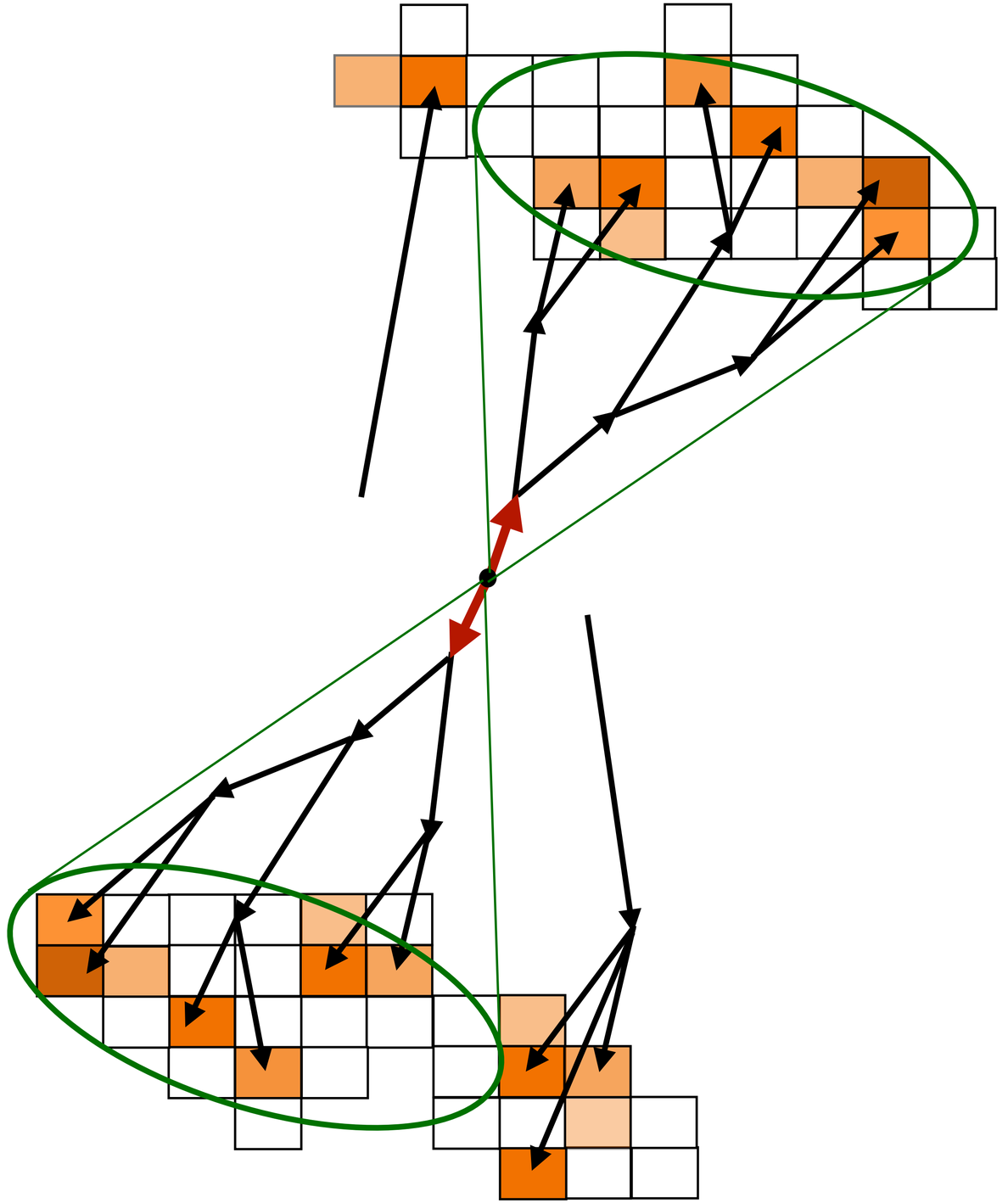
What is jet clustering?

- Problem: Clustering based on free parameter “jet radius” R
- Usually optimized once and kept constant for rest of analysis
  - But what if one jet radius is not ideal for all events? e.g. jets from different quarks, jet substructure, …



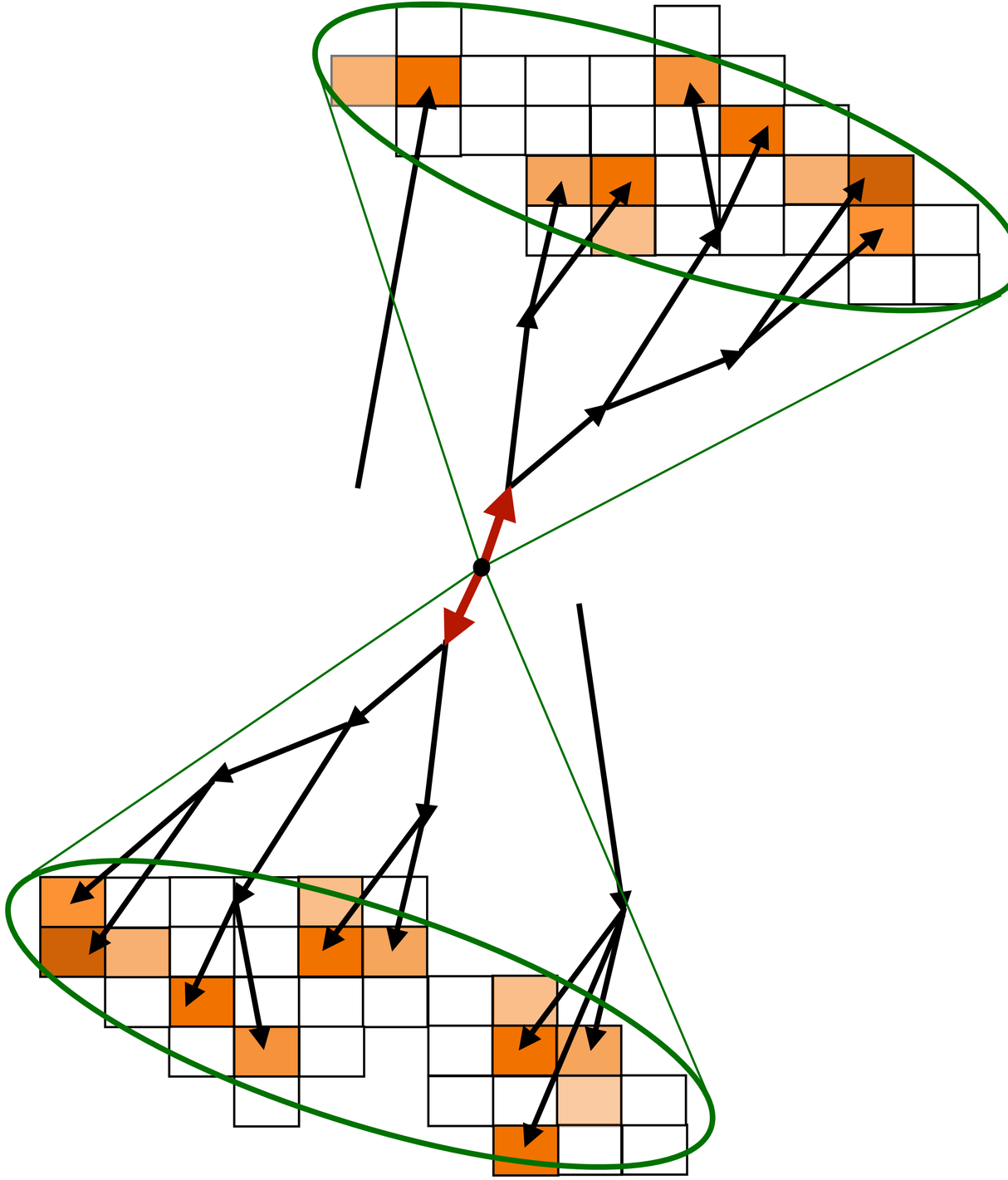
Influence of jet radius

- What does the jet radius do?
- Smaller radius



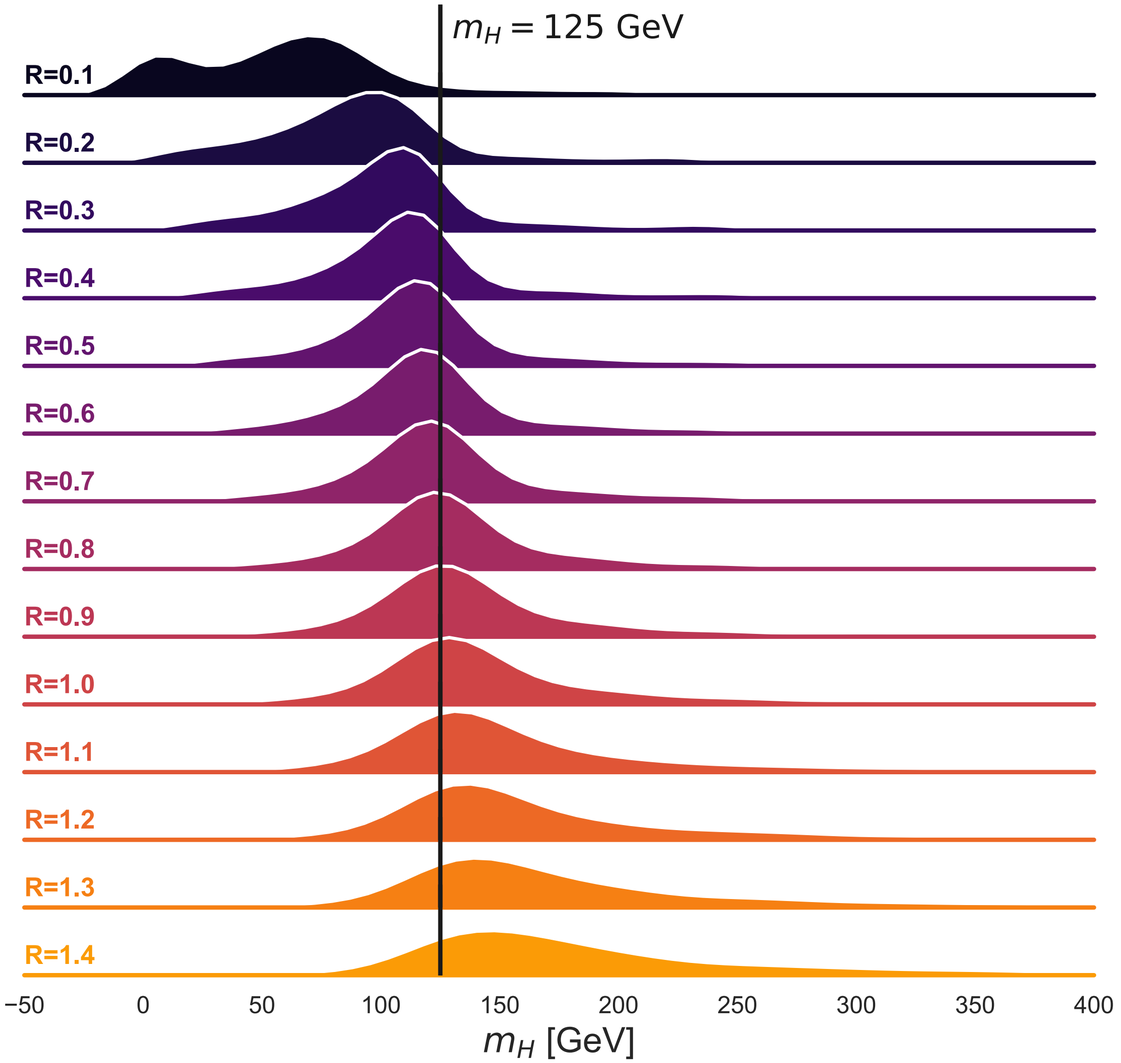
Influence of jet radius

- What does the jet radius do?
- Smaller radius vs. larger radius
- Jet radius influences information about reconstructed initial particles



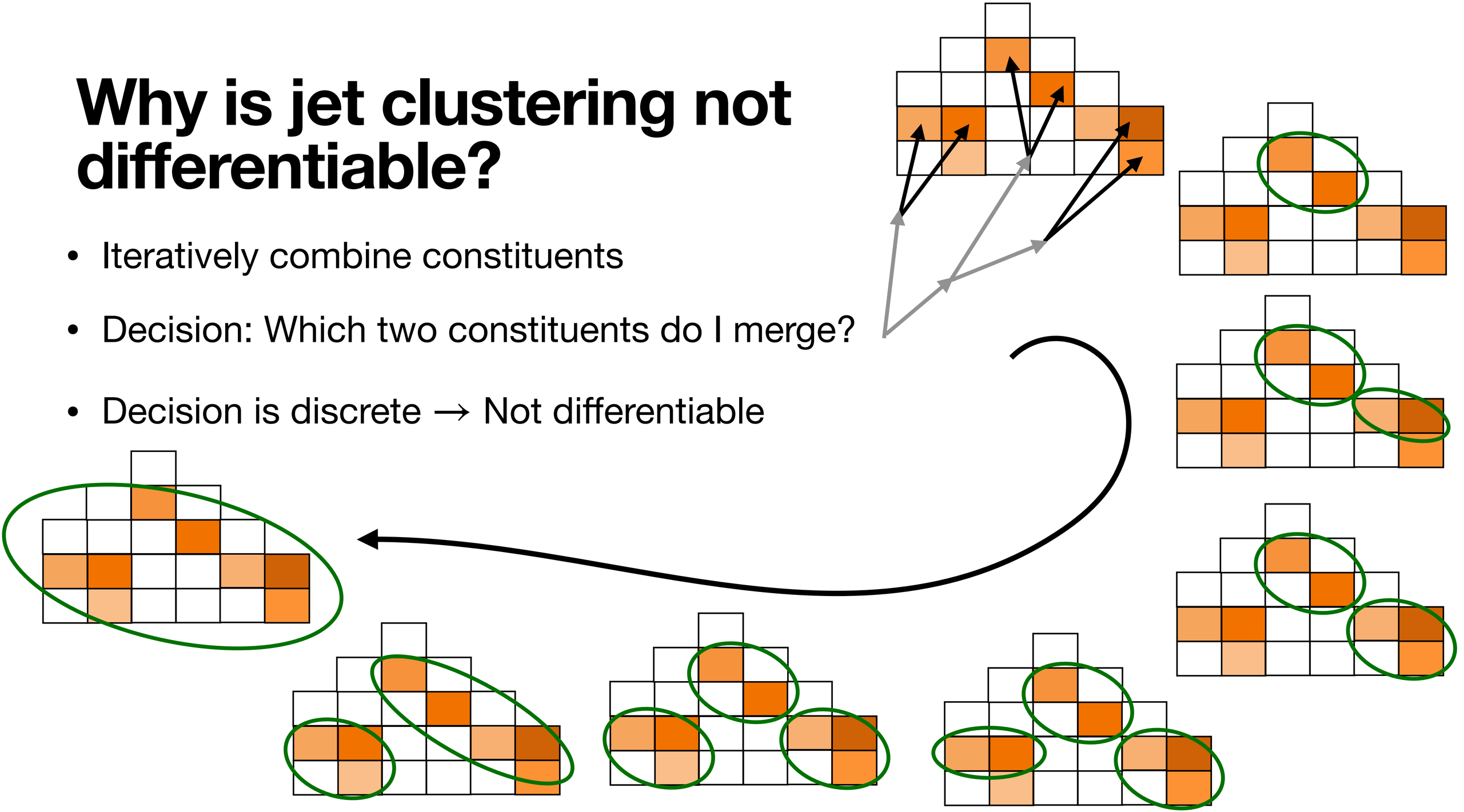
Why is it important to get the correct jet radius?

- Example: Reconstruct Higgs mass from two simulated jets
- Cluster jets with different R and compute m_H
- Position and width of resonance changes



Why is jet clustering not differentiable?

- Iteratively combine constituents
- Decision: Which two constituents do I merge?
- Decision is discrete → Not differentiable



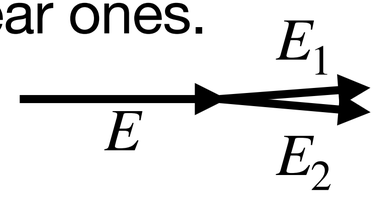
Why do we use special anti-k_T clustering?

- Different clustering algorithms in machine learning
  - K-means, DBSCAN, T-SNE, …
- Physics reason: **Infrared and collinear (IRC) safety**

The clustering result should **not** change …

… if we **add a particle with almost zero energy**. **Infrared safety**

… if **one particle splits into two** collinear ones. **Collinear safety**

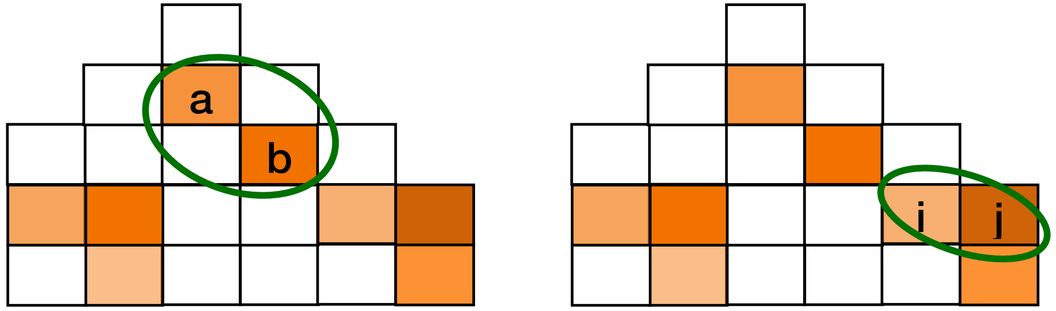


→ Physically unresolvable processes, but predicted by QFT.

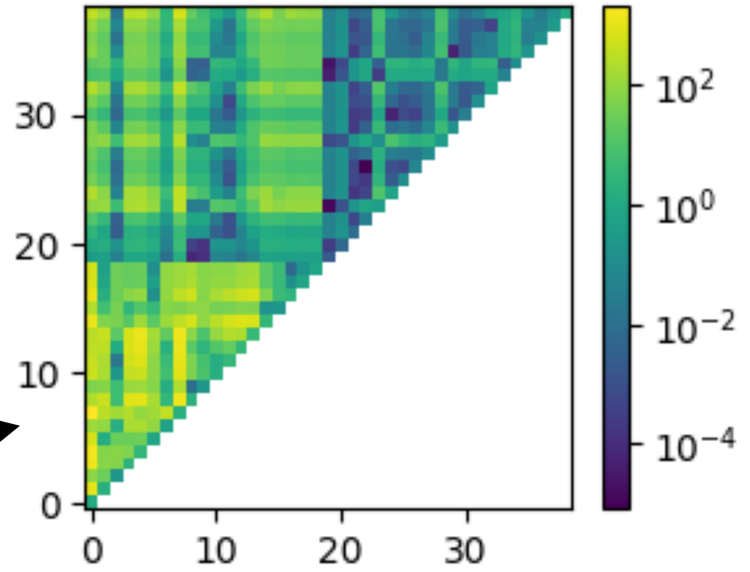
How do we decide which constituents to merge?

Anti-k_T algorithm:

- Why merge a & b and not i & j?
- Distance matrix based on constituent variables (p_T, η, ϕ)
- Off-diagonal: “distance” between all pairs of constituents
- Diagonal: constituent should be removed from clustering



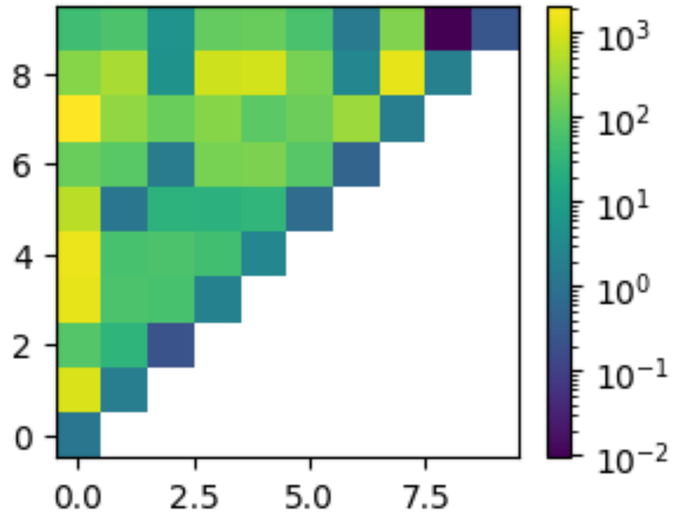
Distance matrix



39 constituents

How do subsequent clustering steps work?

Step 1



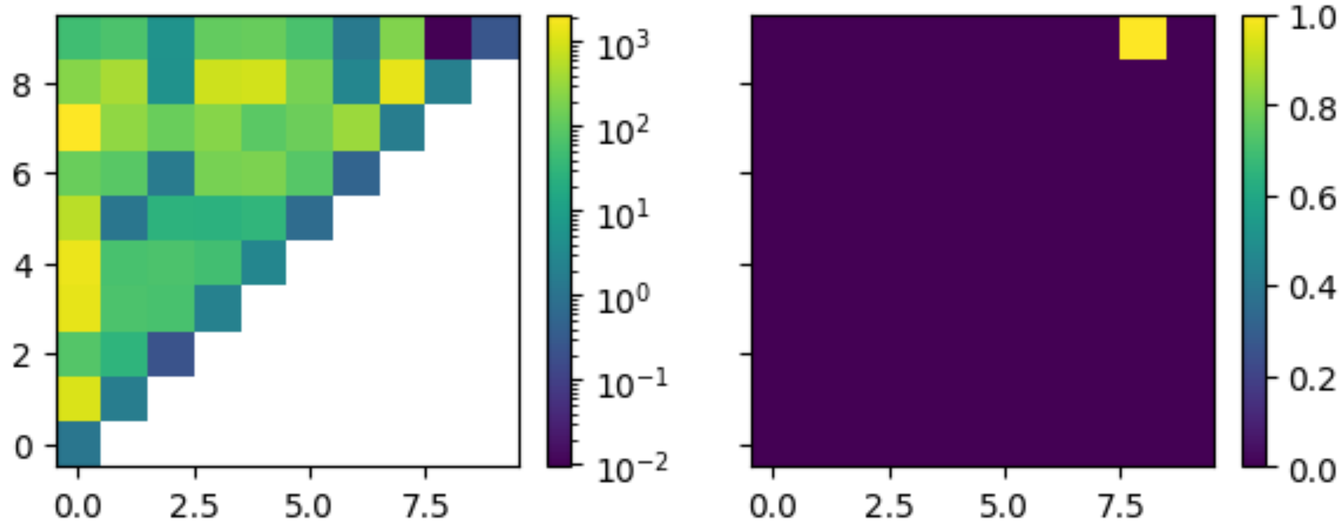
Distance matrix

Compute $\min(D)$ (deterministic)

8 constituents

How do subsequent clustering steps work?

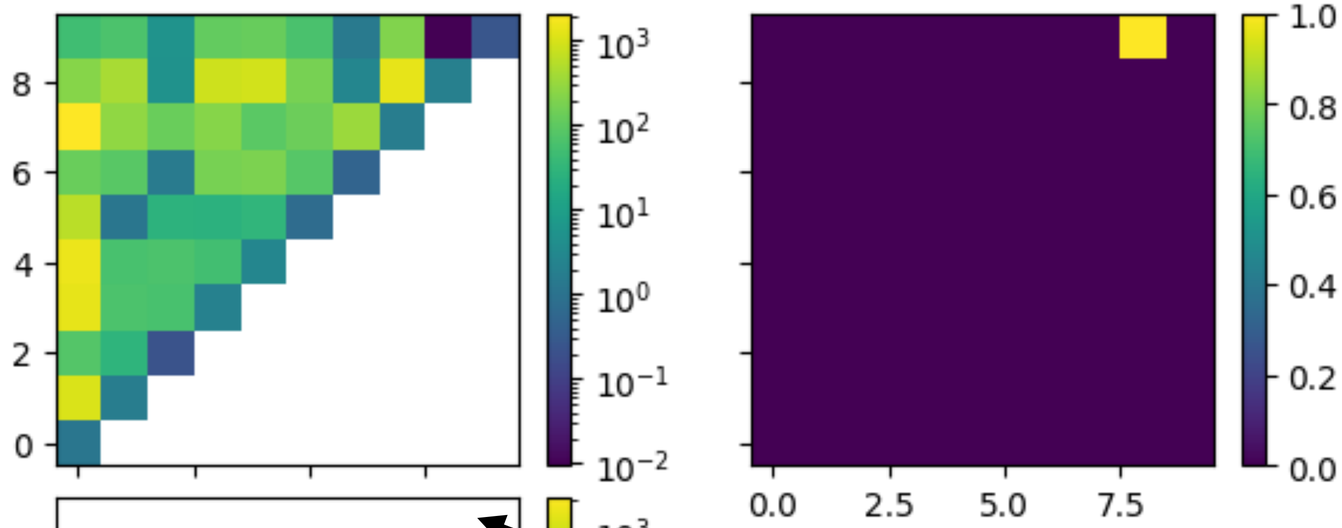
Step 1



Merge constituents
8 and 9

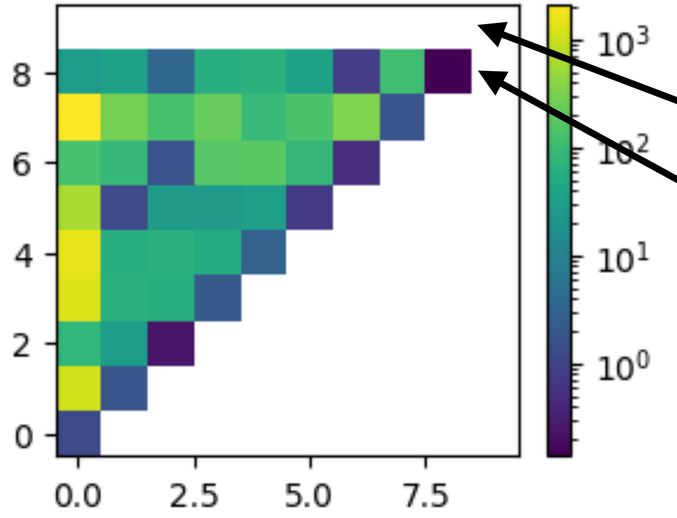
How do subsequent clustering steps work?

Step 1



Merge constituents
8 and 9

Step 2

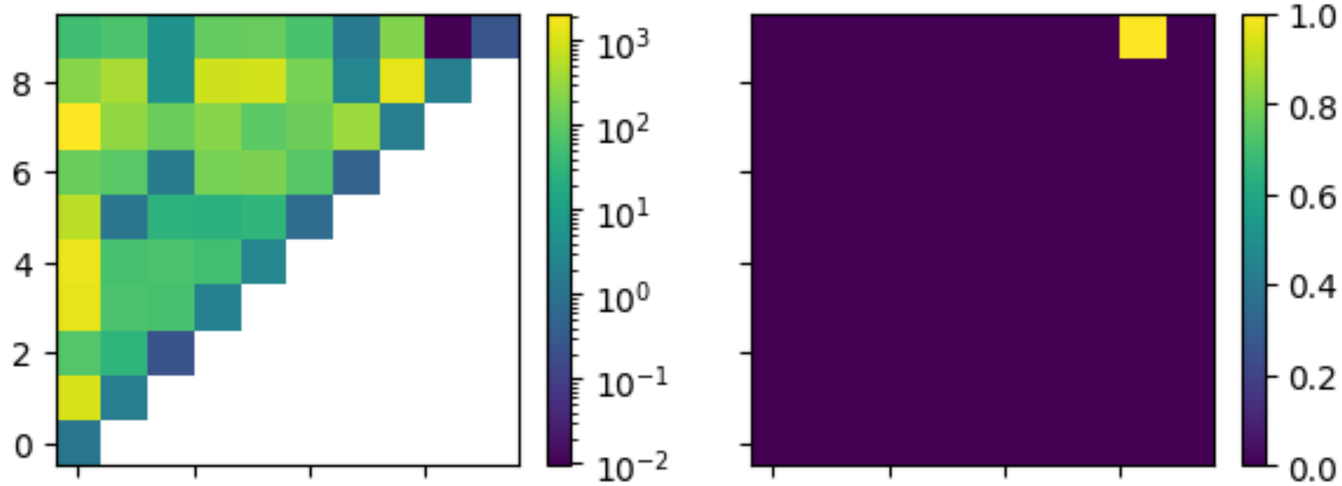


Constituent 9 removed

Constituent 8 updated

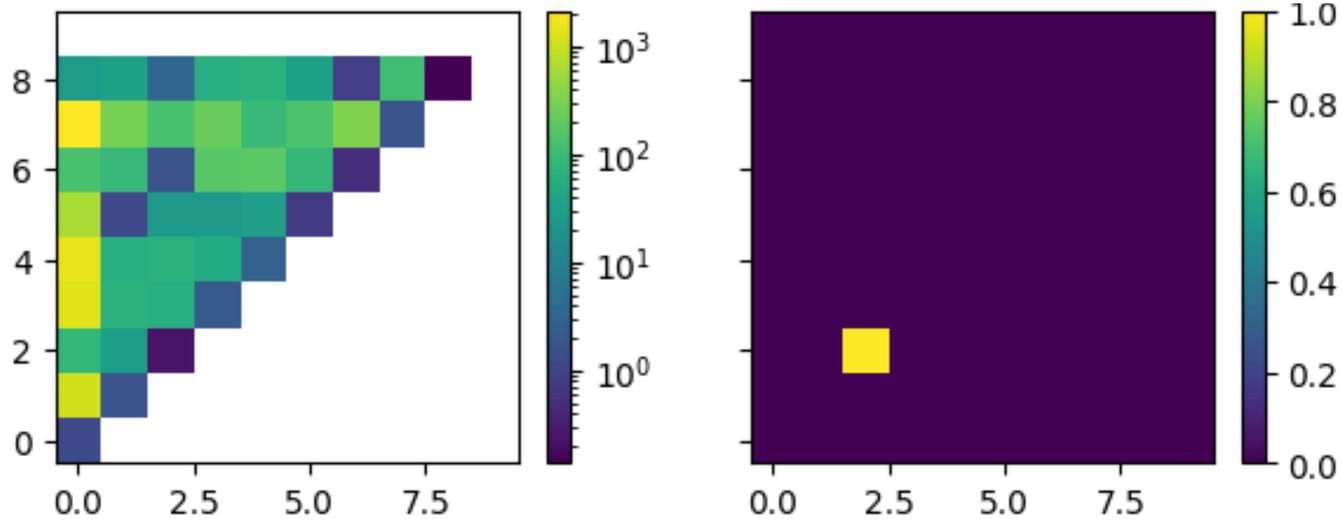
How do subsequent clustering steps work?

Step 1



Merge constituents
8 and 9

Step 2



Remove constituent 2
from clustering

...

Gradients of deterministic jet clustering

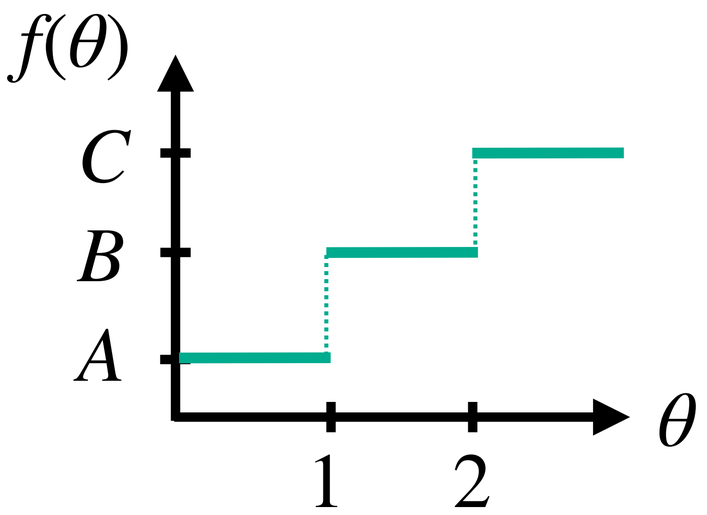
Toy example with
3 options:

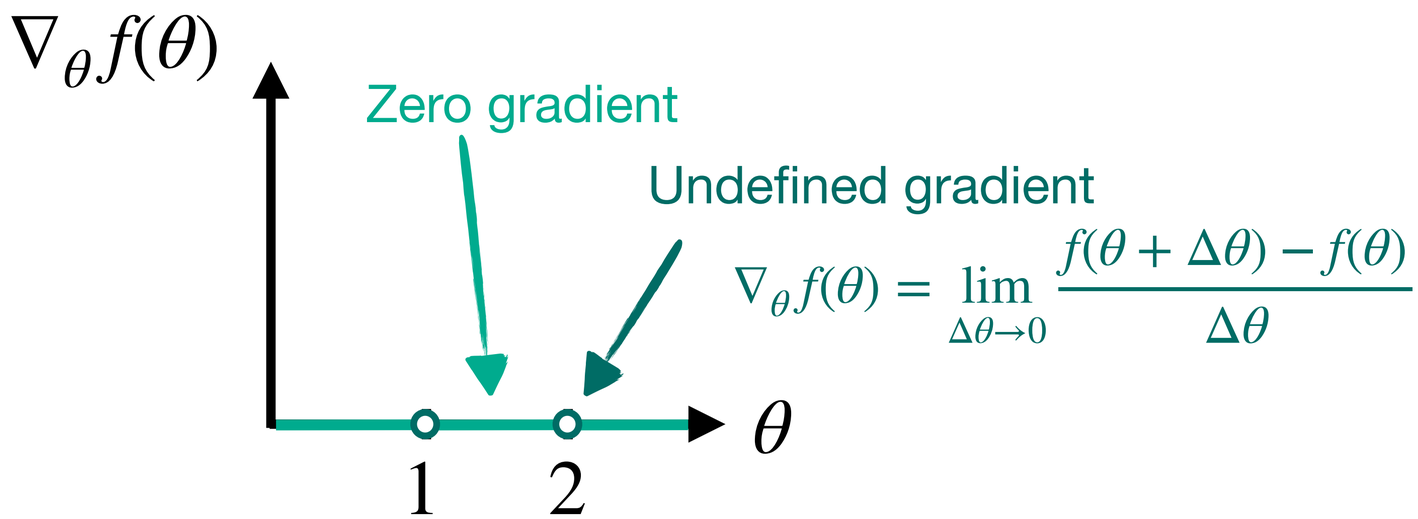
$$f(\theta) = \begin{cases} A & \text{if } \theta < 1 \\ B & \text{if } 1 \leq \theta \leq 2 \\ C & \text{if } \theta > 2 \end{cases}$$

Merge constituents (1, 2)

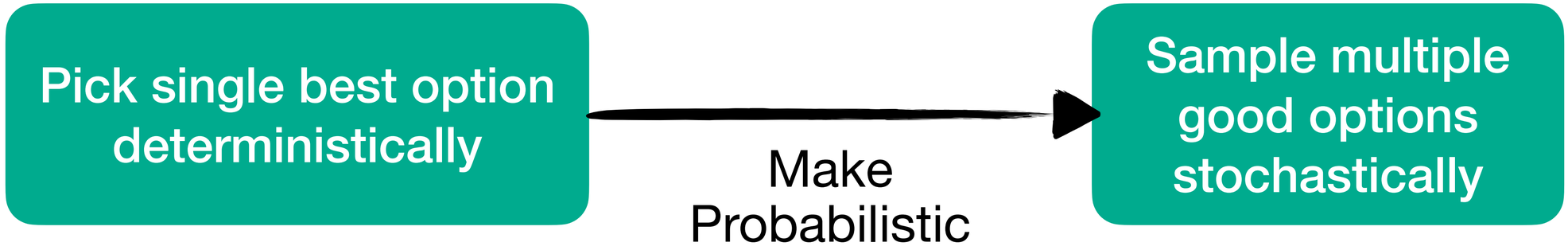
Merge constituents (2, 3)

Remove constituent 3





Reinterpret decision as stochastic



Deterministic

Single output

$$f(\theta)$$

Stochastic

Multiple outputs
→ expectation value

$$f(\theta) = \mathbb{E}_{k \sim p_\theta(k)}[f(k)]$$

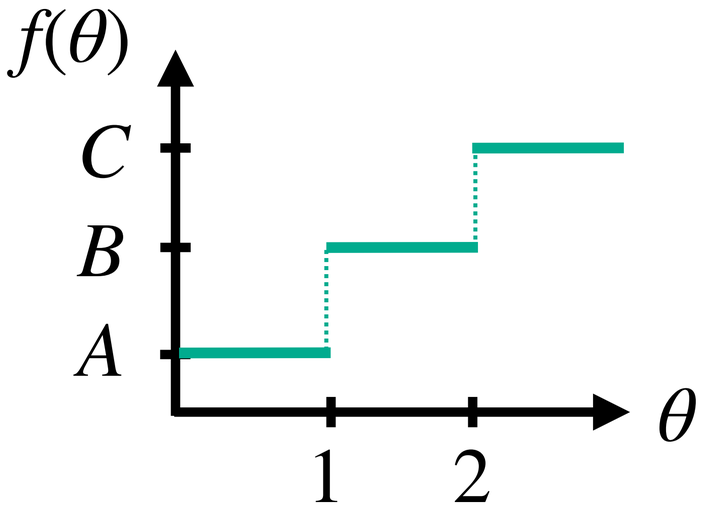
$$= \sum_{k \in \{A,B,C\}} p_\theta(k) f(k)$$

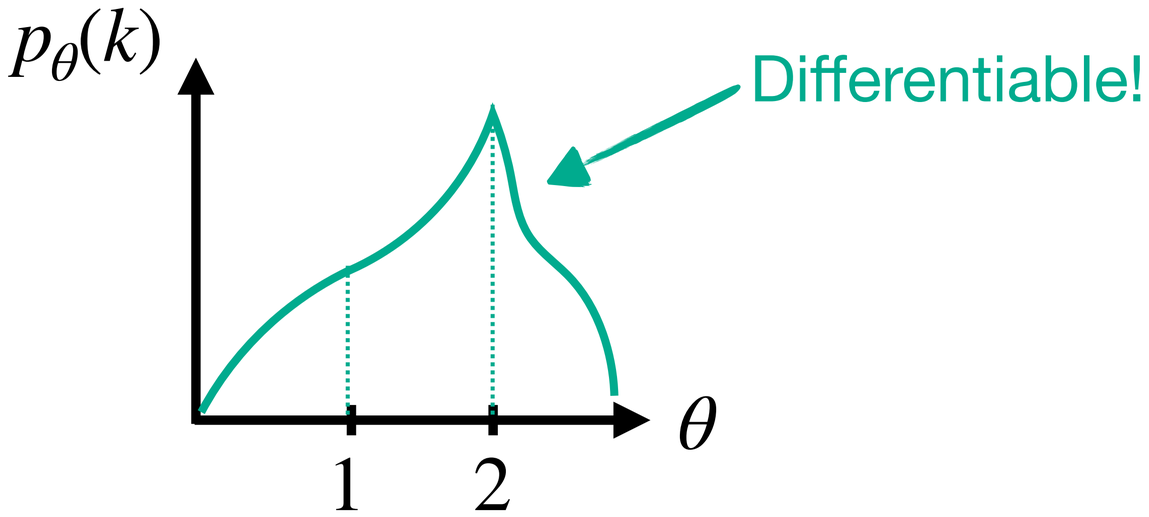
Why is a stochastic decision differentiable?

$$f(\theta) = \mathbb{E}_{k \sim p_\theta(k)}[f(k)]$$
$$= \sum_{k \in \{A,B,C\}} p_\theta(k) f(k)$$

Define probability:

$$p_\theta(k) = \frac{\exp(l_k(\theta))}{\exp(l_A(\theta)) + \exp(l_B(\theta)) + \exp(l_C(\theta))}$$

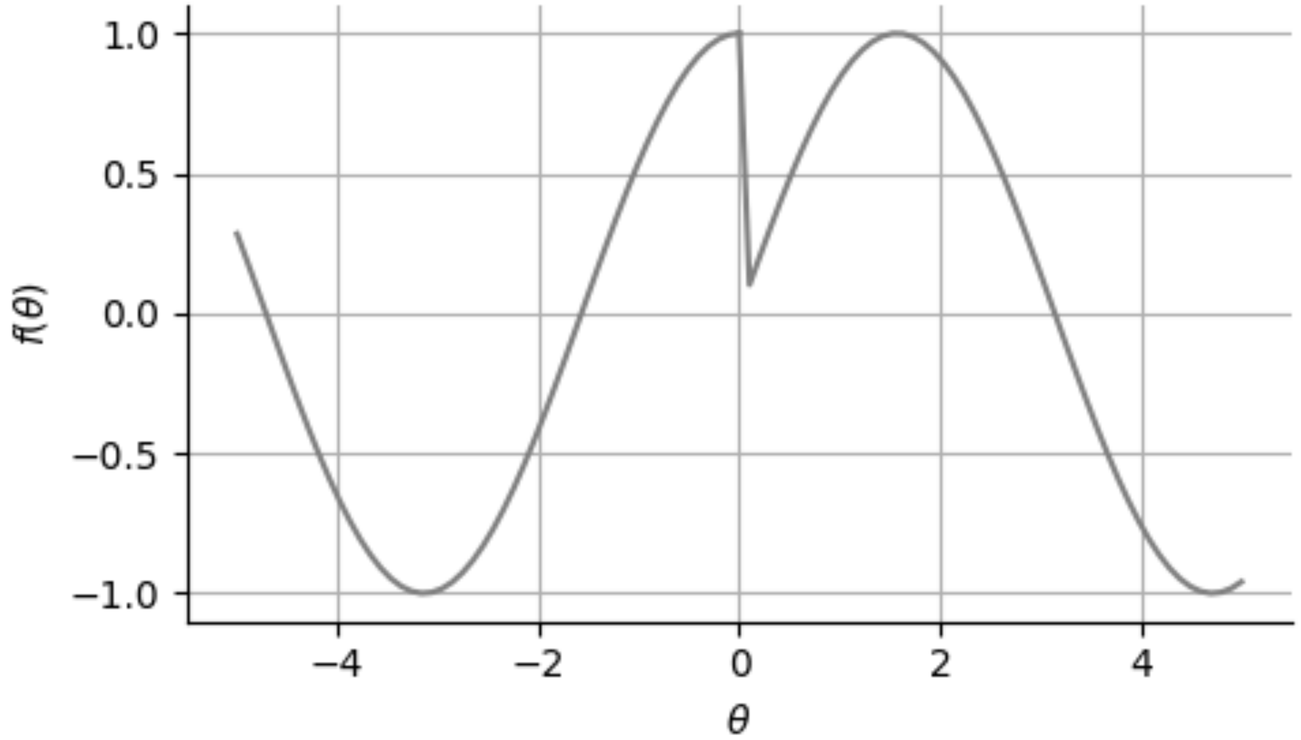




Illustrative example

Concrete example: Sin&Cos

$$f(\theta) = \begin{cases} \cos(\theta) & \text{if } \theta < 0 \\ \sin(\theta) & \text{if } \theta \geq 0 \end{cases}$$

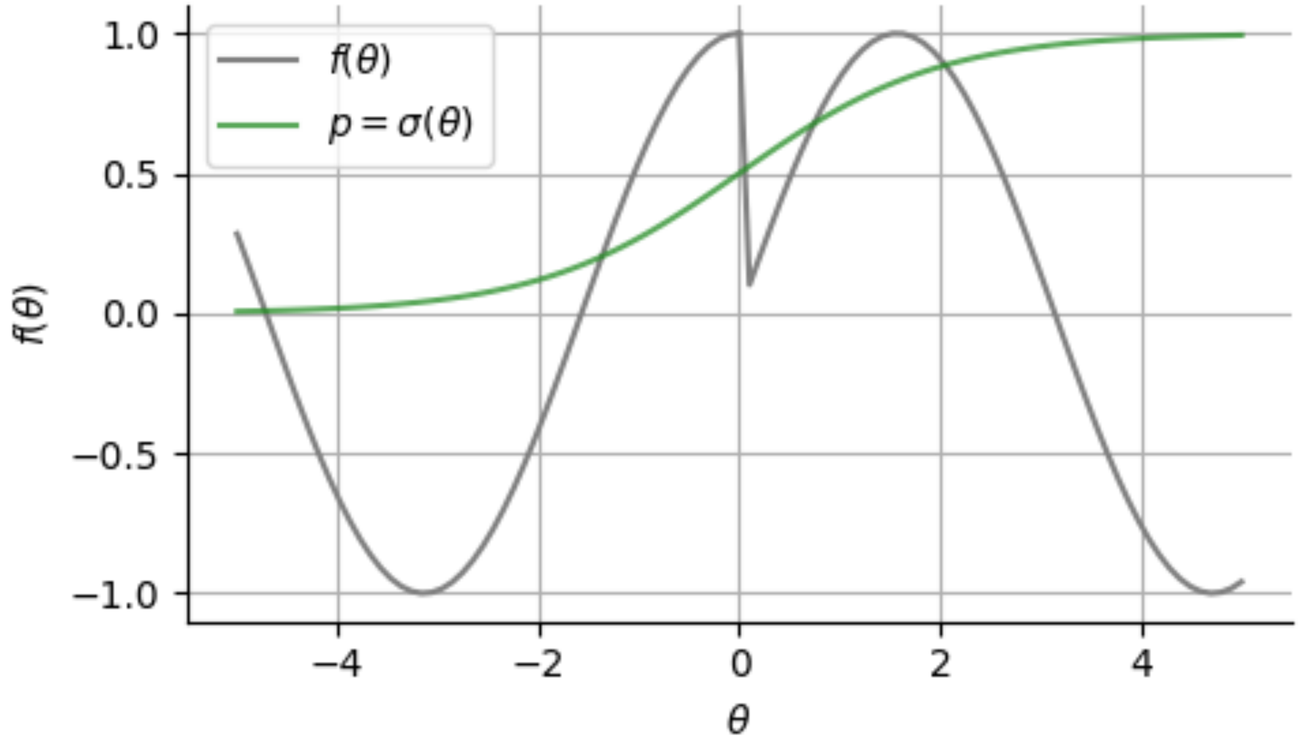


Concrete example: Sin&Cos

$$f(\theta) = \begin{cases} \cos(\theta) & \text{if } \theta < 0 \\ \sin(\theta) & \text{if } \theta \geq 0 \end{cases}$$

Make probabilistic:

$$p = \sigma(\theta) = \frac{1}{1 + \exp(-\theta)}$$



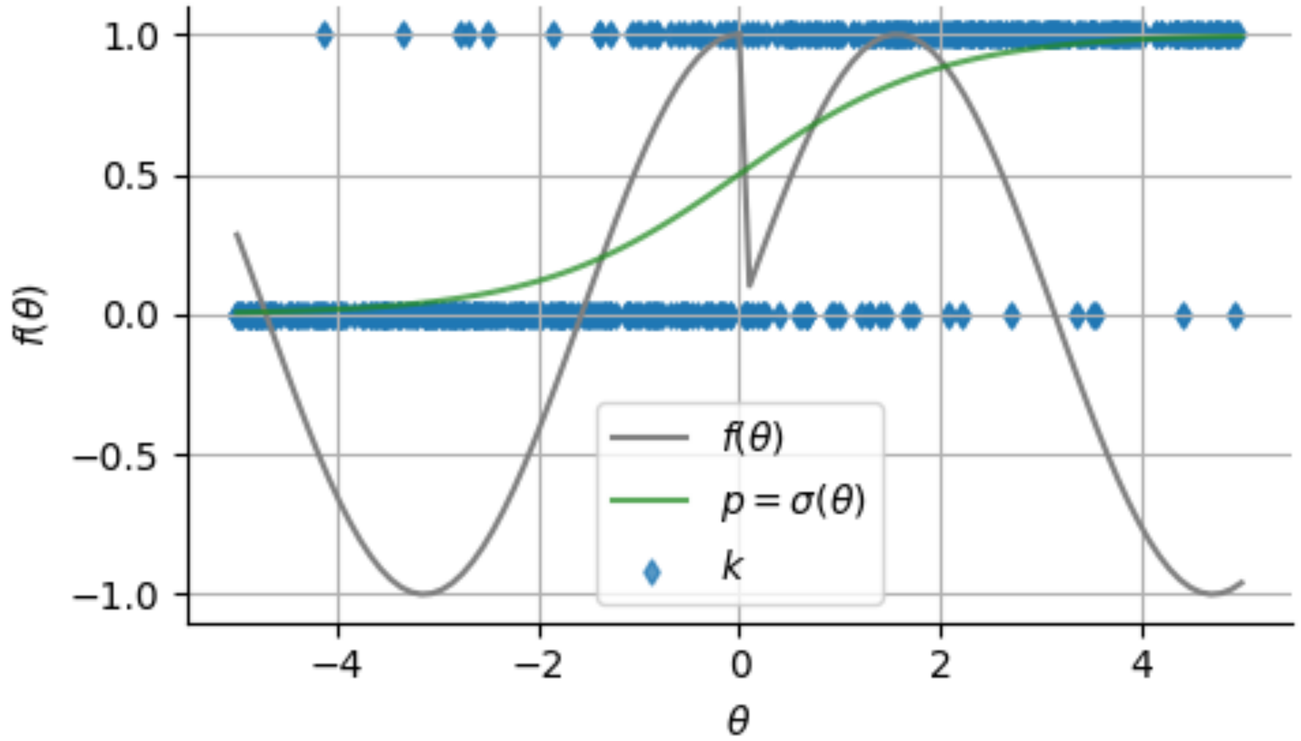
Concrete example: Sin&Cos

$$f(\theta) = \begin{cases} \cos(\theta) & \text{if } \theta < 0 \\ \sin(\theta) & \text{if } \theta \geq 0 \end{cases}$$

Make probabilistic:

$$p = \sigma(\theta) = \frac{1}{1 + \exp(-\theta)}$$

$$k \sim \text{Ber}(k; \theta) = \begin{cases} p & \text{if } k = 1 \\ 1 - p & \text{if } k = 0 \end{cases}$$



Concrete example: Sin&Cos

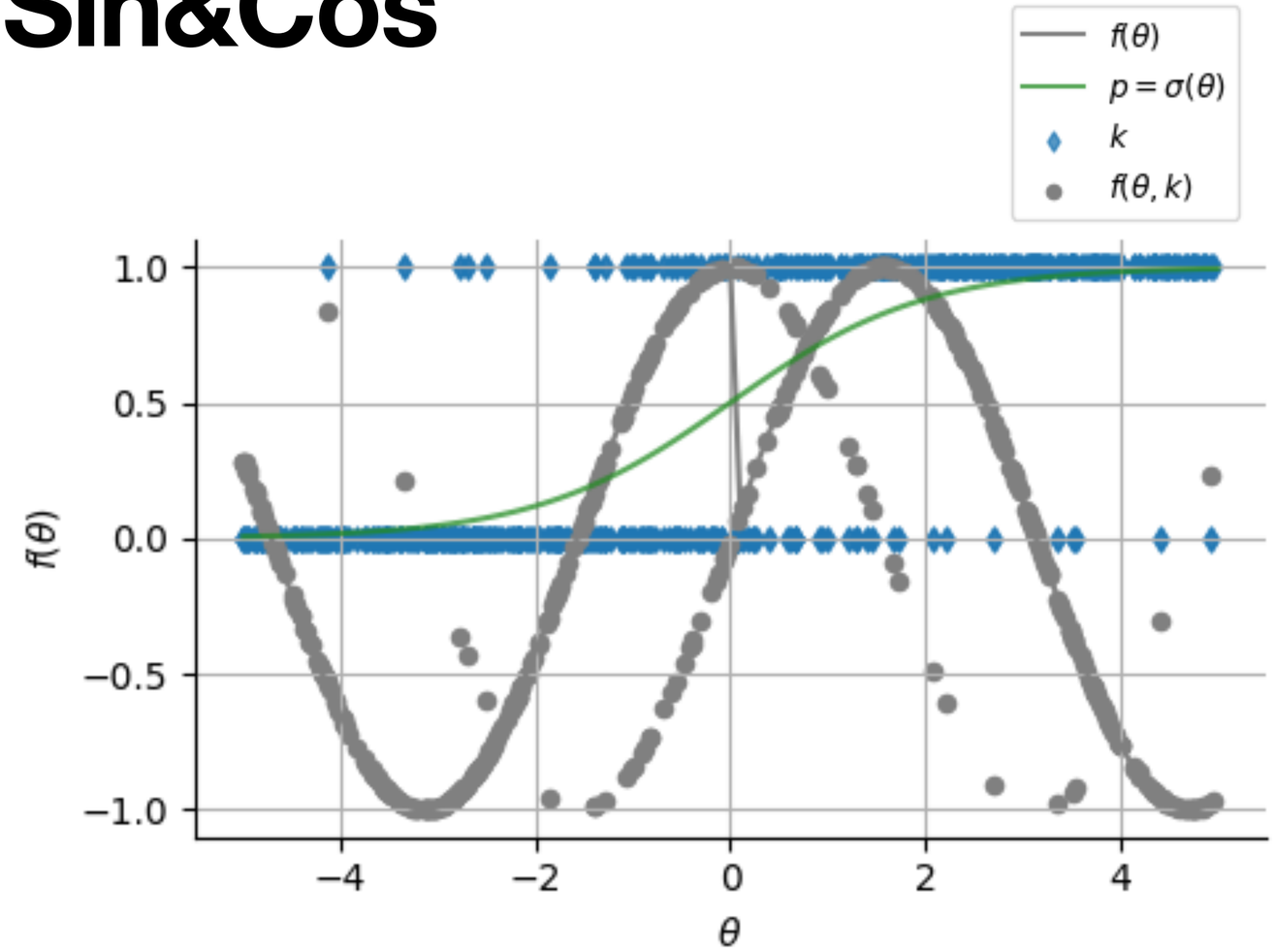
$$f(\theta) = \begin{cases} \cos(\theta) & \text{if } \theta < 0 \\ \sin(\theta) & \text{if } \theta \geq 0 \end{cases}$$

Make probabilistic:

$$p = \sigma(\theta) = \frac{1}{1 + \exp(-\theta)}$$

$$k \sim \text{Ber}(k; \theta) = \begin{cases} p & \text{if } k = 1 \\ 1 - p & \text{if } k = 0 \end{cases}$$

$$f(\theta, k) = \begin{cases} \sin(\theta) & \text{if } k = 1 \\ \cos(\theta) & \text{if } k = 0 \end{cases}$$



Concrete example: Sin&Cos

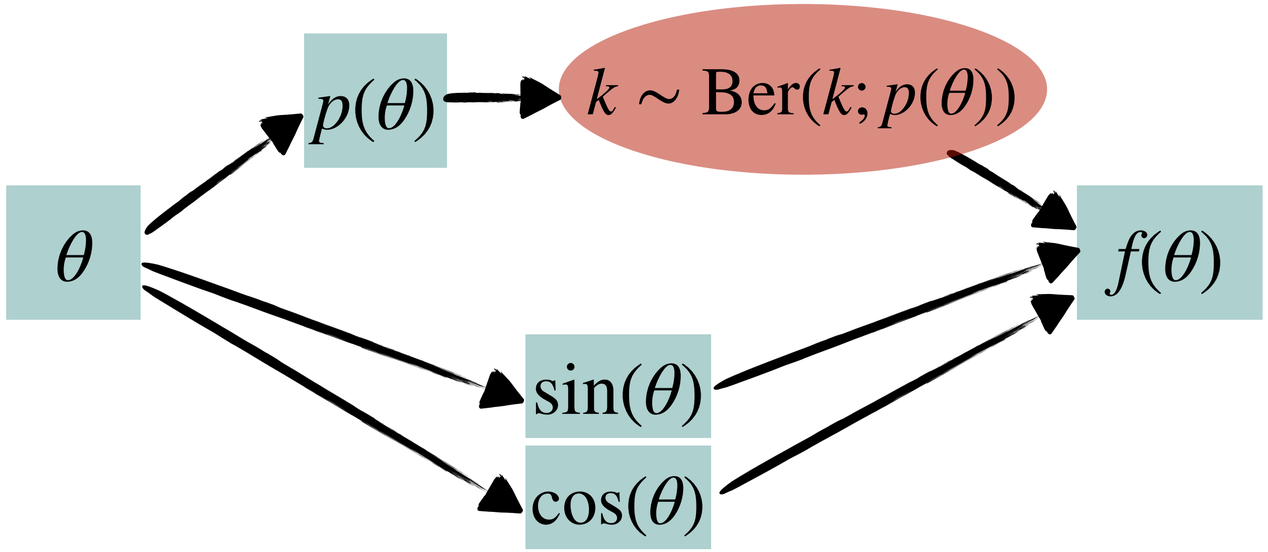
$$f(\theta) = \begin{cases} \cos(\theta) & \text{if } \theta < 0 \\ \sin(\theta) & \text{if } \theta \geq 0 \end{cases}$$

Make probabilistic:

$$p = \sigma(\theta) = \frac{1}{1 + \exp(-\theta)}$$

$$k \sim \text{Ber}(k; \theta) = \begin{cases} p & \text{if } k = 1 \\ 1 - p & \text{if } k = 0 \end{cases}$$

$$f(\theta, k) = \begin{cases} \sin(\theta) & \text{if } k = 1 \\ \cos(\theta) & \text{if } k = 0 \end{cases}$$

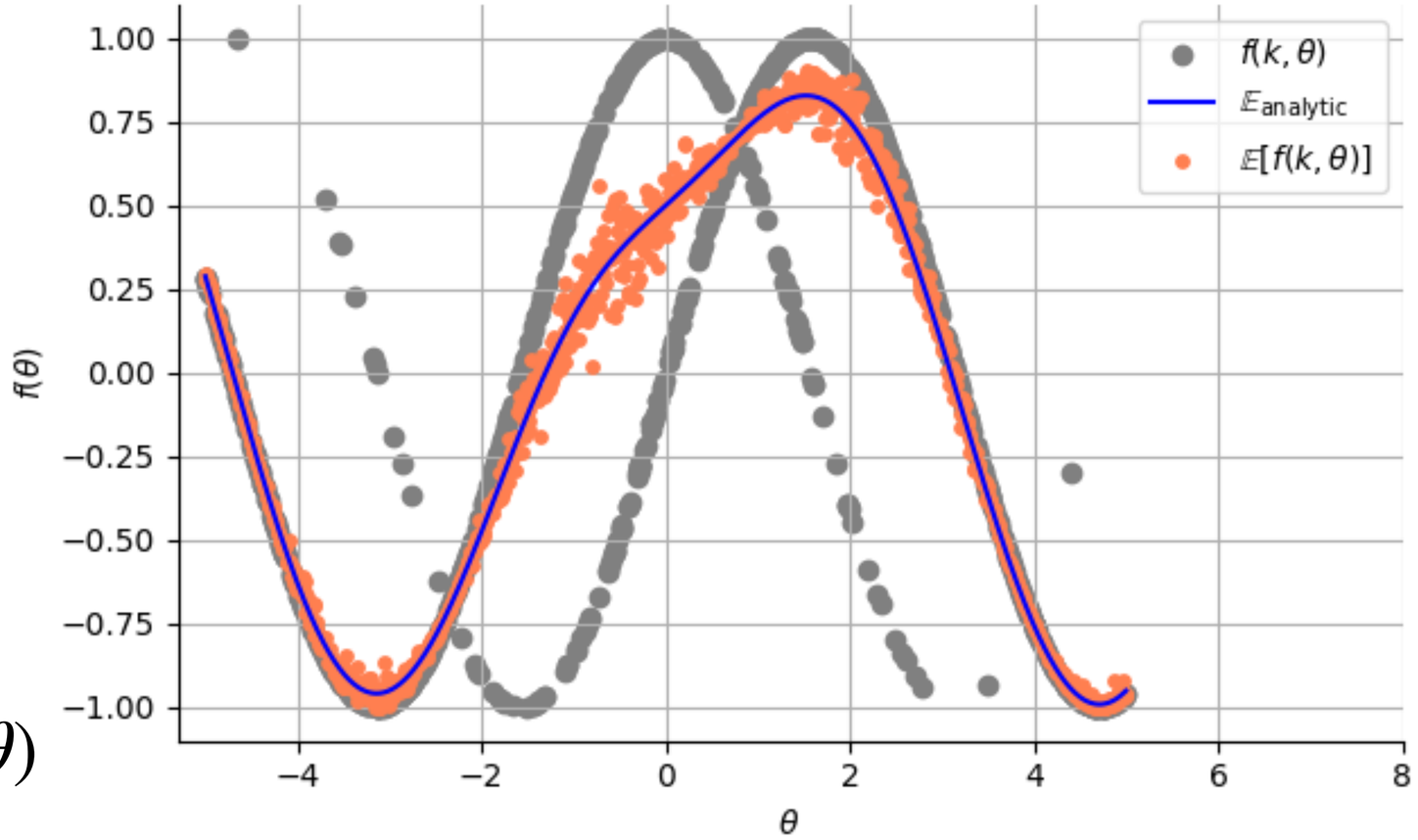


Concrete example: Sin&Cos

Calculate expectation value:

$$f(\theta) = \mathbb{E}_{k\sim p_\theta(k)}[f(k)]$$
$$= \sum_{k\in\{0,1\}} p_\theta(k) f(k)$$

$$\mathbb{E}_{\text{analytical}} = \sigma(\theta)\cdot\sin(\theta)$$
$$+(1-\sigma(\theta))\cdot\cos(\theta)$$



Back to jet clustering …

Step 1: Making jet clustering probabilistic

Probabilistic jet clustering

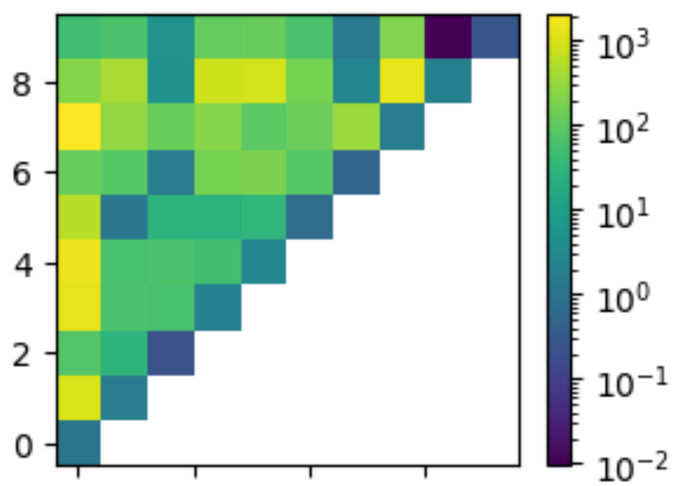
- Nothing new: Kahawala et al. 2013, arXiv:1304.2394
- Qanti-k_T:
  - Sample index pair $k = (i, j)$ based on distance matrix D:

$$k \sim p_R(k \mid e) = \frac{w_k}{\sum_{k'} w_{k'}} \qquad \text{with} \qquad w_k = \exp\left(-\alpha \frac{D_k - D_{k,\min}}{D_{k,\min}}\right)$$

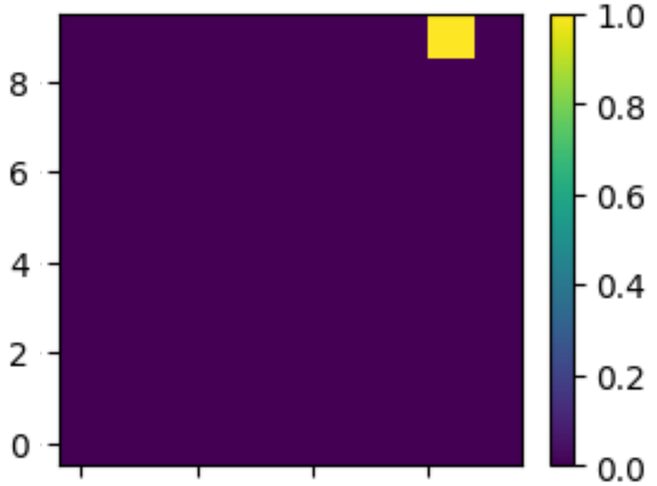
  - $\alpha \to \infty$ recovers deterministic anti-k_T algorithm
  - Preserves IRC safety

Sampling in every step

Step 1

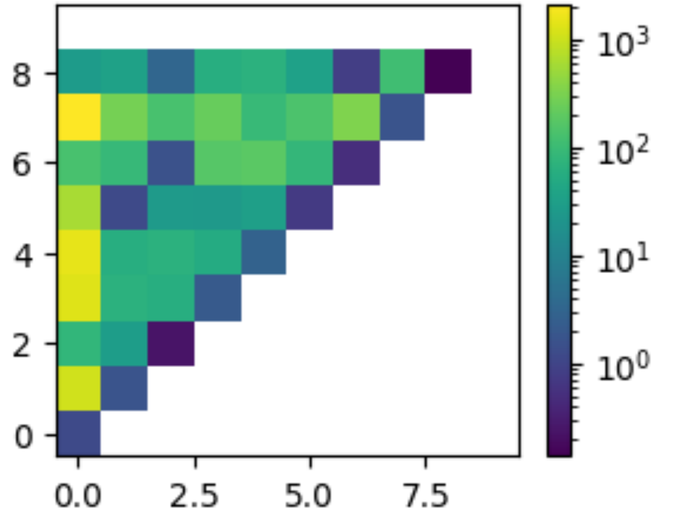


$\sim$

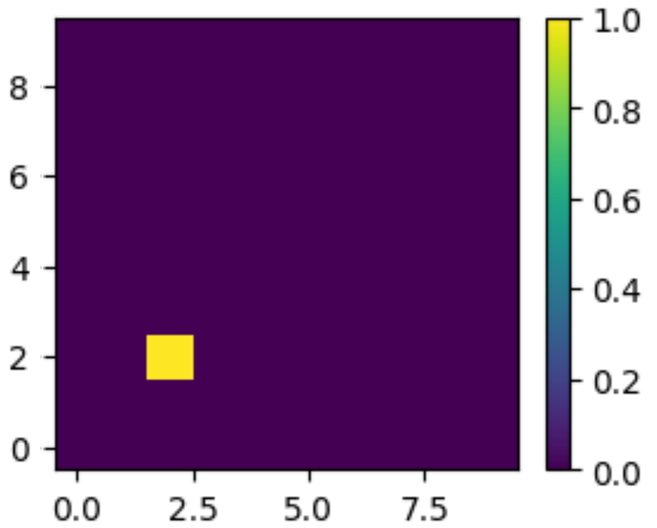


Merge constituents 8 and 9

Step 2



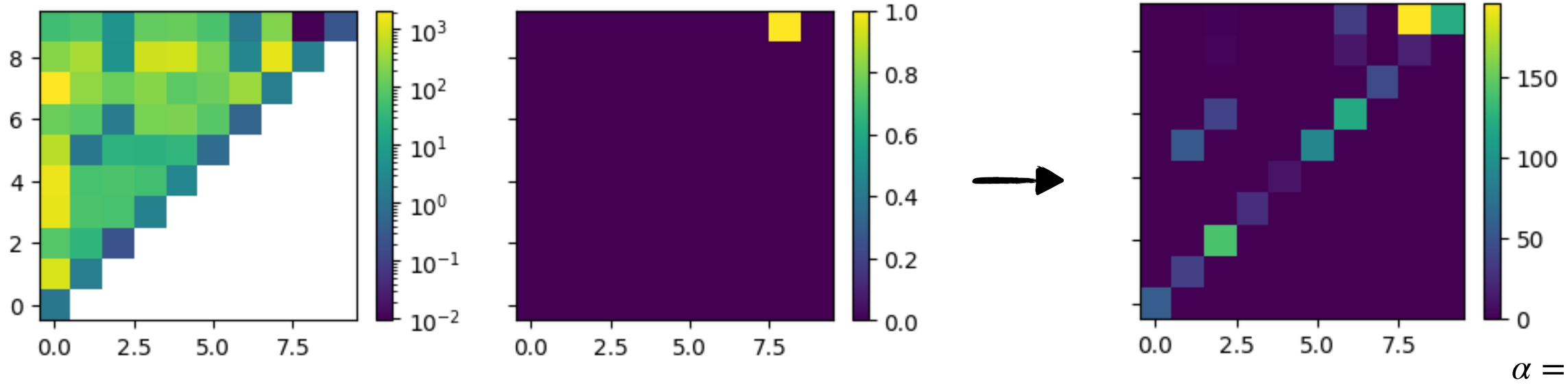
$\sim$



Remove constituent 2 from clustering

...

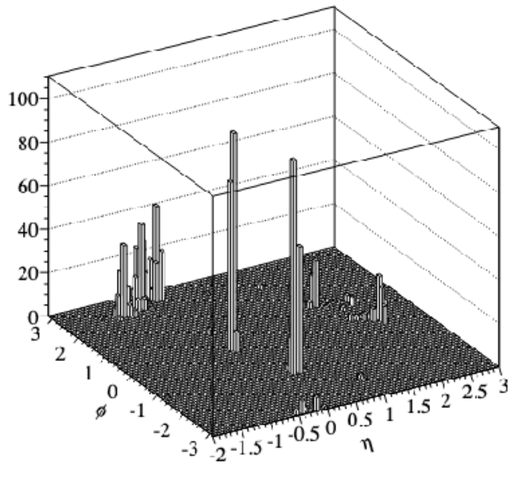
Different decisions in same clustering step



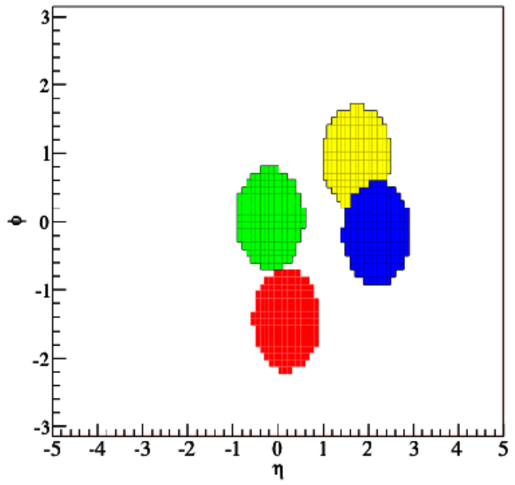
→ Cluster same event repeatedly to obtain expectation value

Qanti-k_T

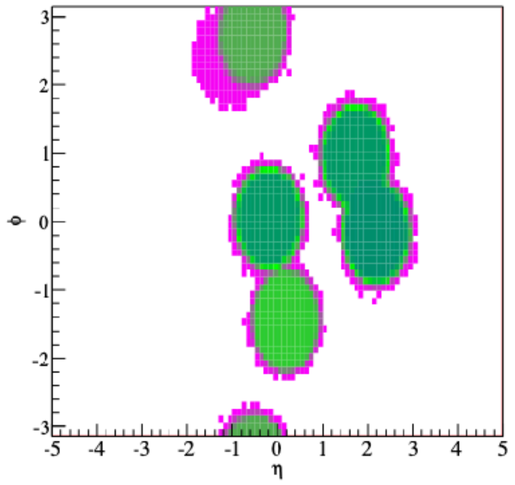
Influence of α on the final clustering



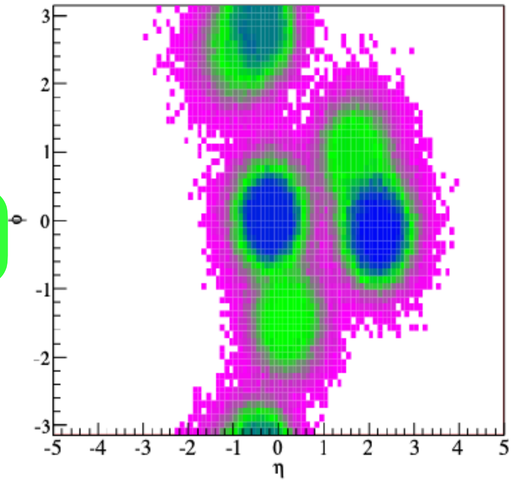
classical anti-k_T



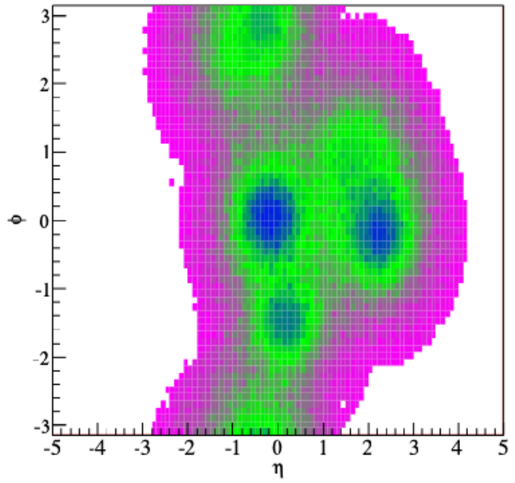
$\alpha = 10$



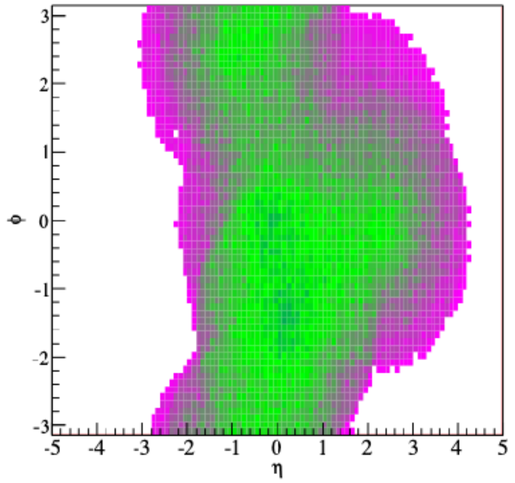
$\alpha = 1.0$



$\alpha = 0.1$



$\alpha = 0.01$



Almost always included

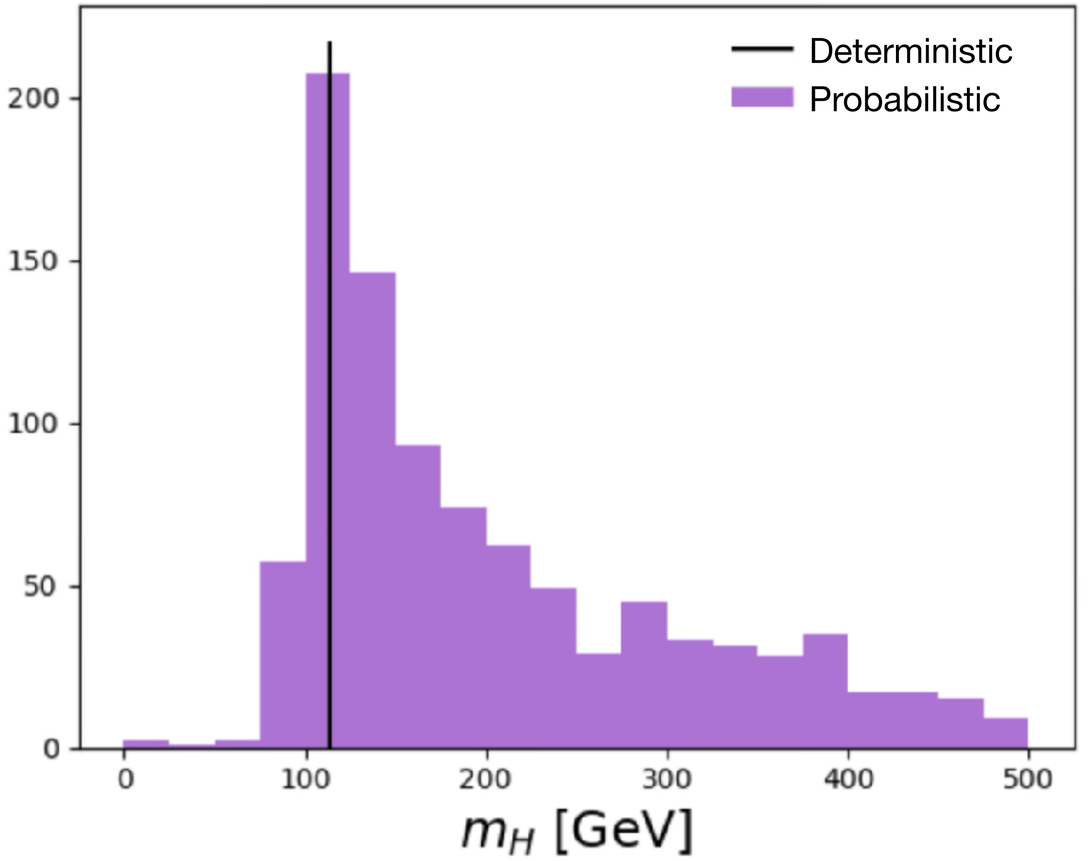
Included 50% of the time

Rarely included

Fig 1 from arXiv:1304.2394

Our quantity of interest: m_H^{reco}

- Same event, different clusterings → reconstructed m_H changes
- α determines width of distribution



Step 2: Making a stochastic program differentiable

Gradient estimators

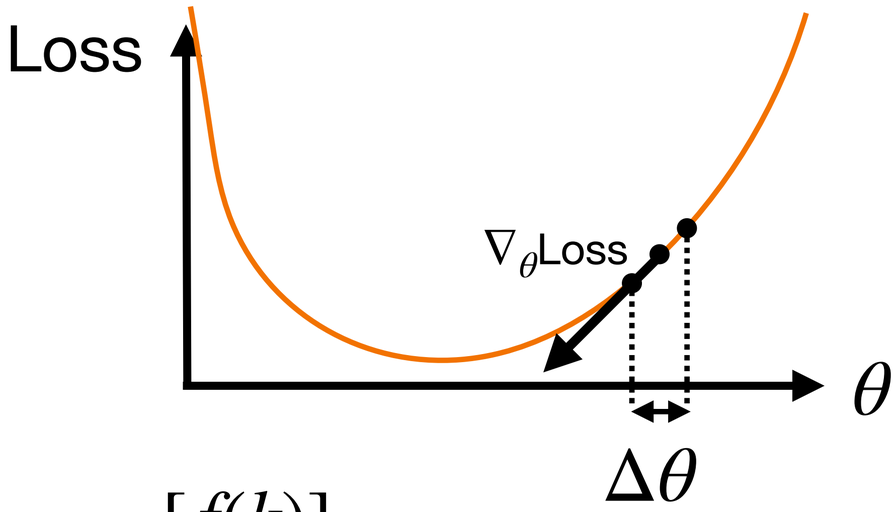
Different options:

- Finite differences — This talk
- Score function gradient (REINFORCE) — Glynn 1990, doi:10.1145/84537.84552
- Gumbel-softmax straight through estimator (GS-STE) — Jang et al. 2017, arXiv:1611.01144
- Alternative path estimator (stochAD) — Arya et al. 2024, arXiv:2210.08572
- …

Not clear which one is the best*.

*= No bias, lowest variance.

Finite differences



- Choose discretization $\Delta\theta$

$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}\left[f(k)\right] = \frac{\mathbb{E}_{k\sim p_{\theta+\Delta\theta}(k)}[f(k)] - \mathbb{E}_{k\sim p_\theta(k)}[f(k)]}{\Delta\theta}$$

- Issue: Diverges for $\Delta\theta \to 0$
- Difficult in high dimensions

Score-based gradient

Why “score”?

$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}\left[f(k,\theta)\right] = \int \nabla_\theta f(k,\theta) p_\theta(k)\, \mathrm{d}k$$

$$\nabla_\theta p_\theta(k) = p_\theta(k)\, \nabla_\theta \log p_\theta(k)$$

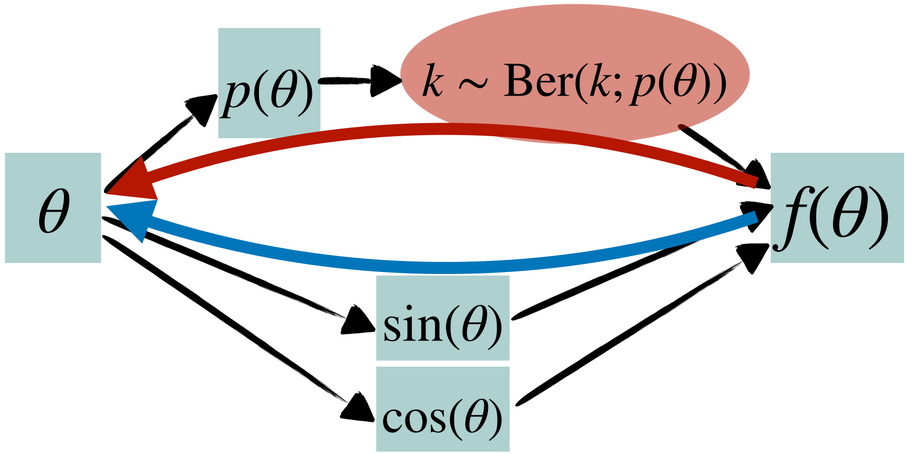
$$= \int p_\theta(k) \nabla_\theta f(k,\theta)\, \mathrm{d}k + \int f(k,\theta) \nabla_\theta p_\theta(k)\, \mathrm{d}k$$

$$= \int p_\theta(k) \left(\nabla_\theta f(k,\theta) + f(k,\theta) \nabla_\theta \log p_\theta(k) \right) \mathrm{d}k$$

$$= \mathbb{E}_{k\sim p_\theta(k)} \left[\nabla_\theta f(k,\theta) + f(k,\theta) \nabla_\theta \log p_\theta(k) \right]$$

“Score”

Score-based gradient



Gradient contributions:

$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}\left[f(k,\theta)\right] = \mathbb{E}_{k\sim p_\theta(k)}\left[\nabla_\theta f(k,\theta) + f(k,\theta)\nabla_\theta \log p_\theta(k)\right]$$

Direct dependency of f on θ

“Path-wise gradient”

Probabilistic dependency of k on θ

“Score-function”

Unbiased, but large variance

Glynn, 1990, doi:10.1145/84537.84552
Williams, 1992, doi:10.1007/BF00992696
Kleijnen & Rubinstein, 1996, doi:10.1016/0377-2217(95)00107-7

Introducing a baseline

- Reduce variance: subtract baseline ('control variate') $c(\theta)$

$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}\left[f(k,\theta)\right] = \mathbb{E}_{k\sim p_\theta(k)}\left[\nabla_\theta f(k,\theta) + \left(f(k,\theta) - c(\theta)\right)\nabla_\theta \log p_\theta(k)\right]$$

- Need unbiased baseline: $\mathbb{E}_{k\sim p_\theta(k)}\left[c(\theta)\nabla_\theta \log p_\theta(k)\right] = 0$

- Common choice: mean baseline $c(\theta) = \bar{f} = \mathbb{E}_{k\sim p_\theta(k)}[f(k,\theta)]$

Greensmith et al., 2004, JMLR, doi:10.5555/1005332.1044710

Gumbel-Softmax Straight Through Estimator (GS-STE)

- Idea: reparametrize categorical sampling $k \sim \mathrm{Cat}[p_1, \ldots, p_K]$
- Given $w_k \in (0,\infty)$ and $u_k \sim U[0,1]$

$$k = \underset{k\in\{1,\ldots,K\}}{\arg\max} \left[\log w_k \underbrace{- \log(-\log u_k)}\right]$$

$$= g_k$$

$$g_k \sim \mathrm{Gumbel}(0,1) = \exp\left((-g_k - \exp(-g_k)\right)$$

- "Gumbel-Max Trick"

$$k = \underset{k\in\{1,\ldots,K\}}{\arg\max} \left[\log w_k + g_k\right]$$

Jang et al. 2017, arXiv:1611.01144

Gumbel-Softmax Straight Through Estimator (GS-STE)

- Reparametrize expectation value:

$$\mathbb{E}_{k\sim p_\theta(k)}\left[f(k)\right] = \mathbb{E}_{g_k\sim G(0,1)}\left[f(k_\theta(g_k))\right]$$

- Expectation value is differentiable:

$$\nabla_\theta \mathbb{E}_{k\sim p_\theta(k)}\left[f(k)\right] = \mathbb{E}_{g_k\sim G(0,1)}\left[\nabla_\theta f(k_\theta(g_k))\right]$$

- But $k_\theta = \underset{k\in\{1,\ldots,K\}}{\arg\max}\left[\log w_k(\theta) + g_k\right]$ not differentiable

Jang et al. 2017, arXiv:1611.01144

Gumbel-Softmax Straight Through Estimator (GS-STE)

- Keep forward pass discrete: $\vec{z}_\theta = \text{one-hot}(k_\theta)$
- Apply **straight-through estimator** in backward pass

$$\nabla_\theta \vec{z}_\theta \approx \nabla_\theta \vec{s}_\theta(g_1, \ldots, g_K)$$

$$= \nabla_\theta \ \text{softmax} \ (\log p_\theta(k) + g_k)$$

$$= \nabla_\theta \frac{\exp[(\log p_\theta(k) + g_k)/\tau]}{\sum_{k=1}^{K} \exp[(\log p_\theta(k) + g_k)/\tau]}$$

With temperature τ

Jang et al. 2017, arXiv:1611.01144

Gumbel-Softmax Straight Through Estimator (GS-STE)

Temperature τ controls bias-variance trade-off

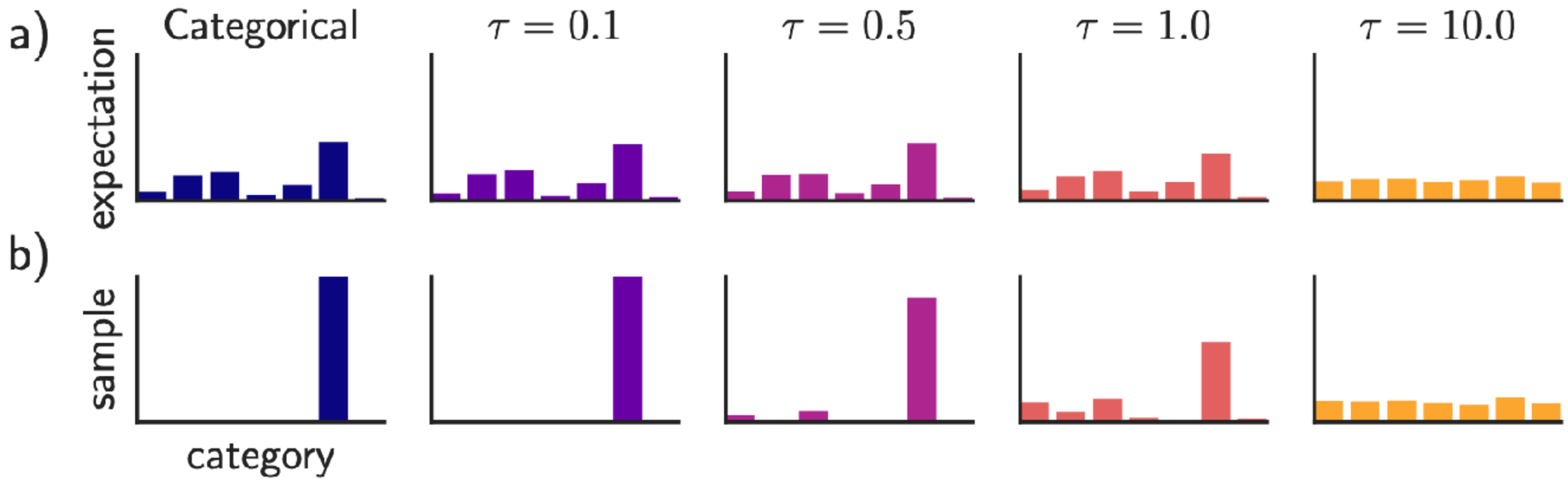


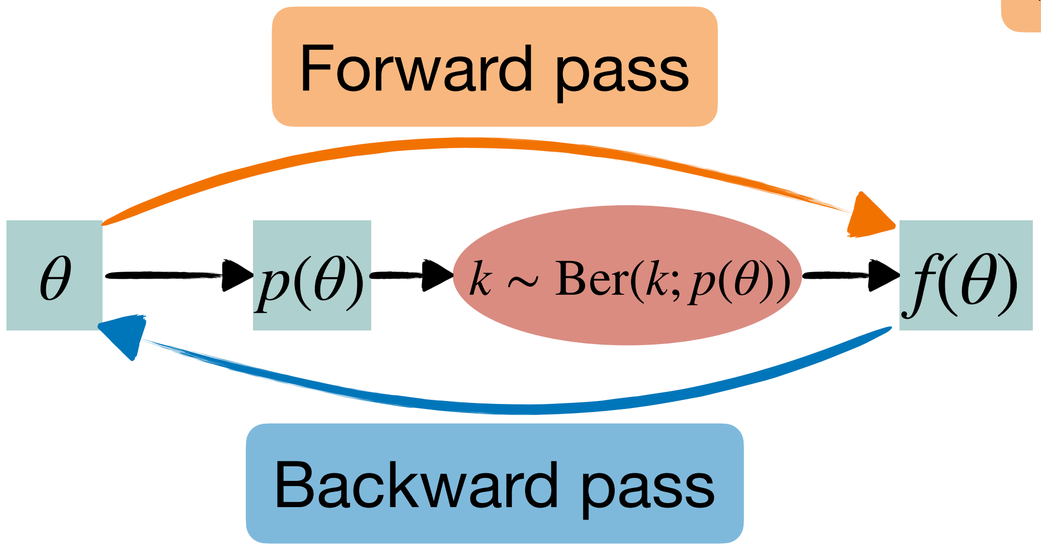
Figure 1 from arXiv:1611.01144

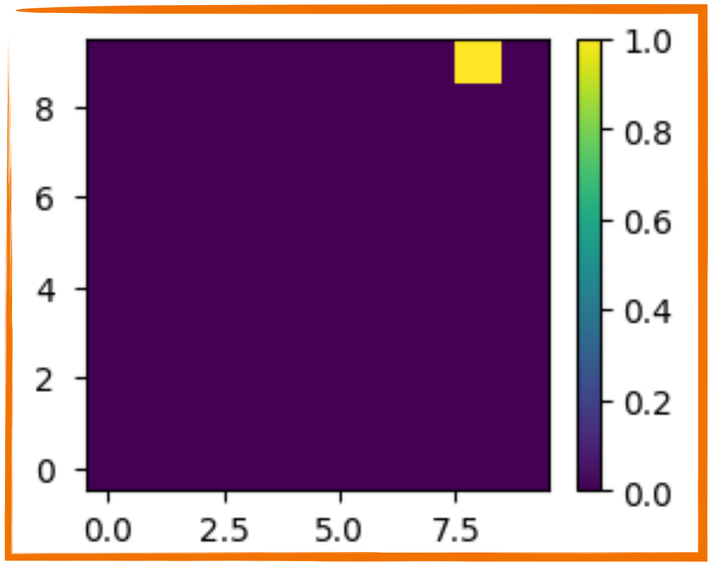
Jang et al. 2017, arXiv:1611.01144

Gumbel-Softmax Straight Through Estimator (GS-STE)

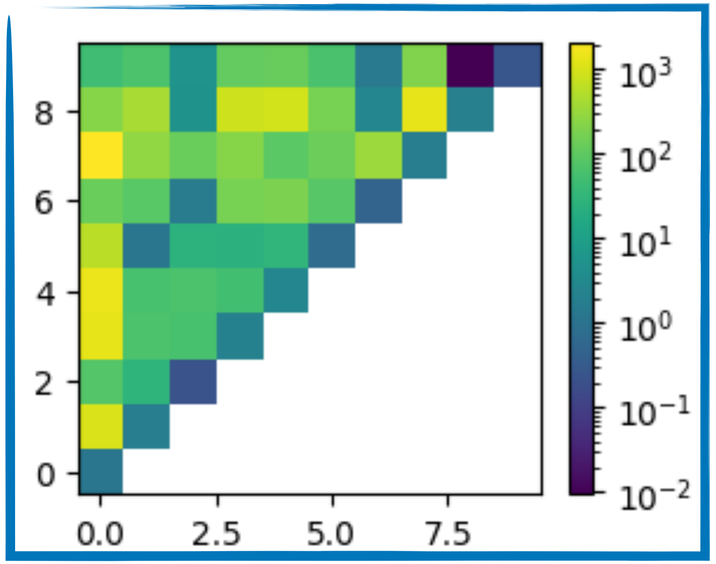
Illustration:

$$\vec{z}_\theta = \text{one-hot}(k_\theta)$$





$$\nabla_\theta \vec{z}_\theta \approx \nabla_\theta \vec{s}_\theta(g_1, \ldots, g_K)$$

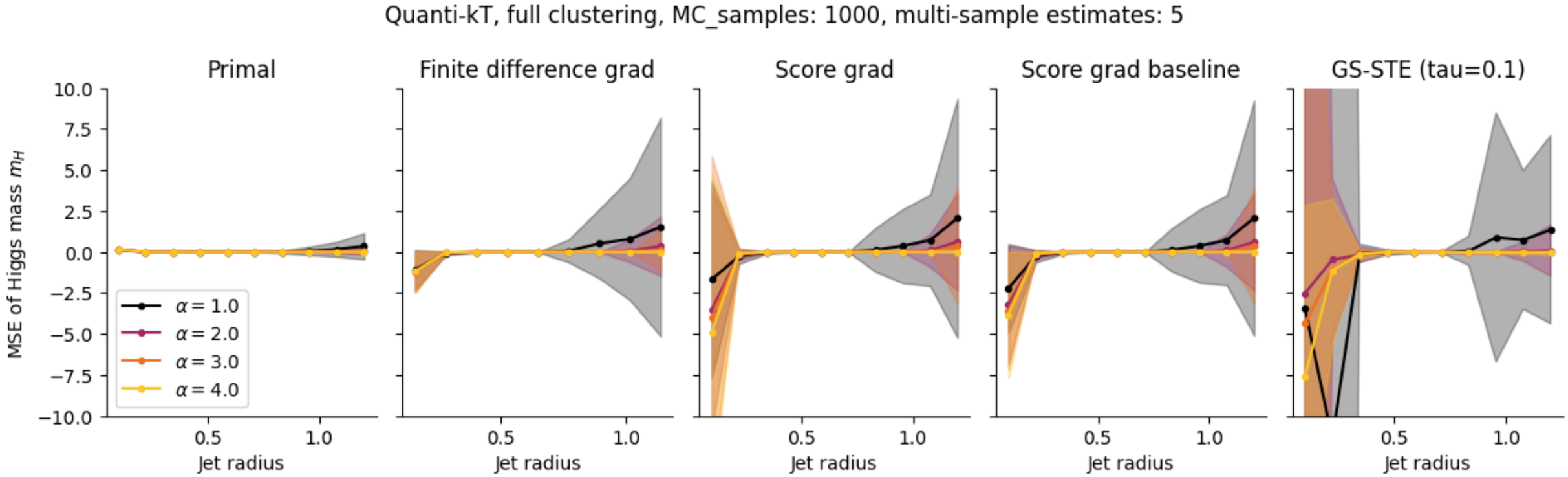


Jang et al. 2017, arXiv:1611.01144

Back to jet clustering …

Differentiable jet clustering

Single event: Comparing gradient estimators



Finally: Optimizing jet clustering

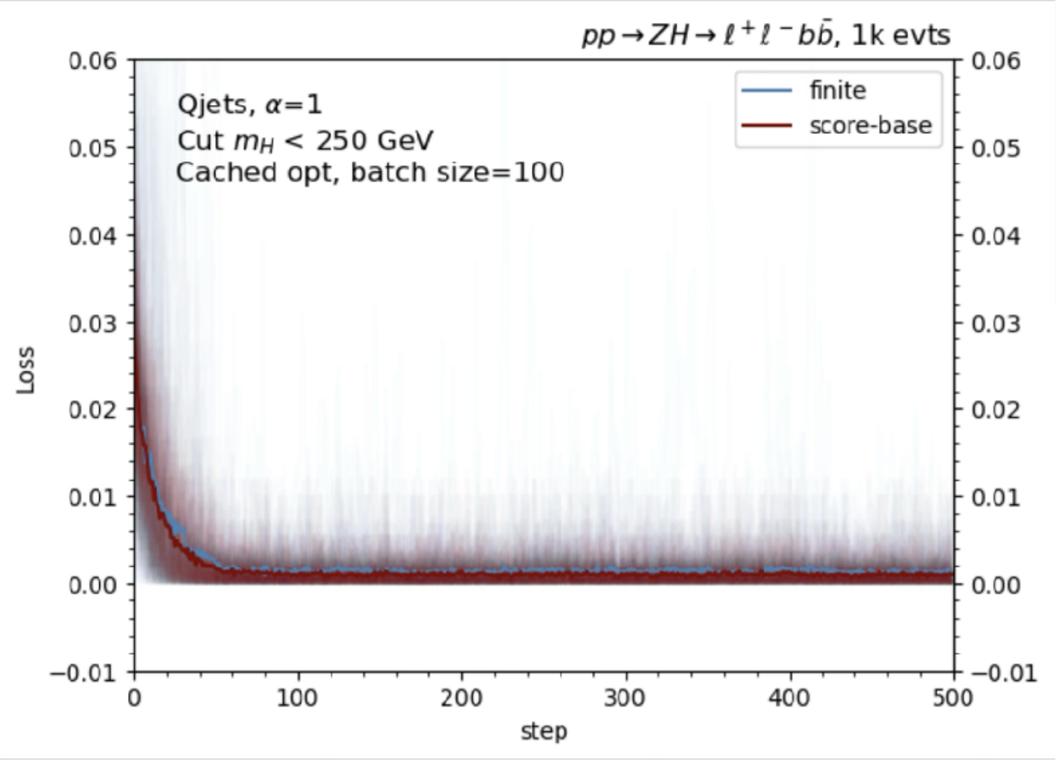
Ingredients for an optimization

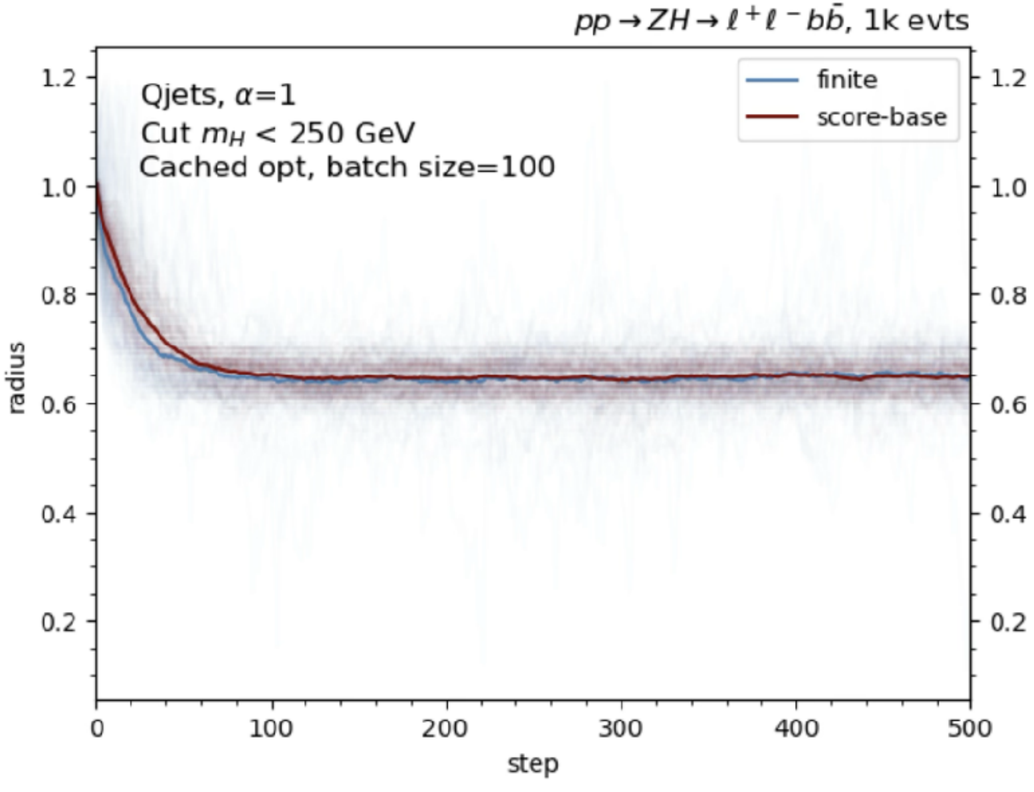
- Gradients ✅
- Loss function: $\mathscr{L}(R) = \mathrm{MSE}\left[m_H^{\mathrm{reco}}(e,\kappa),\ m_H^{\mathrm{true}}\right]$ ← Skipping some details here

- Optimizer: SGD or Adam

Multi-event optimization

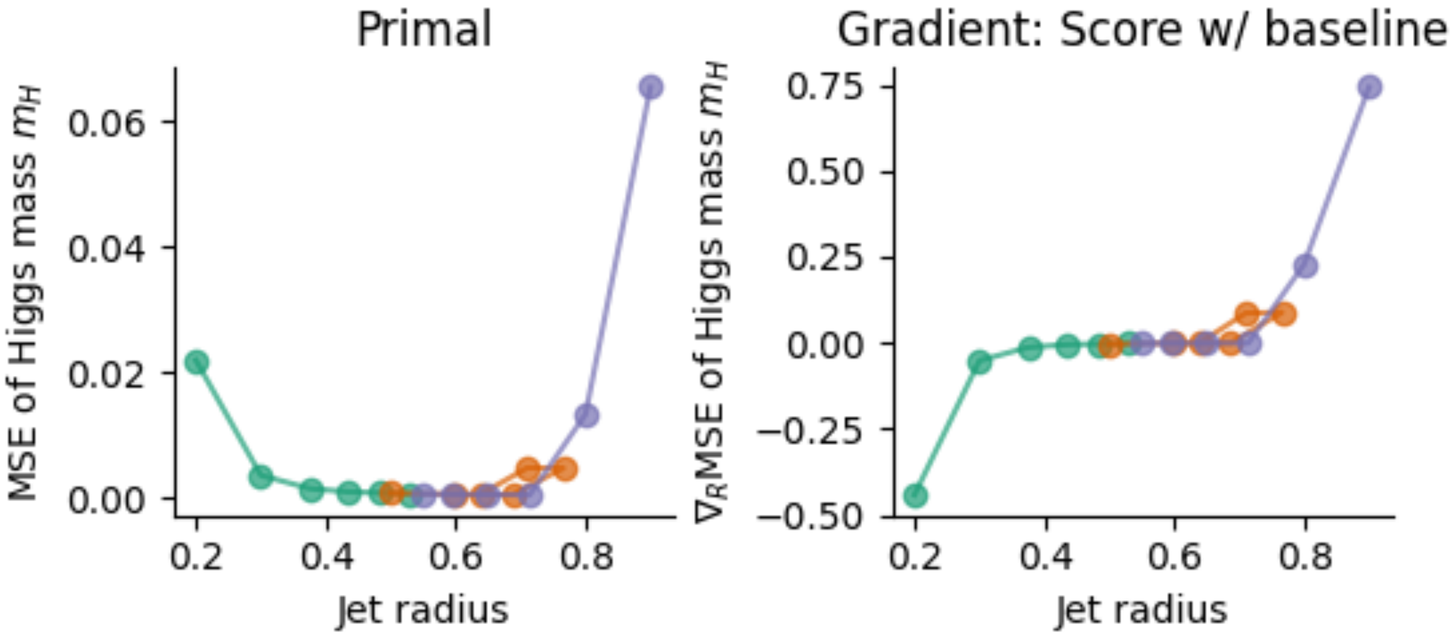
- Goal: global jet radius for 1000 events





Single-event optimization

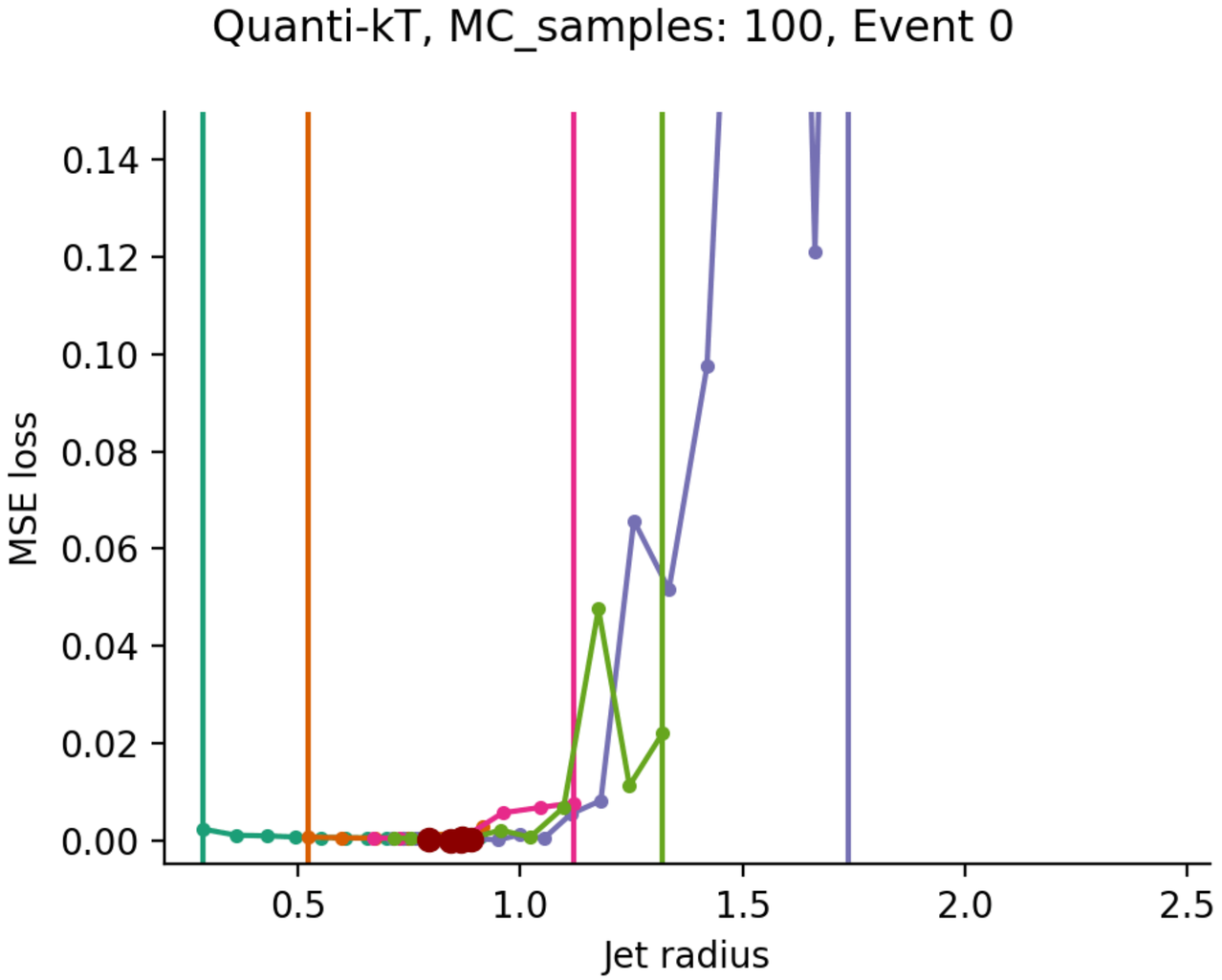
- Goal: local jet radius for each event individually



$\alpha = 1, N = 1000$

Single-event optimization

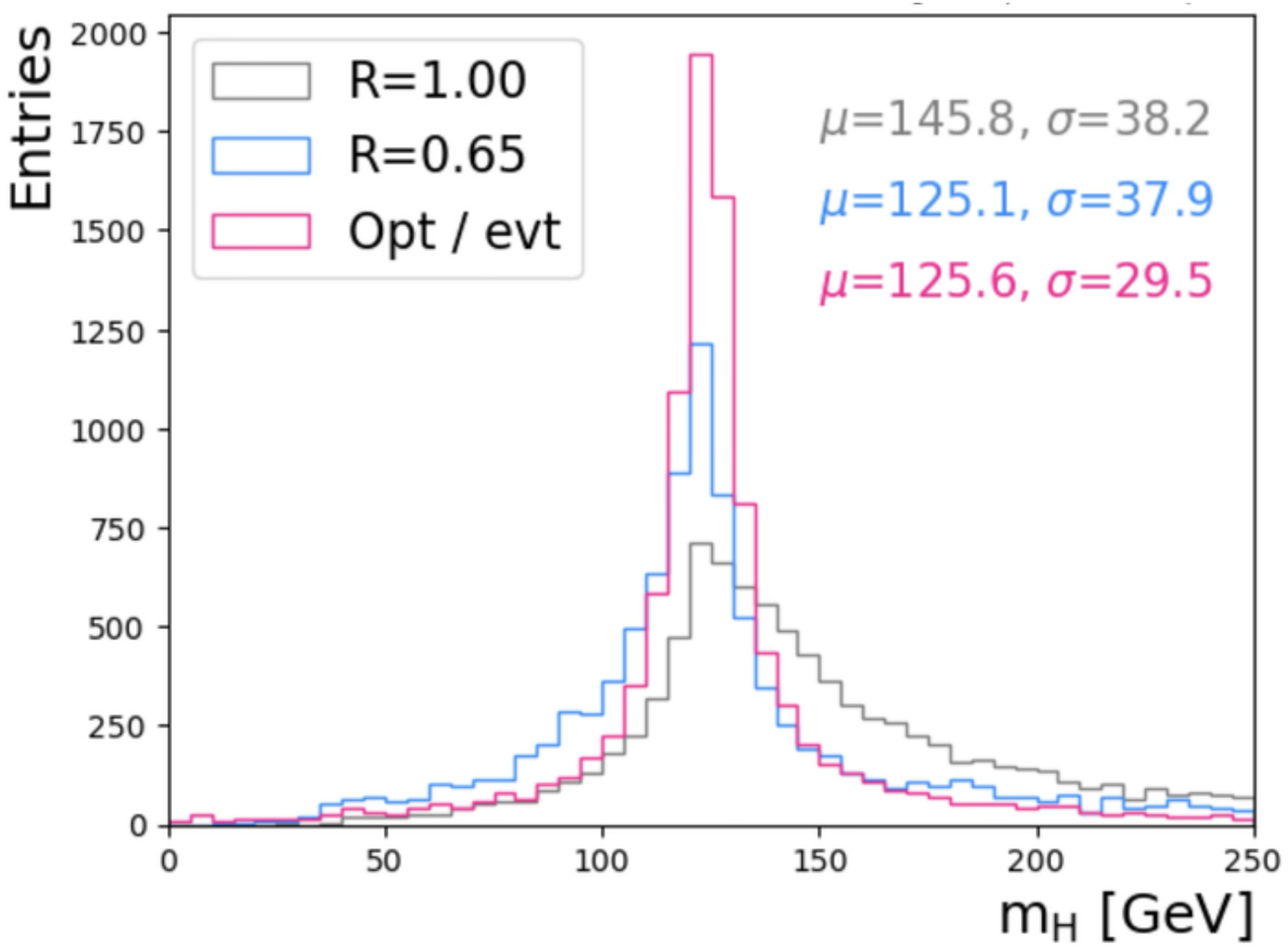
- Optimize multiple events
- Loss landscape changes from event to event
- Different initializations converge to a similar minimum



$\alpha = 1$

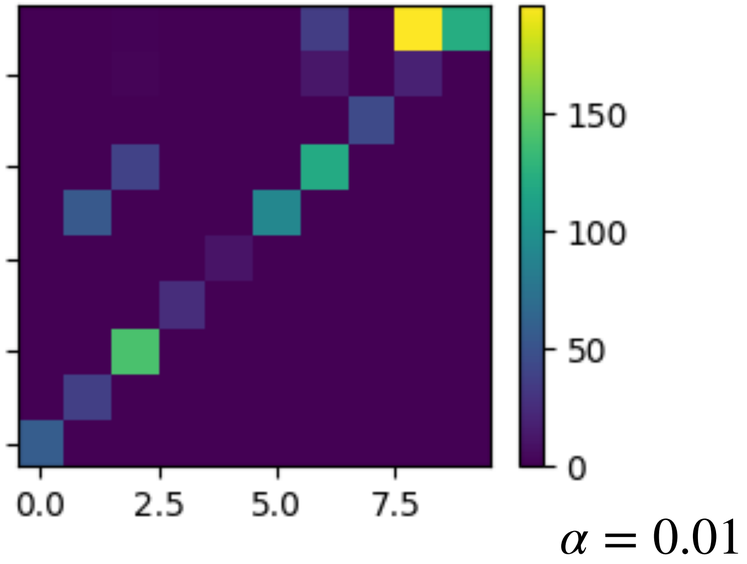
Final goal

- How does an optimized jet radius help us?
- Sharper resonance of m_H



Take aways

- Discrete decisions not differentiable
- Solution:
  Step 1: Make it probabilistic
  Step 2: Make it differentiable
- Different gradient estimators:
  - Finite differences
  - Score based gradient
  - Gumbel-softmax straight through estimator (GS-STE)
  - …



$\alpha = 0.01$