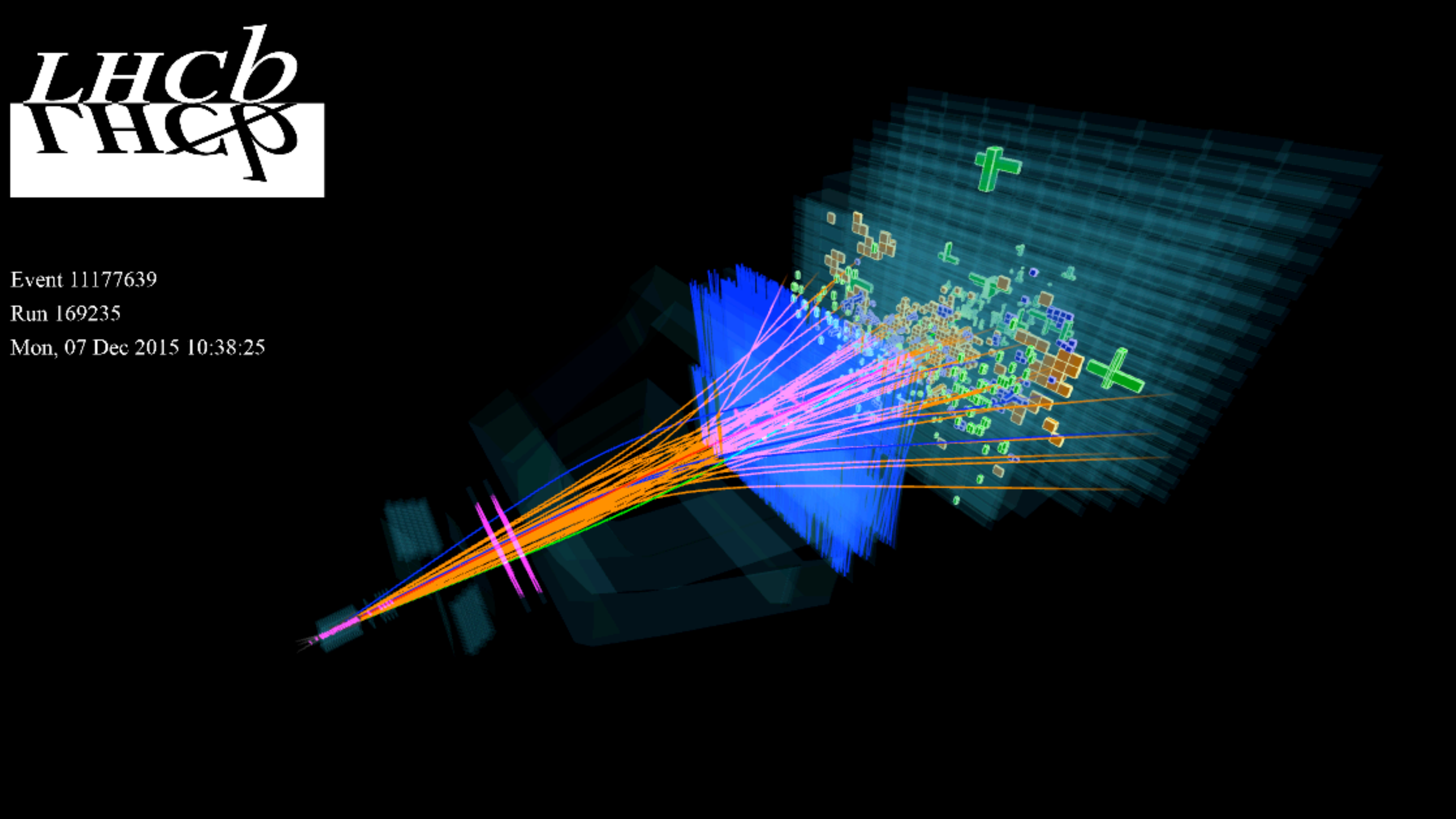
Lipschitz Networks
Optimal Transport
Other Things

Niklas Nolte - MLHEP - 2023

Overview

1. The LHCb trigger
2. Lipschitz Networks - Robustness and Monotonicity
3. Energy Flow & EMD
4. Emergent Capabilities of Neural Networks

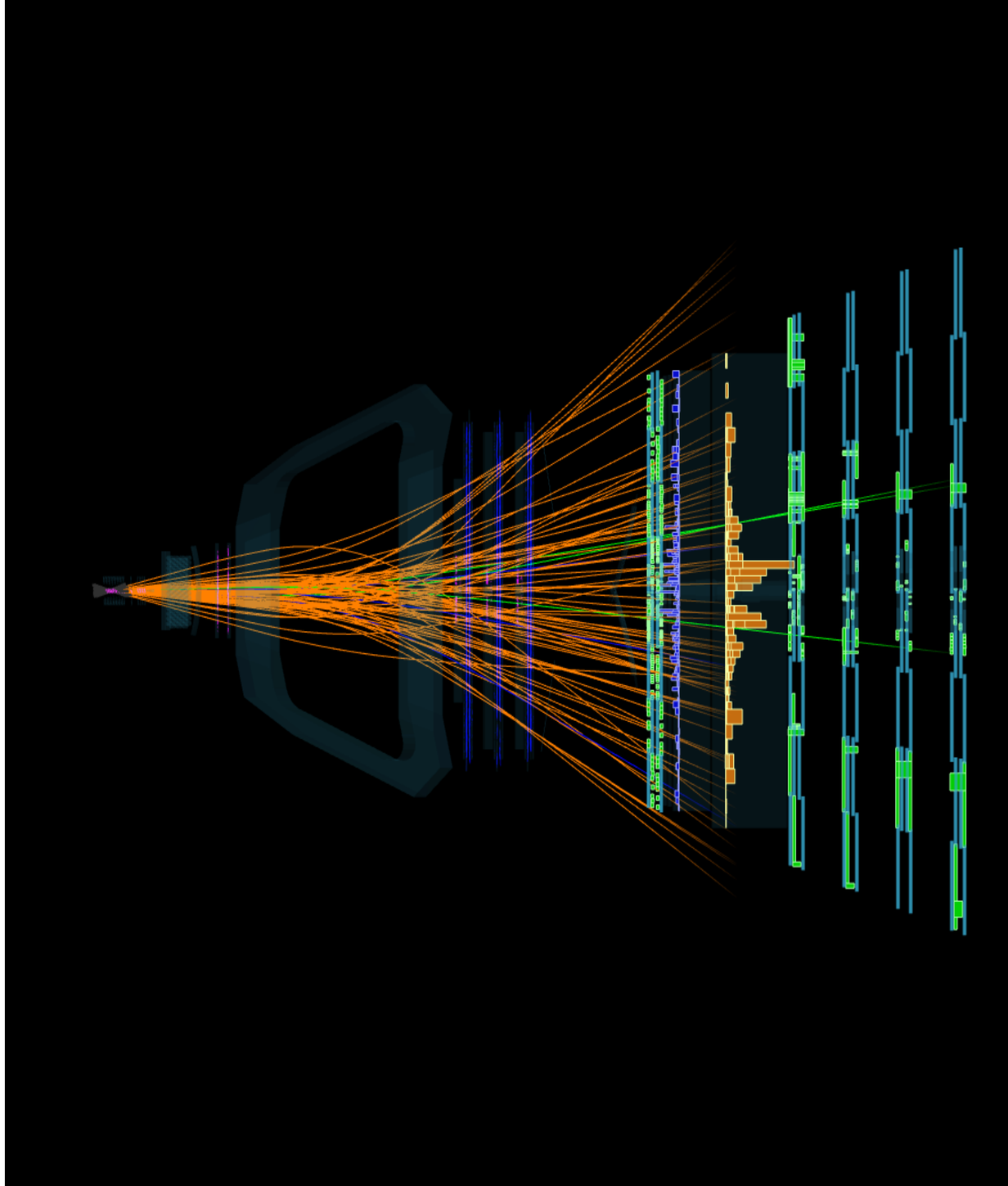

IAIFI.org


LHCb
Event 11177639
Run 169235
Mon, 07 Dec 2015 10:38:25

The Trigger at LHCb

LHCb Raw data
15000 PB/year

LHCb storage capacity
30 PB/year



Select **only interesting** events:
Hybrid approach of expert systems and ML

Real time data reduction: 5 TB/s → 10GB/s

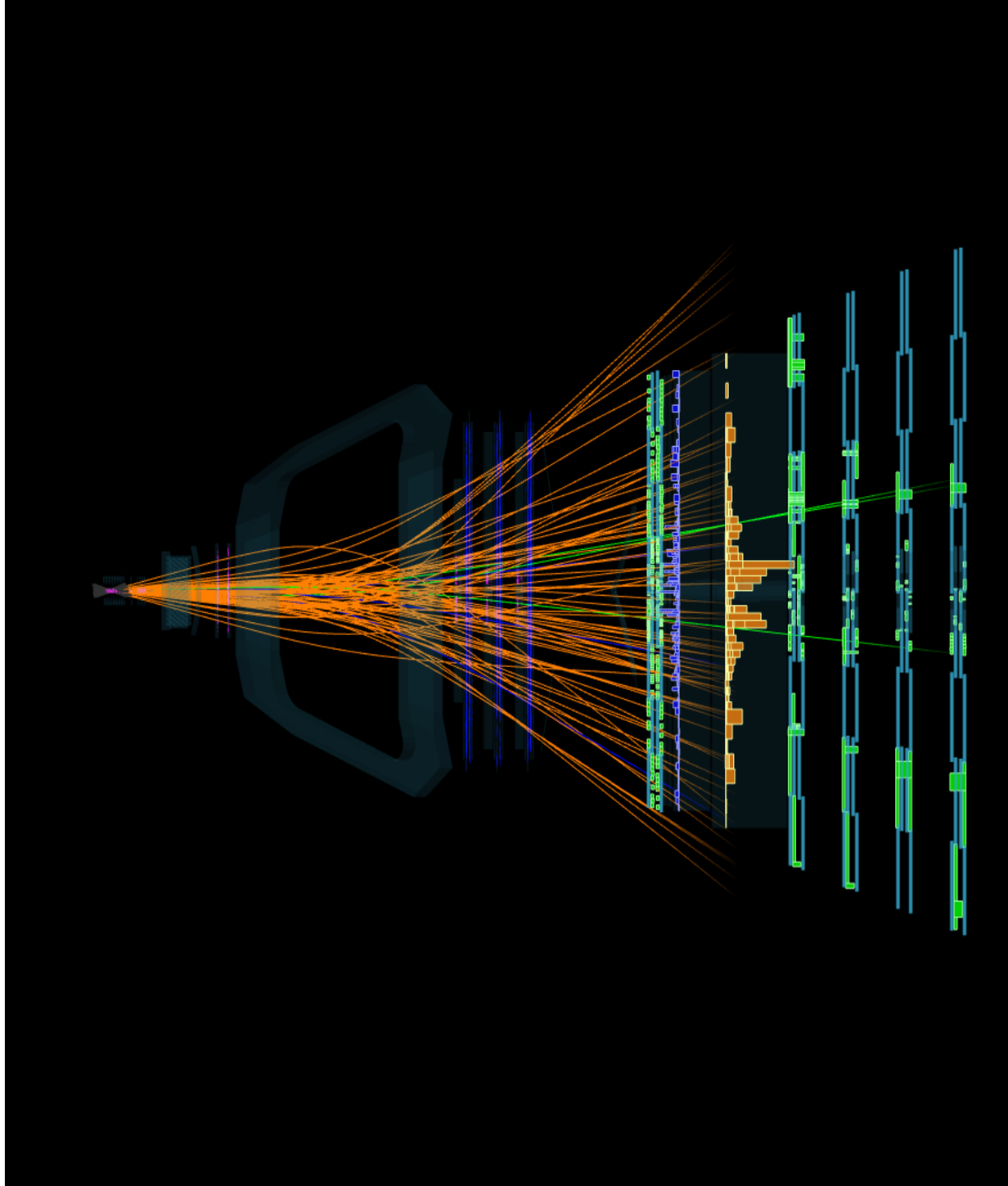
The Trigger at LHCb

Real time expert systems + ML
to process 5 TB/s

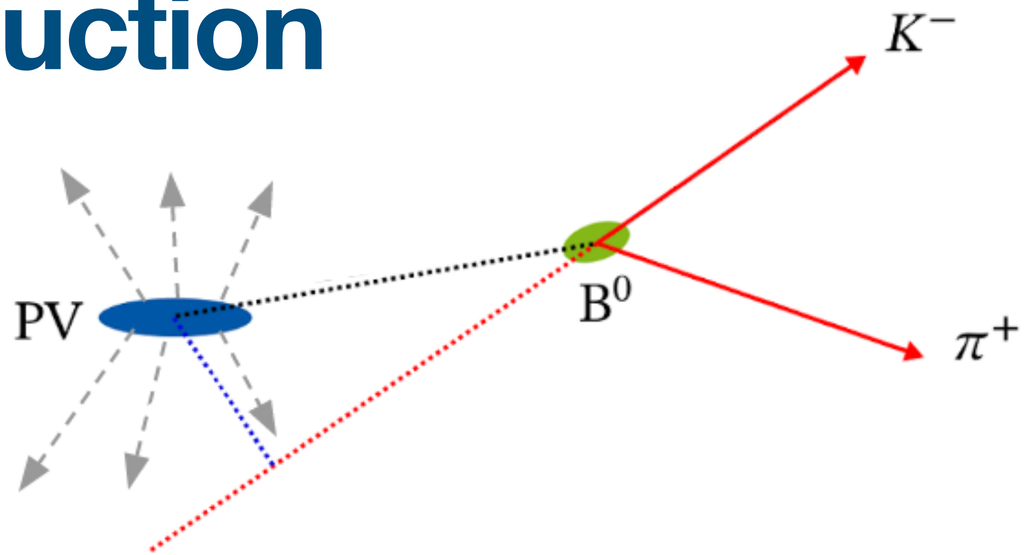
→ High performance software
on GPU and CPU

500x data reduction

→ high purity selection and
good event compression



Event Selection -- data reduction



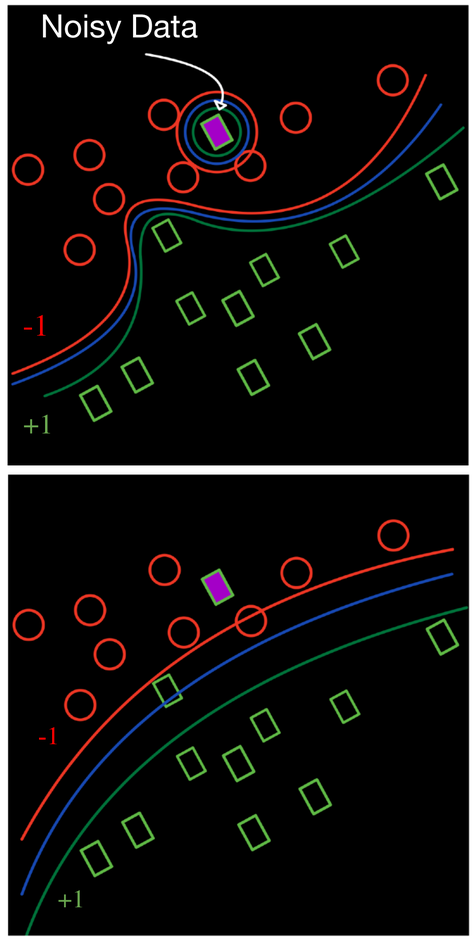
Trigger: mostly an expert system

Many subsystems look for particular decays
→ Strong reduction and purity ✅

Some look for general signatures, weaker selection
→ Achieve good purity with ML classifiers
Need guarantees to employ these! **No room for error**

Guarantees needed:
1. Robustness w.r.t small changes
2. Monotonicity in certain features for OOD guarantees

(Adversarial) Robustness

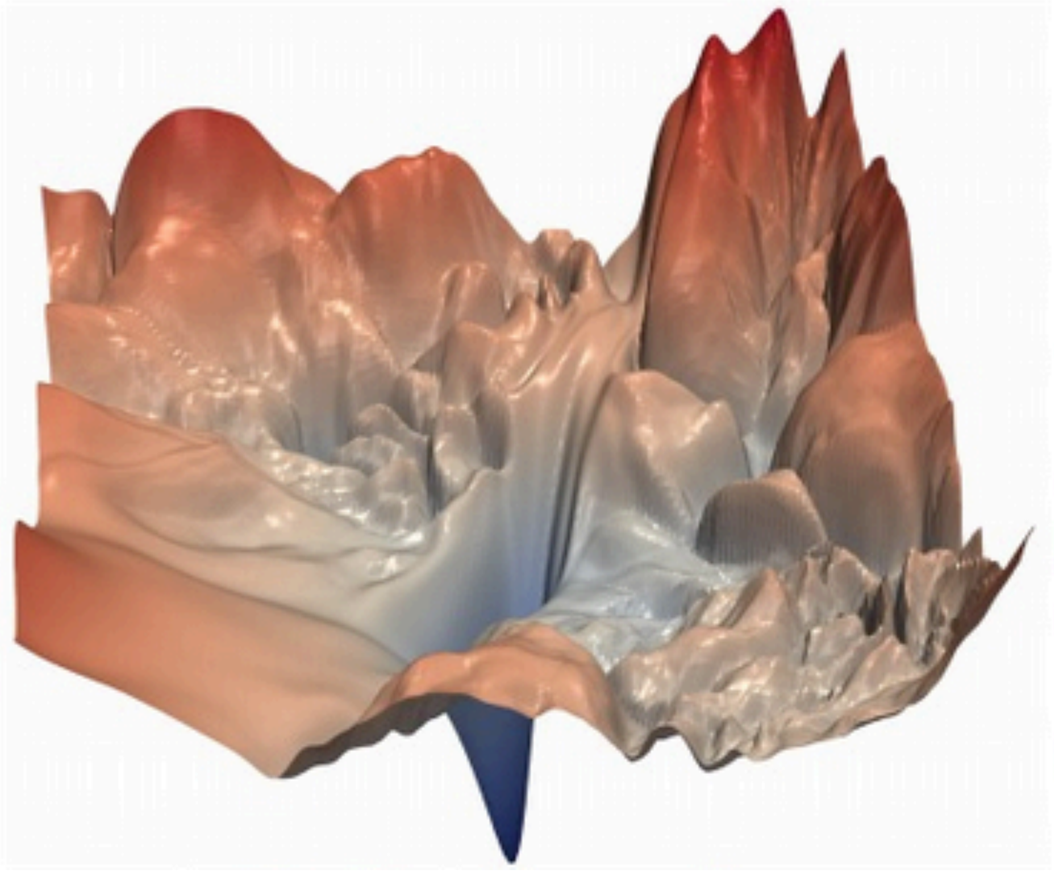


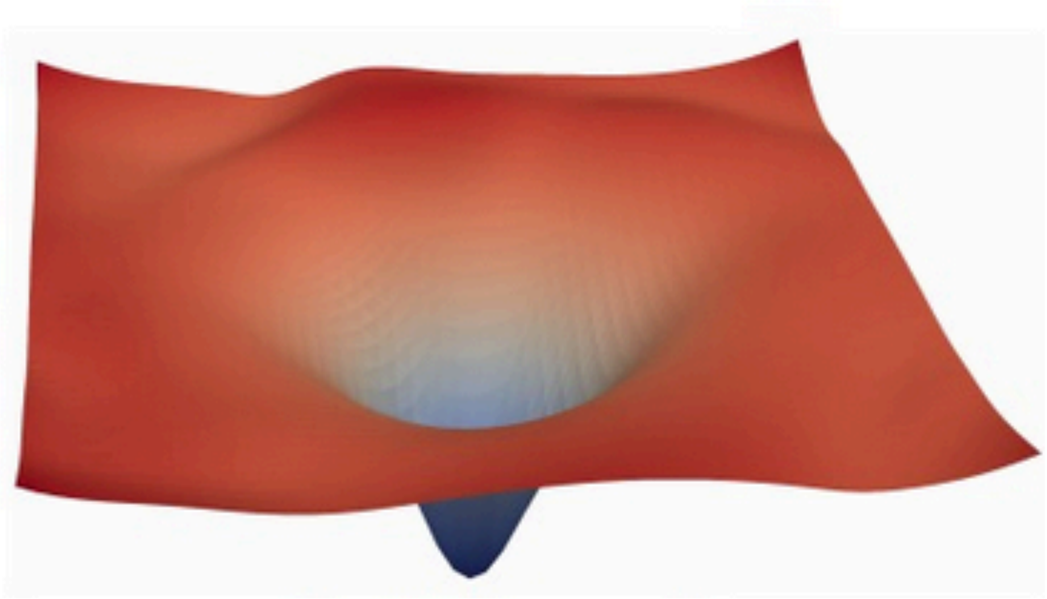
many SOTA ML models are proven to be highly unstable

Robustness $:= \quad F(x+\epsilon) = F(x) + O(\epsilon)$

I am looking for deterministic robustness, i.e. provably robust networks!

Two Decision Landscapes





Deterministic Robustness - How?

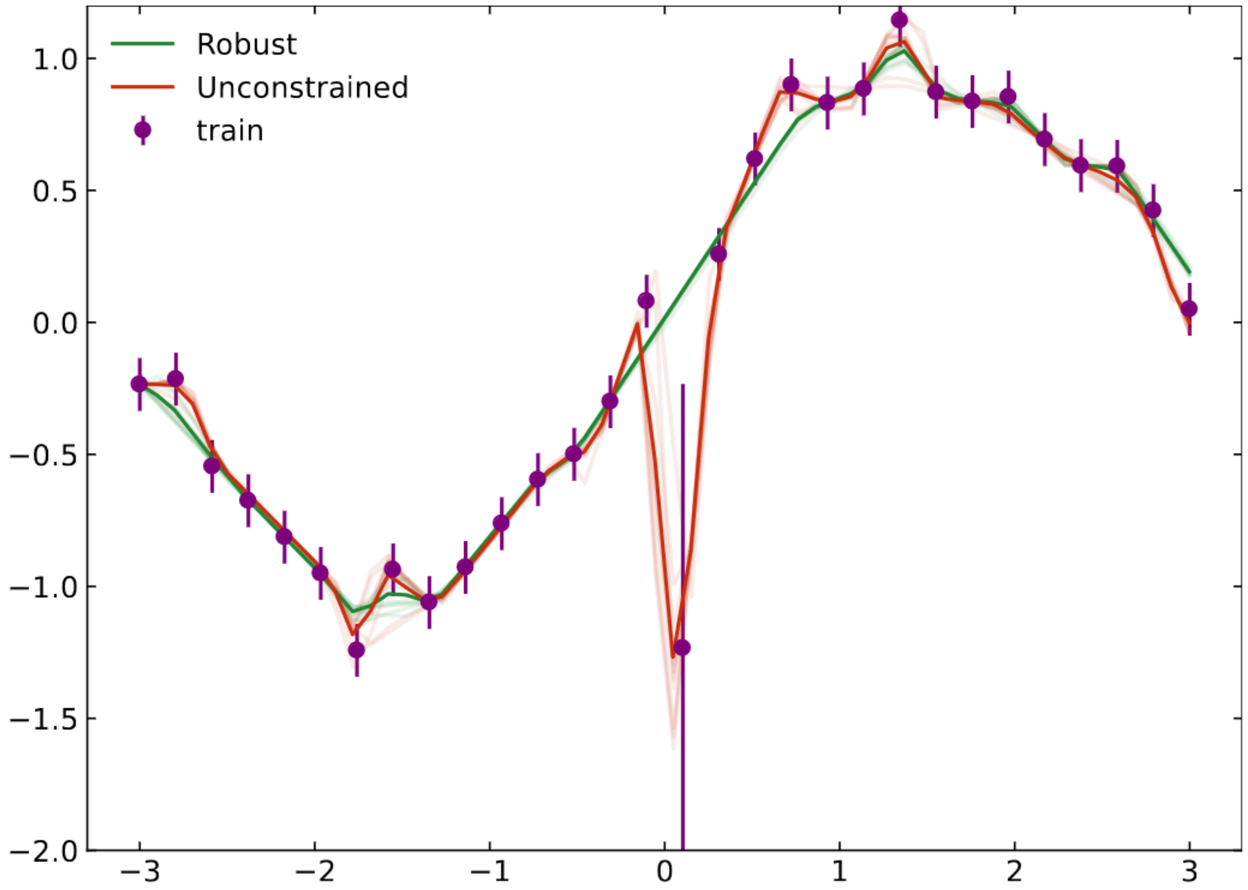
WLOG: Binary classifier: $F : \mathbb{R}^n \to \mathbb{R}$

Constrain gradient wrt inputs, i.e. make it Lipschitz-L: $\|\nabla F\| \leq L$

A perturbation ϵ to an input x needs certain magnitude to flip the sign:

$$\operatorname{sign} F(x+\epsilon) = -\operatorname{sign} F(x) \Rightarrow$$

$$\|\epsilon\| > \frac{|F(x)|}{L}$$



Lipschitz Networks

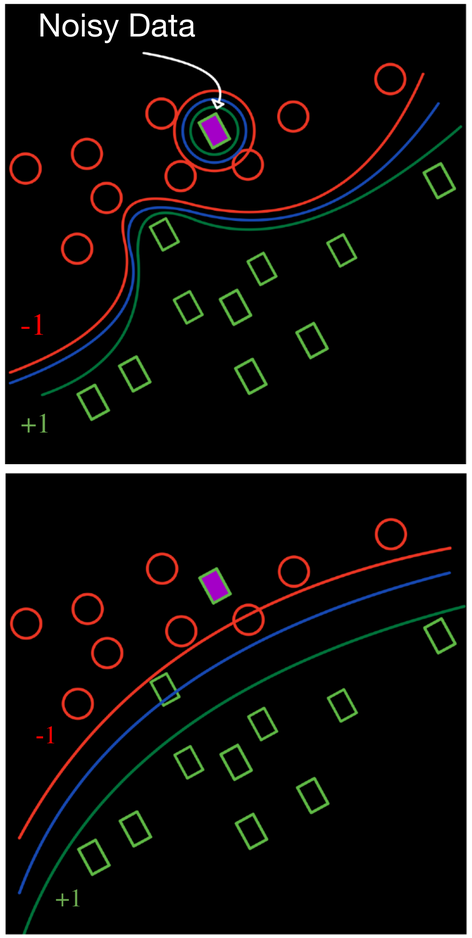
WLOG: $F(x) = W^{(2)} \cdot \sigma(W^{(1)} \cdot x + b^{(1)}) + b^{(2)}$

$\|\nabla F\| \leq L$ can be enforced by constraining weights

In an MLP with Lipschitz-1 activations: $L \leq \prod \|W^{(i)}\|$

Maintain a maximum operator norm $\|W^{(i)}\|$ in every layer

Lipschitz-L guaranteed!



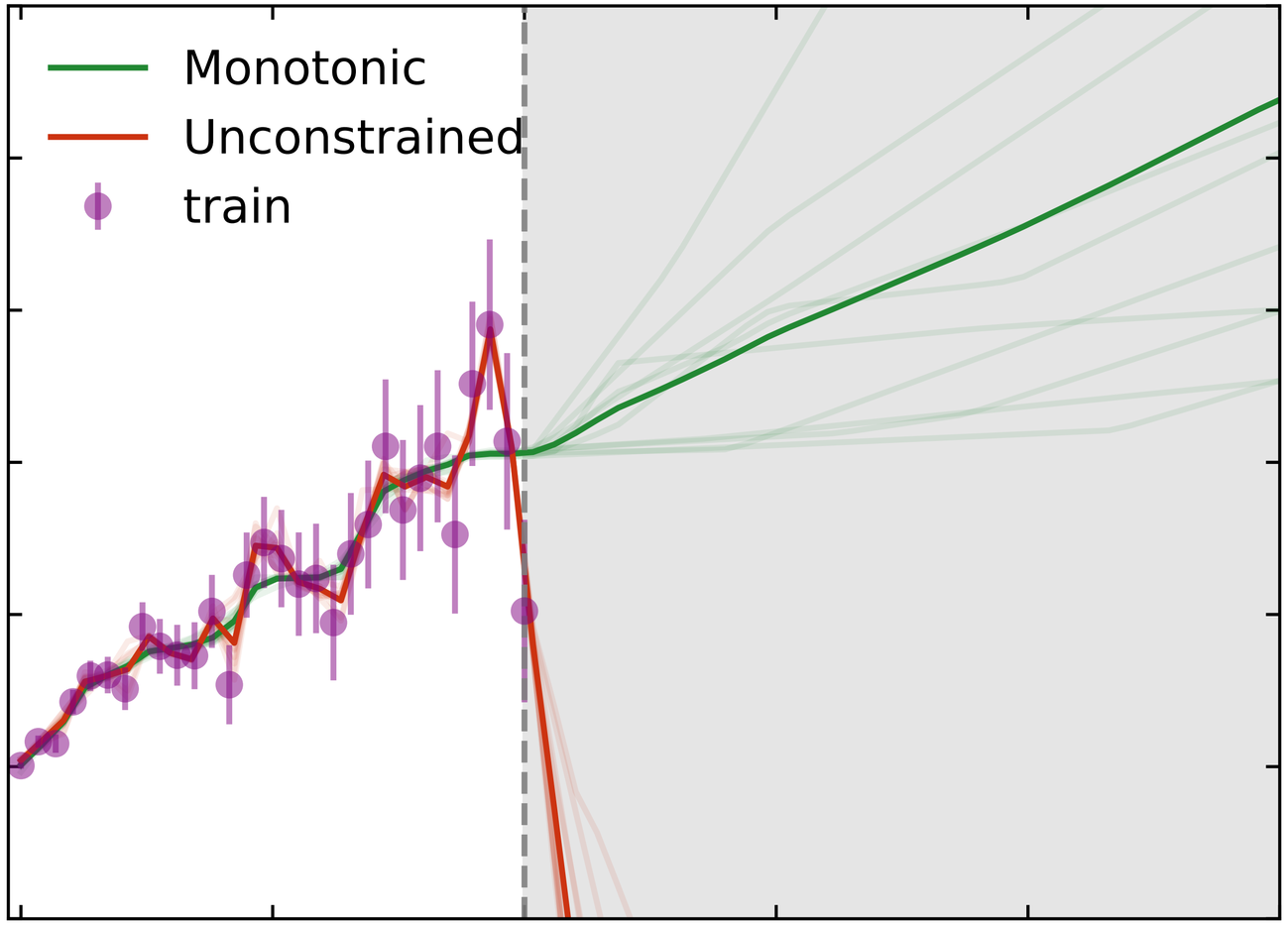
Monotonic Networks

We care about tails!

Guarantees about OOD
with monotonicity

Expressive monotonic networks
are not obvious

Existing algorithms involve
monotonic regularization or non-
expressive architectures



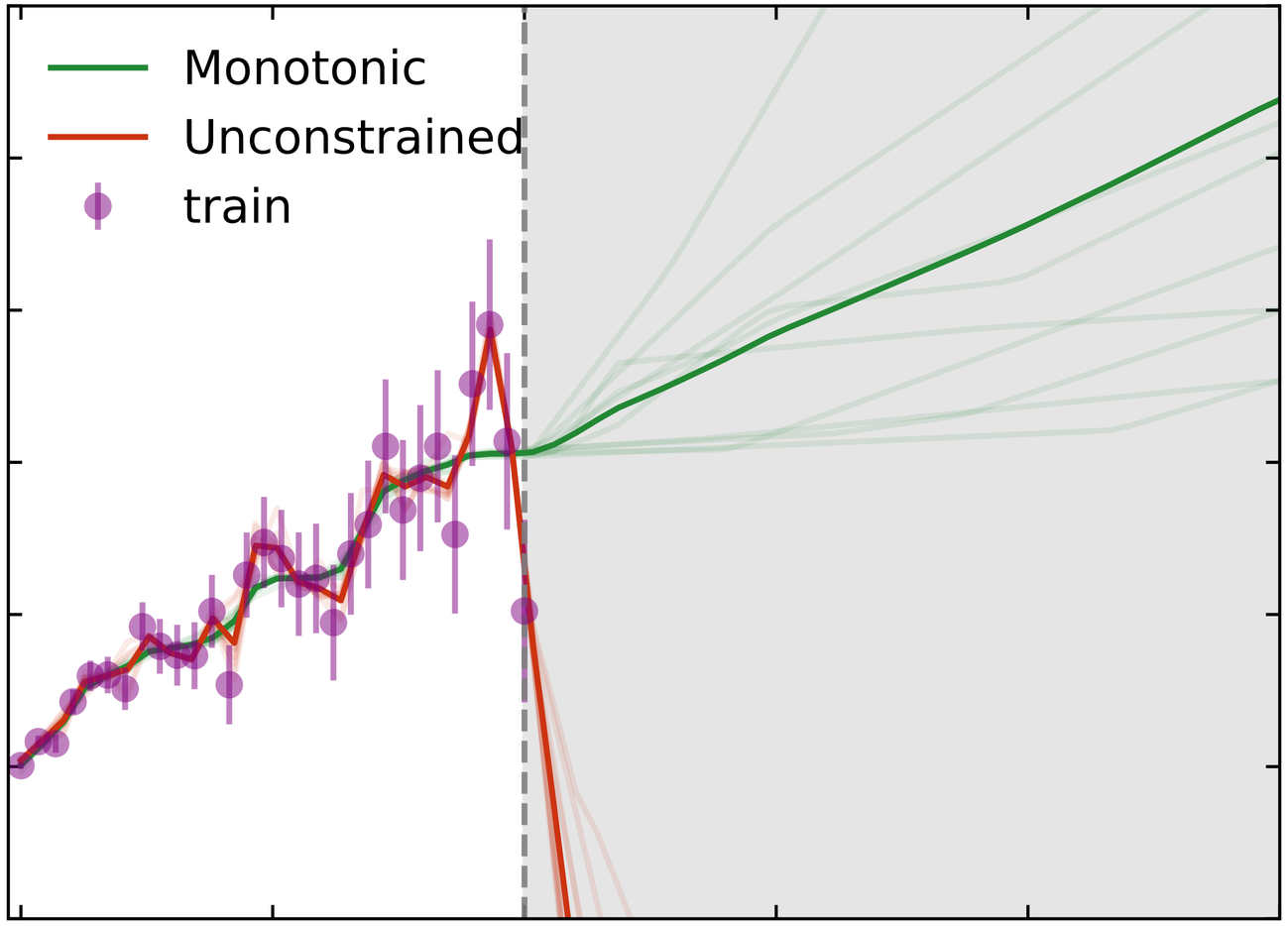
Monotonic Lipschitz Networks

Combine Lipschitz networks with monotonicity!

$$M(x) = F(x) + L \sum_i x_i$$

$$\frac{\partial M}{\partial x_i} = \frac{\partial F}{\partial x_i} + L \geq 0$$

Lipschitz-L : $\|\nabla F\| \leq L$



Monotonic Lipschitz Networks

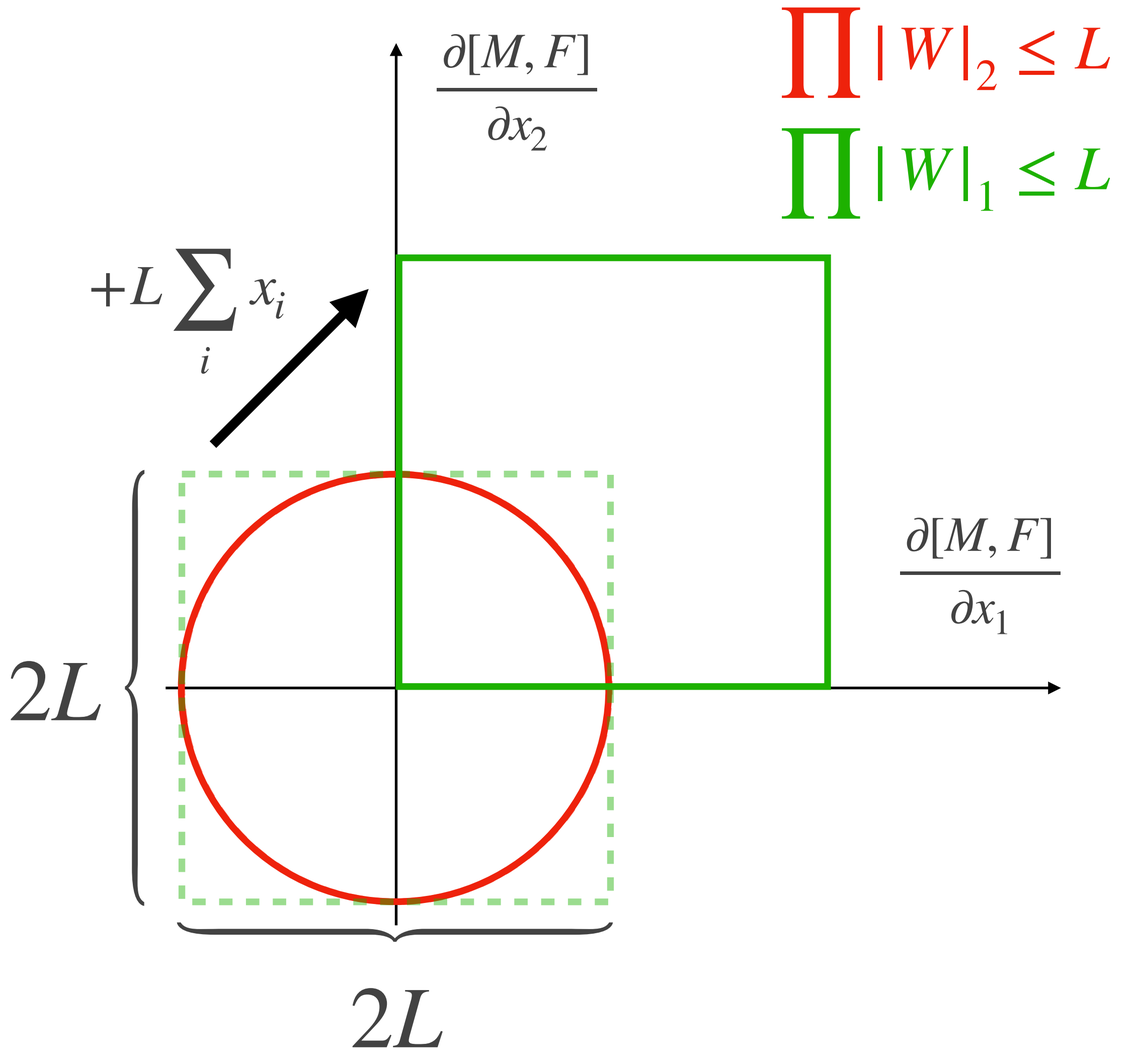
$$M(x) = F(x) + L\sum_i x_i$$

$$\frac{\partial M}{\partial x_i} = \frac{\partial F}{\partial x_i} + L$$

$+L$ contribution in every direction x_i

$\|\nabla F\| \leq L$ is not good enough

We want $\|\nabla F\|_\infty \leq L$!



Gradient Attenuation

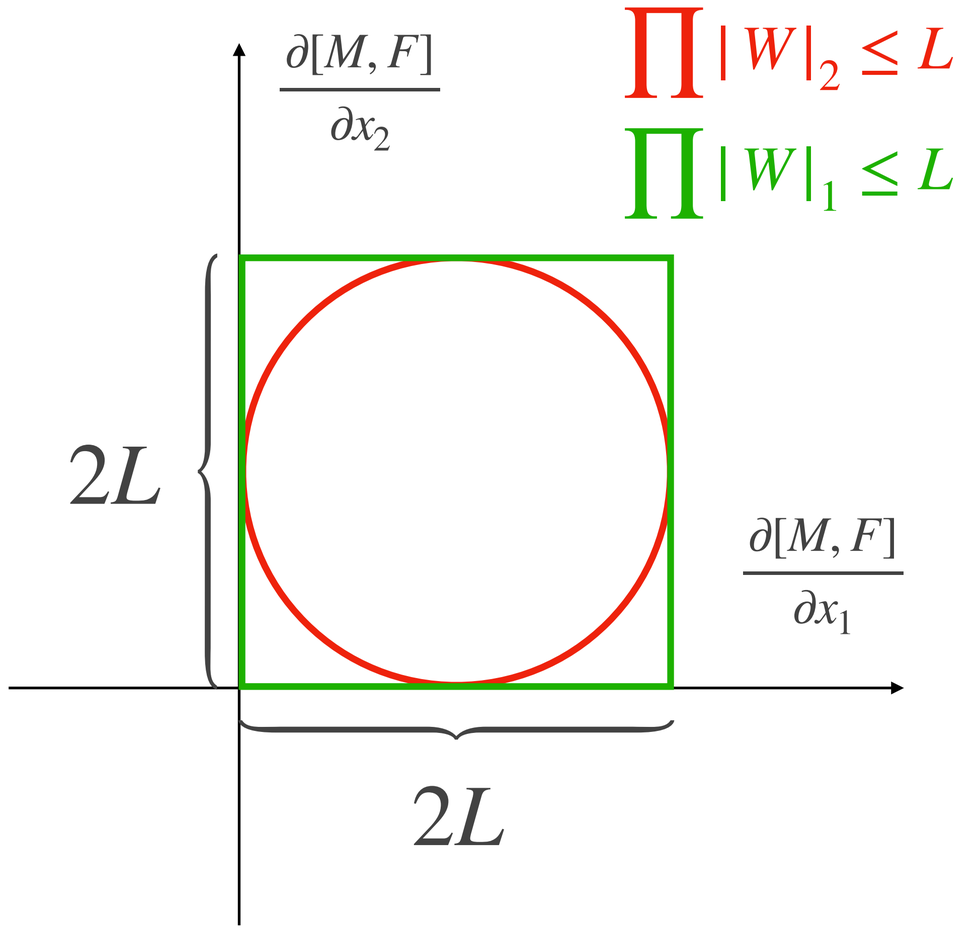
Universal Lipschitz Approximation

$$F(x) = W^{(2)} \cdot \sigma(W^{(1)} \cdot x + b^{(1)}) + b^{(2)}$$

$$F(x) = W^{(2)} \cdot \sigma(W^{(1)} \cdot x)$$

$$F(x) = (f_2 \circ \sigma \circ f_1)(x)$$

$$L[f_i] \leq 1 \leftarrow \text{over-constraining?}$$

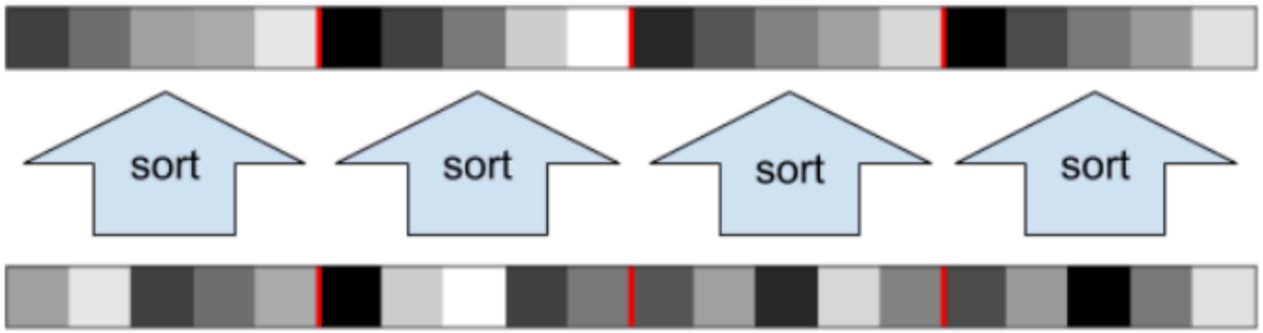


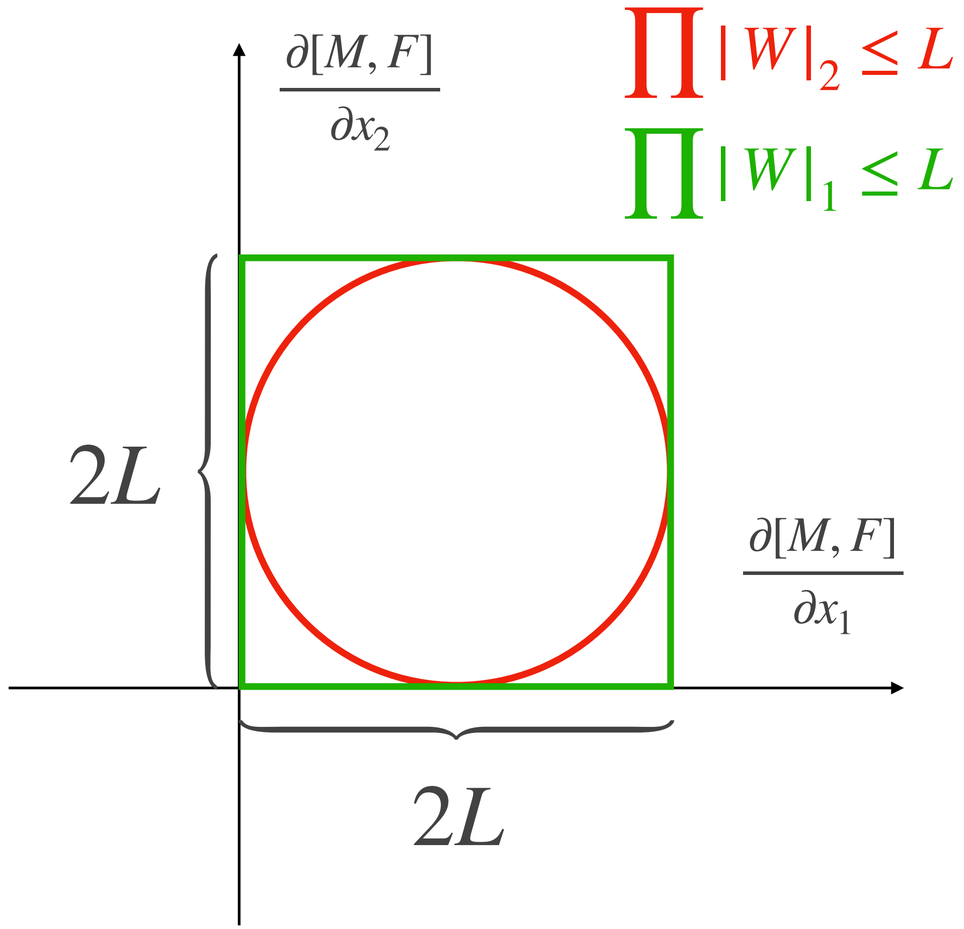
Activations

Activations need $\|\nabla \sigma(x)\| = 1 \quad \forall x$

Pointwise activations are useless!

Solution: GroupSort



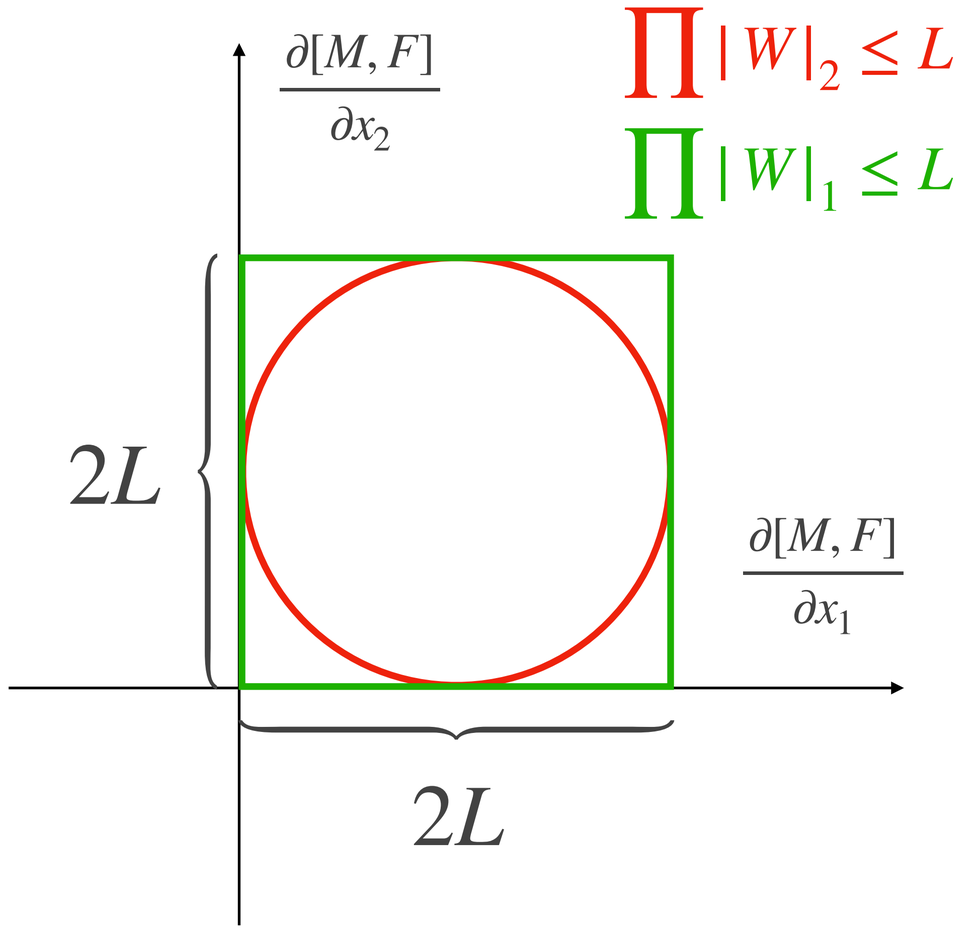


Summary - Monotonic Lipschitz Functions

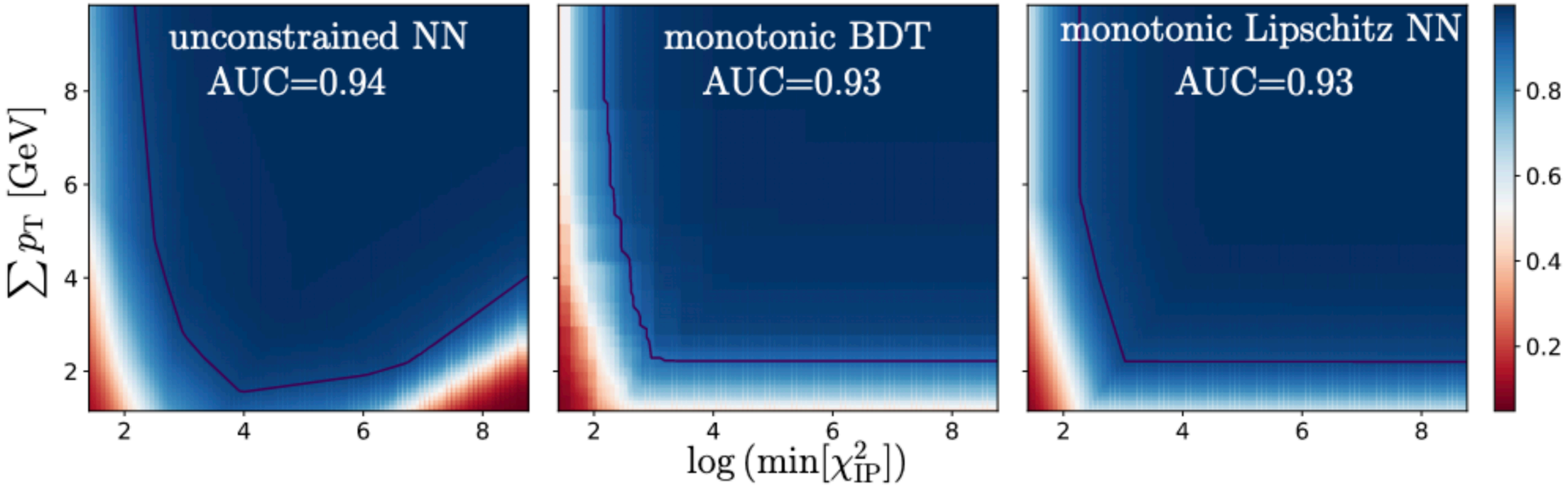
This architecture is

1. provably robust
2. provably monotonic
3. universally approximating the target function class
4. working well in practice

→ Implemented in the LHCb trigger for many major selections



HLT1 Inclusive b&c lines -- 2D subproblem



Energy Flow and Energy Movers Distance

Characterizing Collider Data

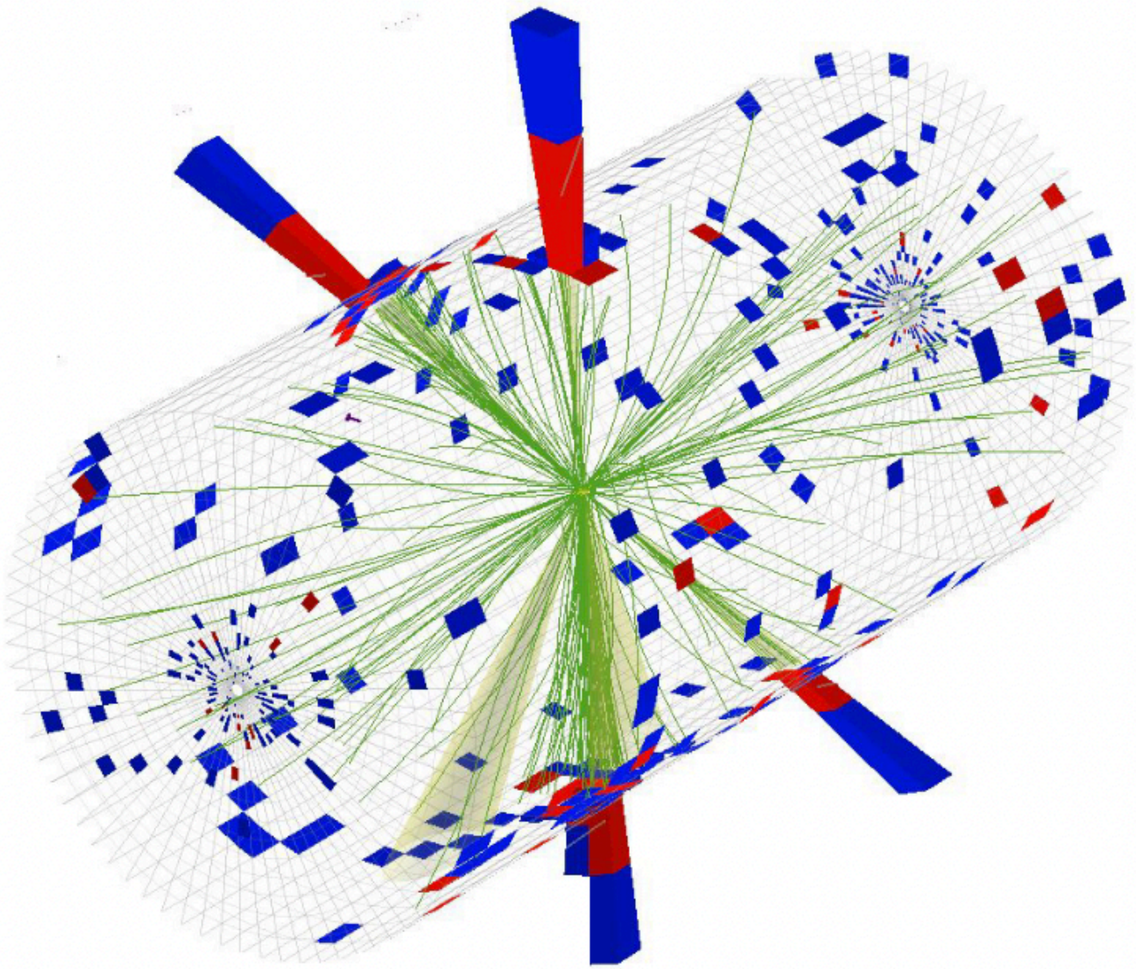
Goal:

"Feature engineering"

Represent collider data:

- Easy to understand
  Theoretically &
  Experimentally

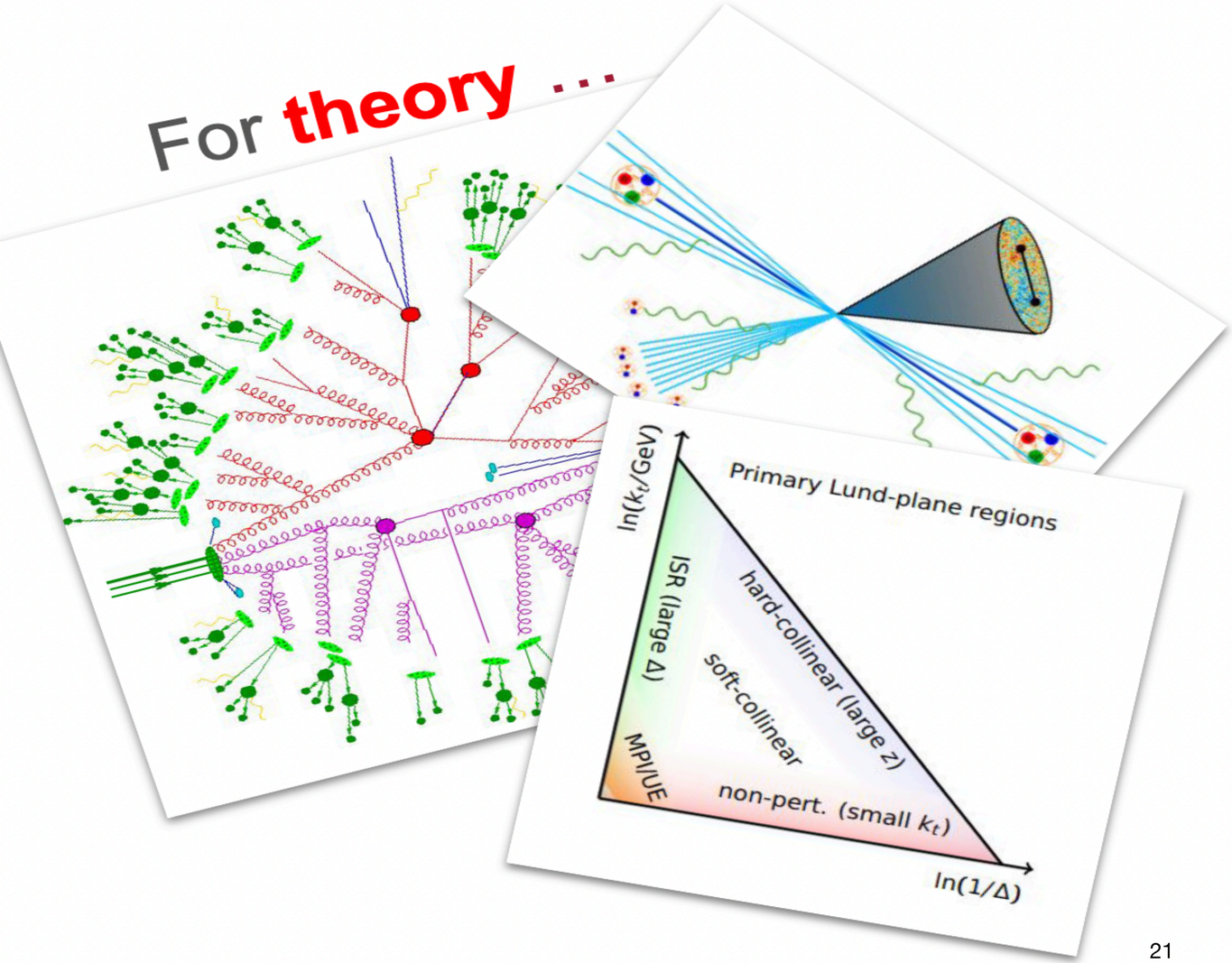
- Interpretable



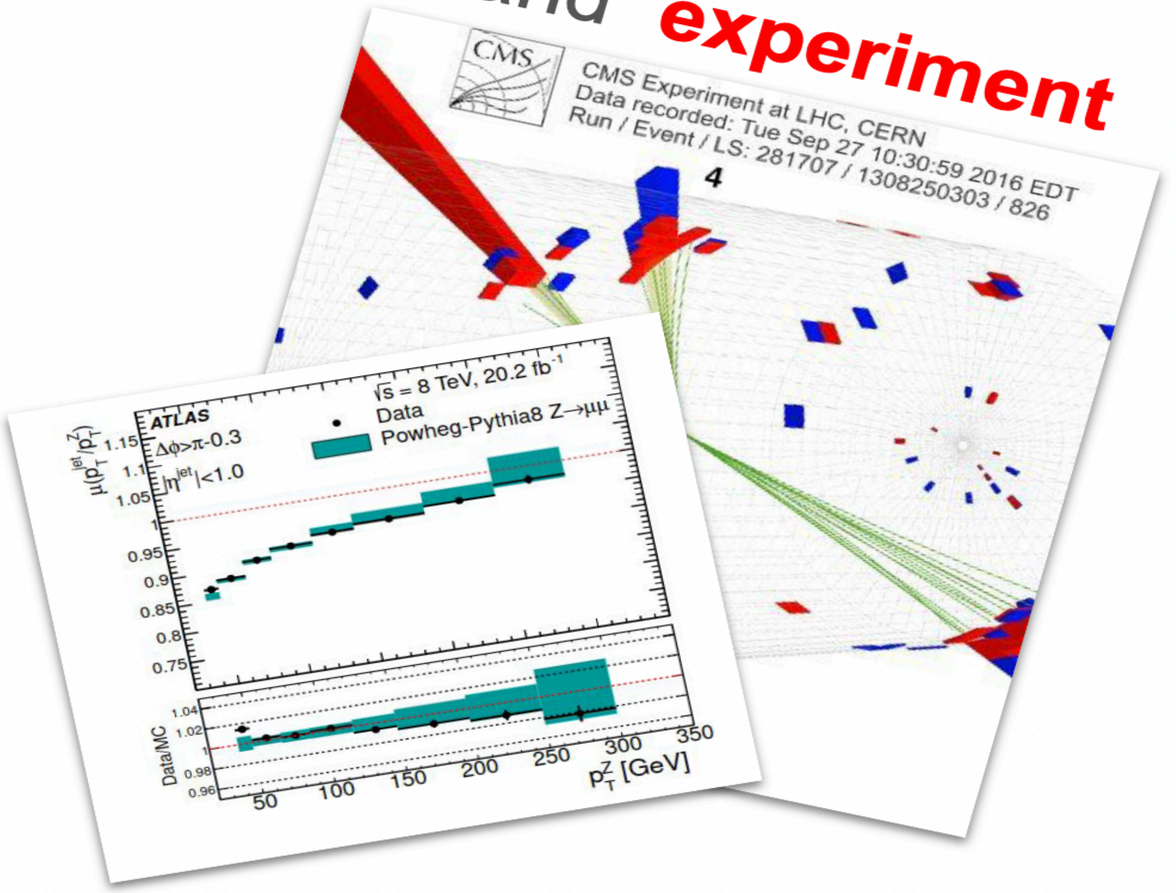
Images from [Bothmann et. al., 1905.09127;
Lee, Męcaj, Moult, 2205.03414;
Dreyer, Salam, Soyez, 1807.04758
CMS, 1810.10069;
ATLAS, 1703.10485]

We want ***Robust Observables***!

For **theory** …



and **experiment**



Slide stolen from Rikab Ghambir

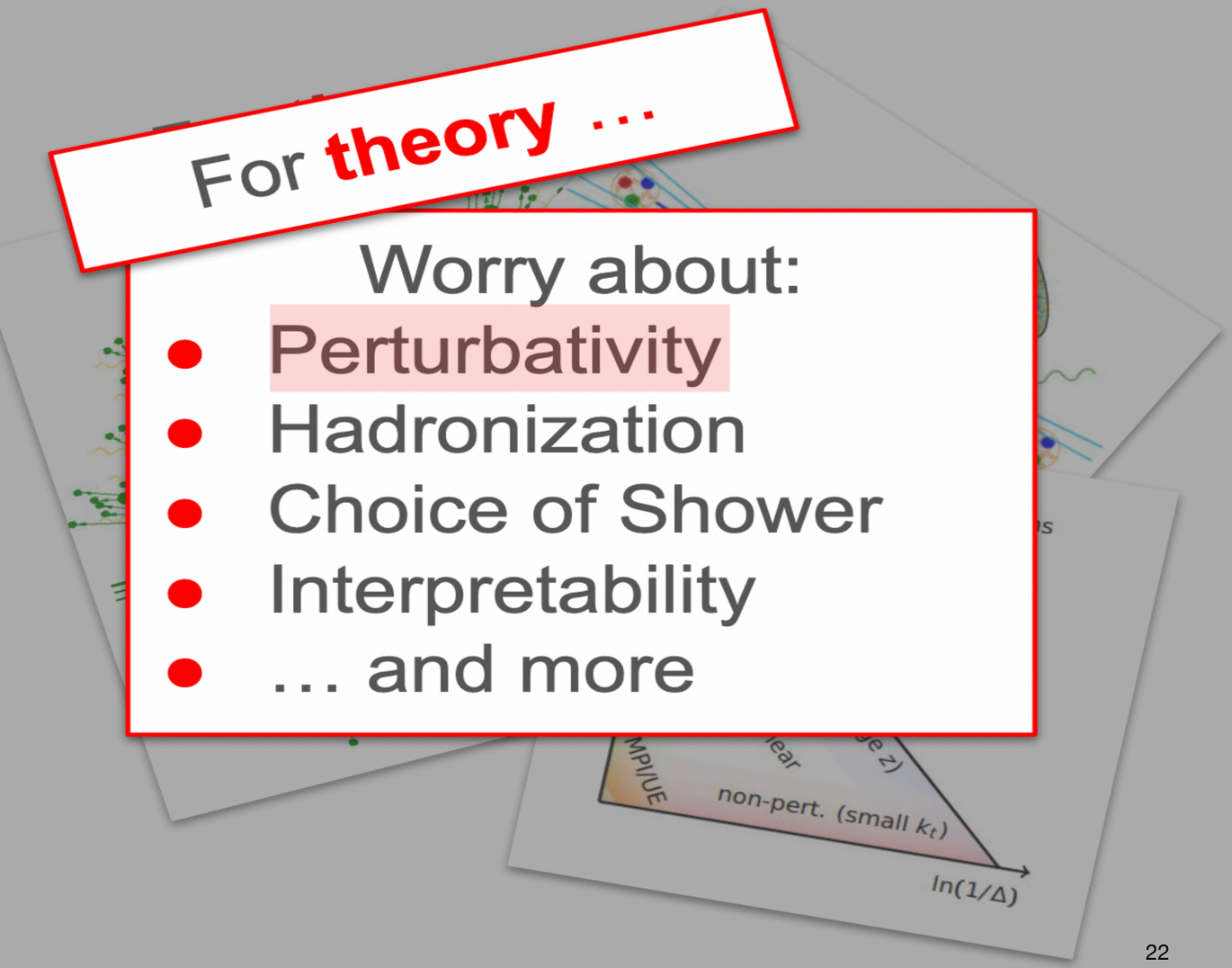
Images from [Bothmann et. al., 1905.09127;
Lee, Męcaj, Moult, 2205.03414;
Dreyer, Salam, Soyez, 1807.04758
CMS, 1810.10069;
ATLAS, 1703.10485]

We want ***Robust Observables***!

For **theory** …

Worry about:

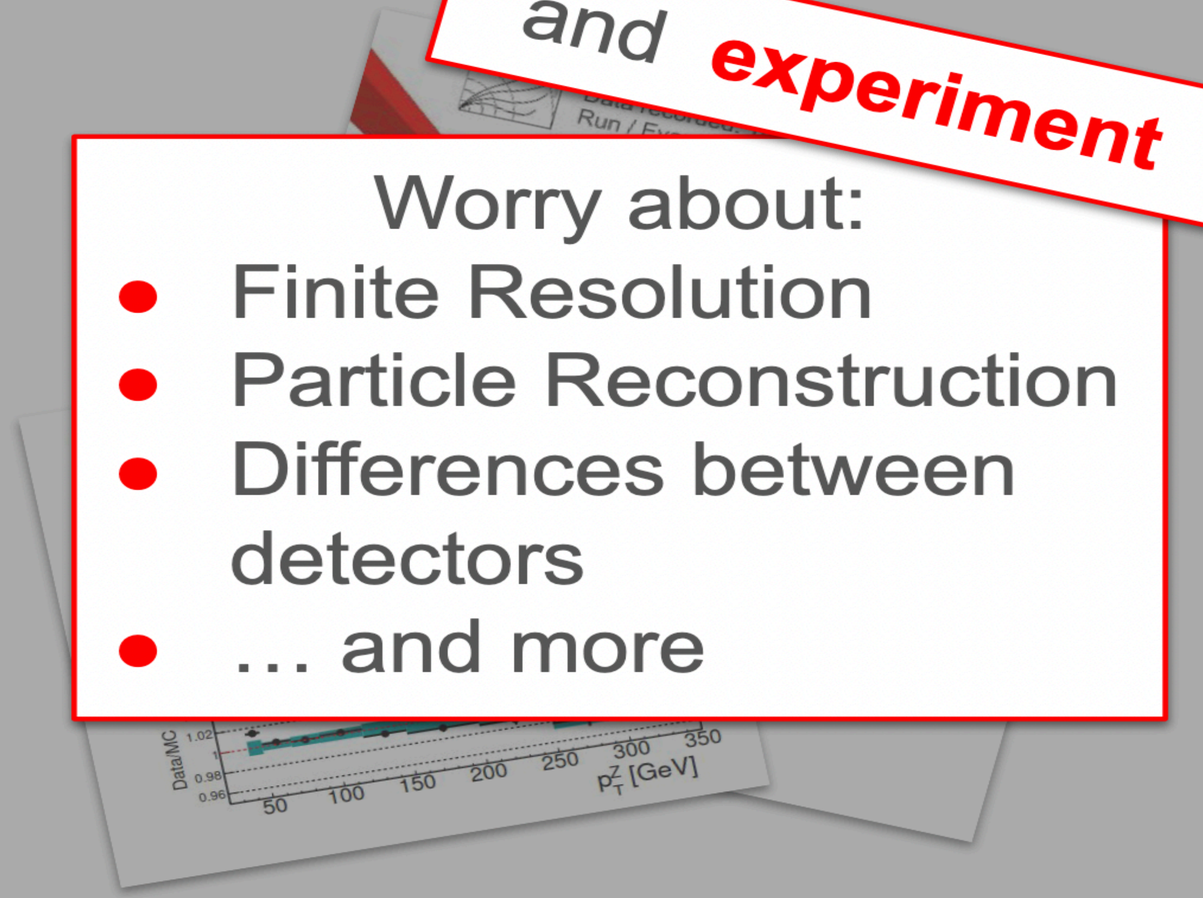
- Perturbativity
- Hadronization
- Choice of Shower
- Interpretability
- … and more



and **experiment**

Worry about:

- Finite Resolution
- Particle Reconstruction
- Differences between detectors
- … and more



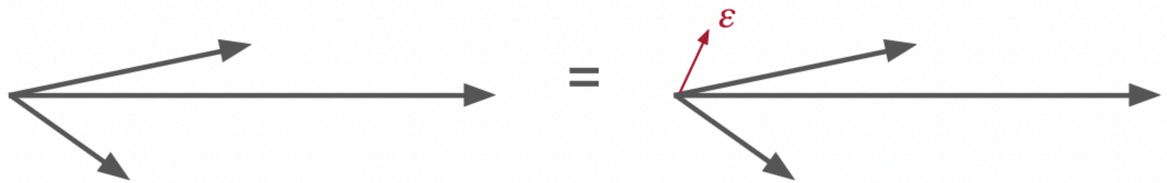
Slide stolen from Rikab Ghambir

Cannot calculate every observable :(

IRC divergence messes with perturbative calculations!

What we need: IRC safety

Infrared Safety: An observable is unchanged under a soft emission



Collinear Safety: An observable is unchanged under a collinear splitting

Slide stolen from Rikab Ghambir

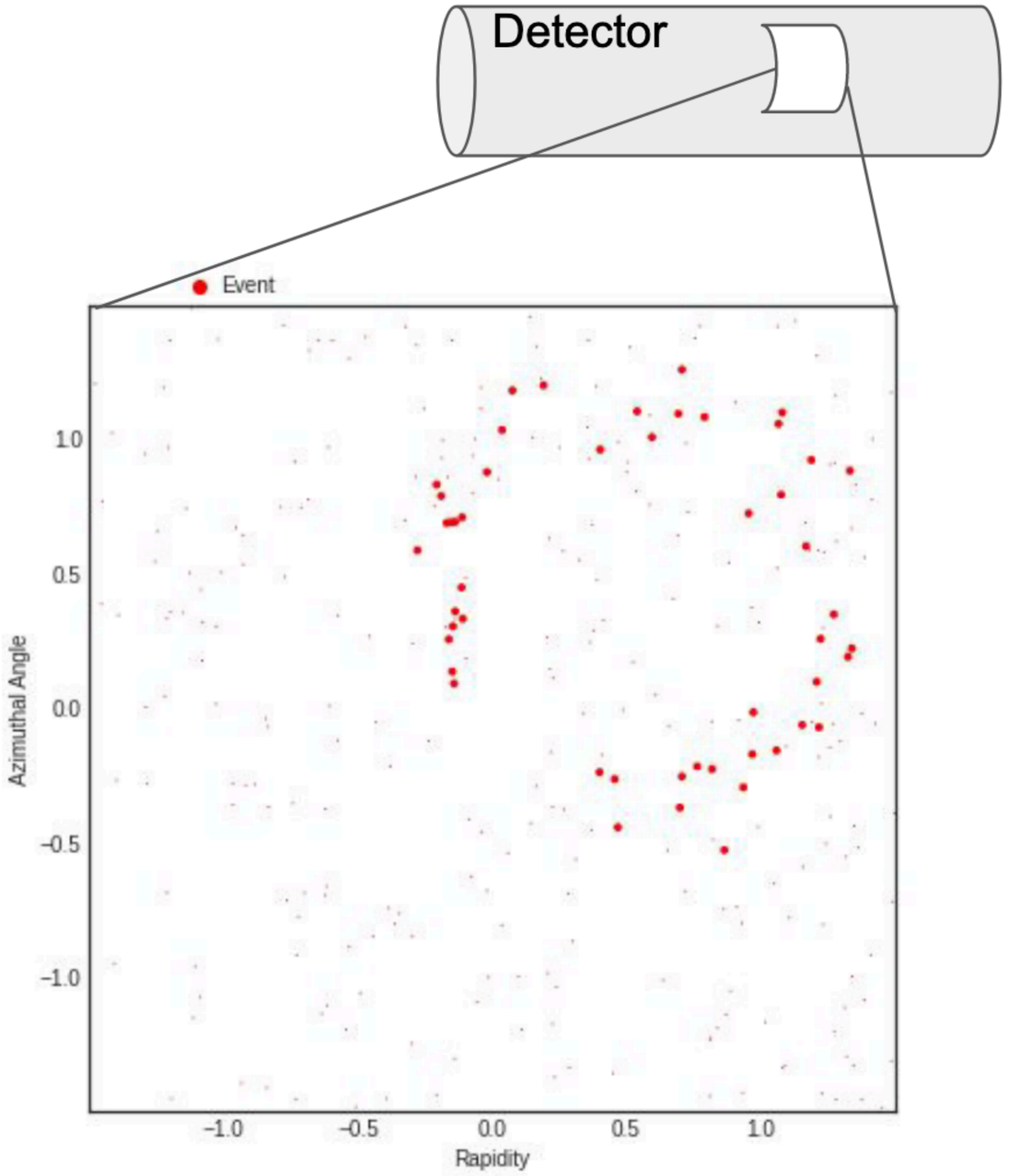
A measure: Energy Flow

$$\mathscr{E}(x) = \sum_i E_i \, \delta(x - x_i)$$

here: 2D projection on (y, ϕ)

IRC safe!

+ contains all IRC safe info!

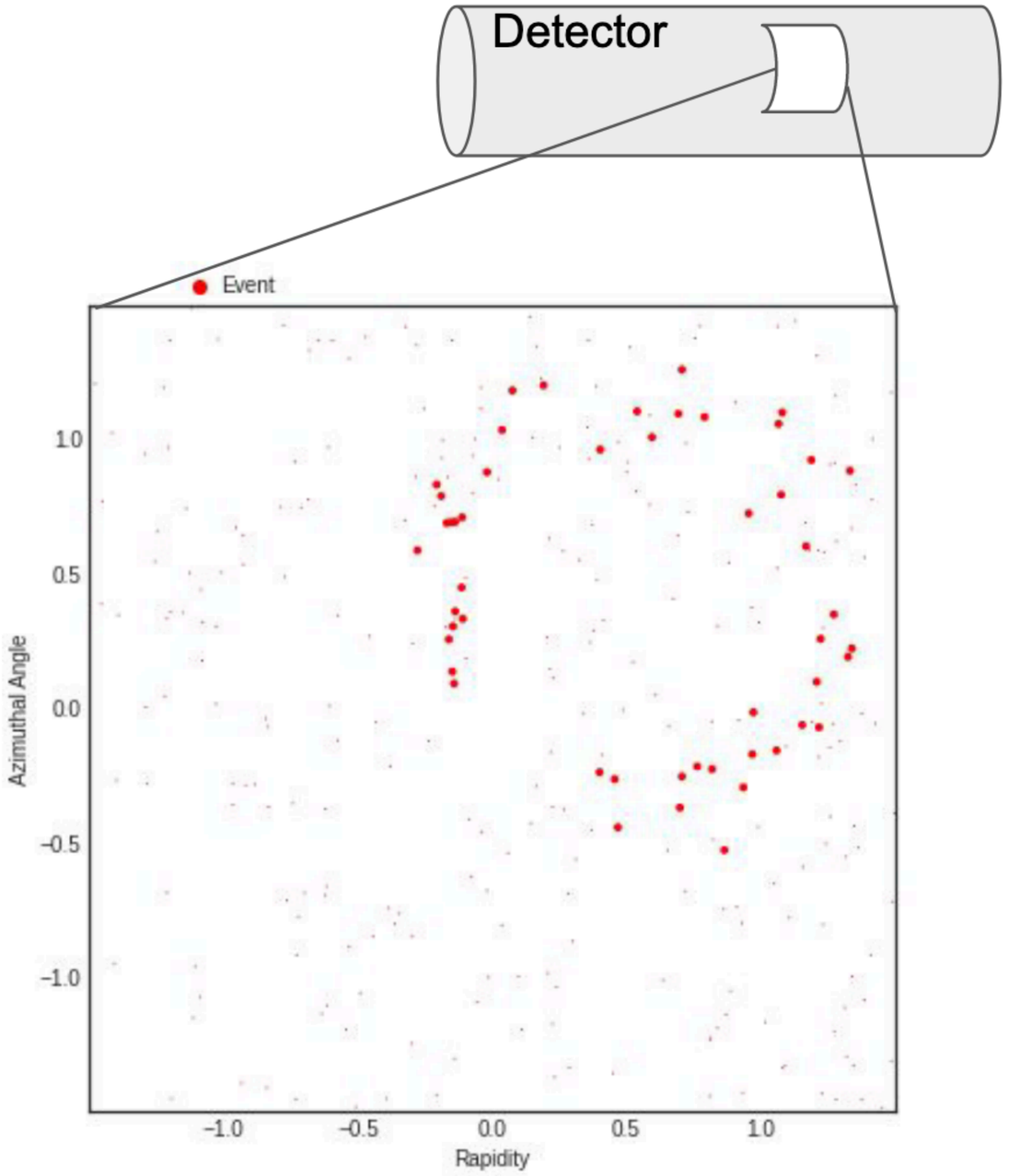


Comparison?

All the relevant information in Energy Flow

Now we want to compare events and define observables!

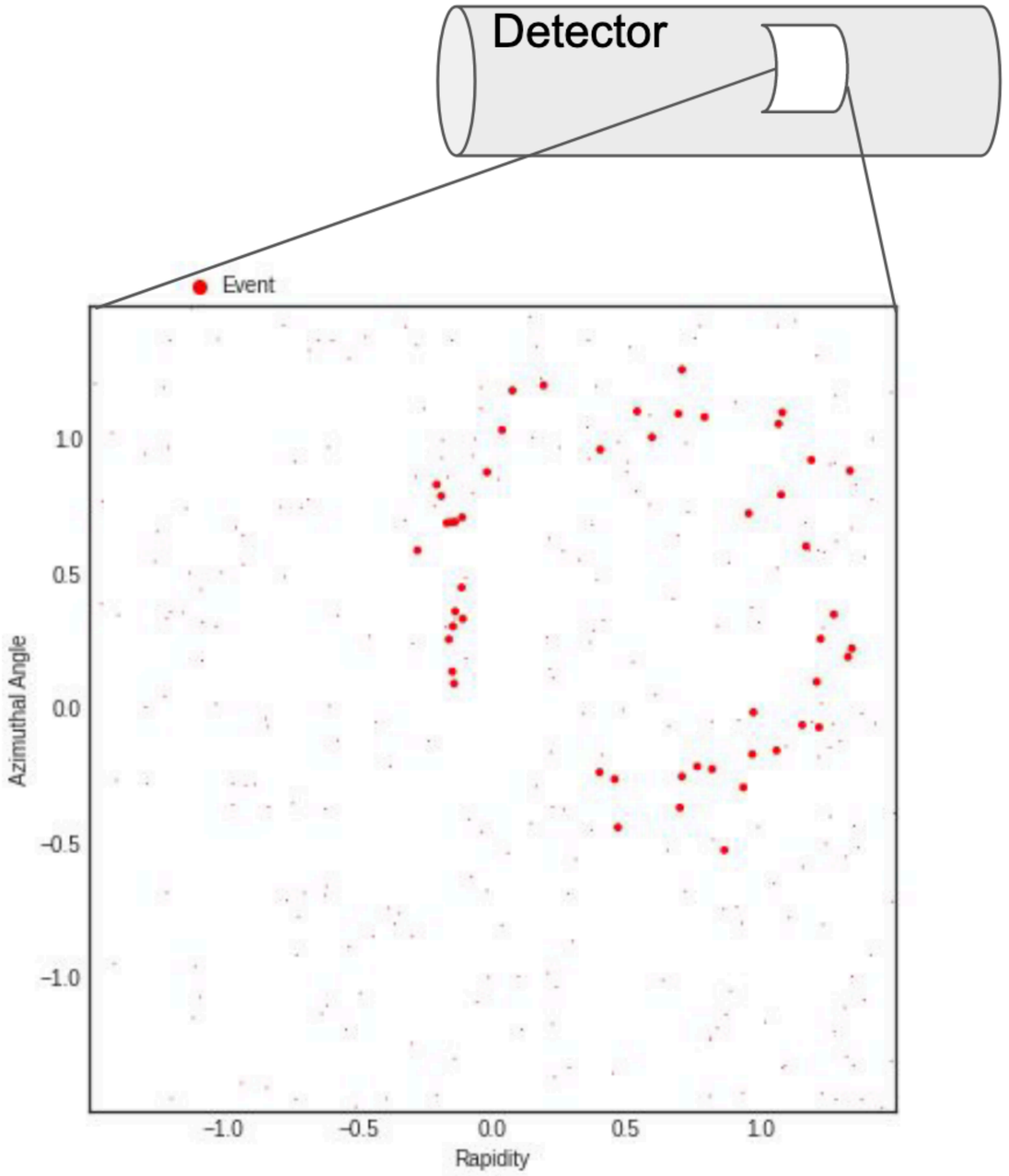
Need Metric/Distance!



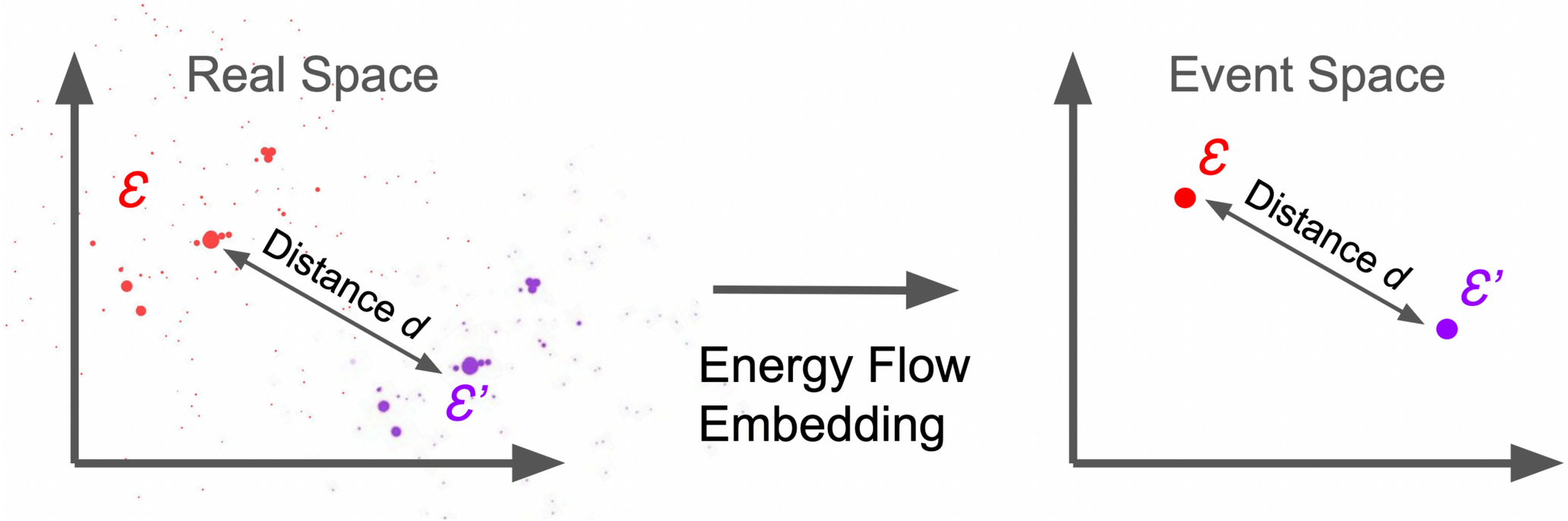
Comparison?

Requirements

1. A proper metric
2. IRC safe
3. faithfully lifting the detector metric



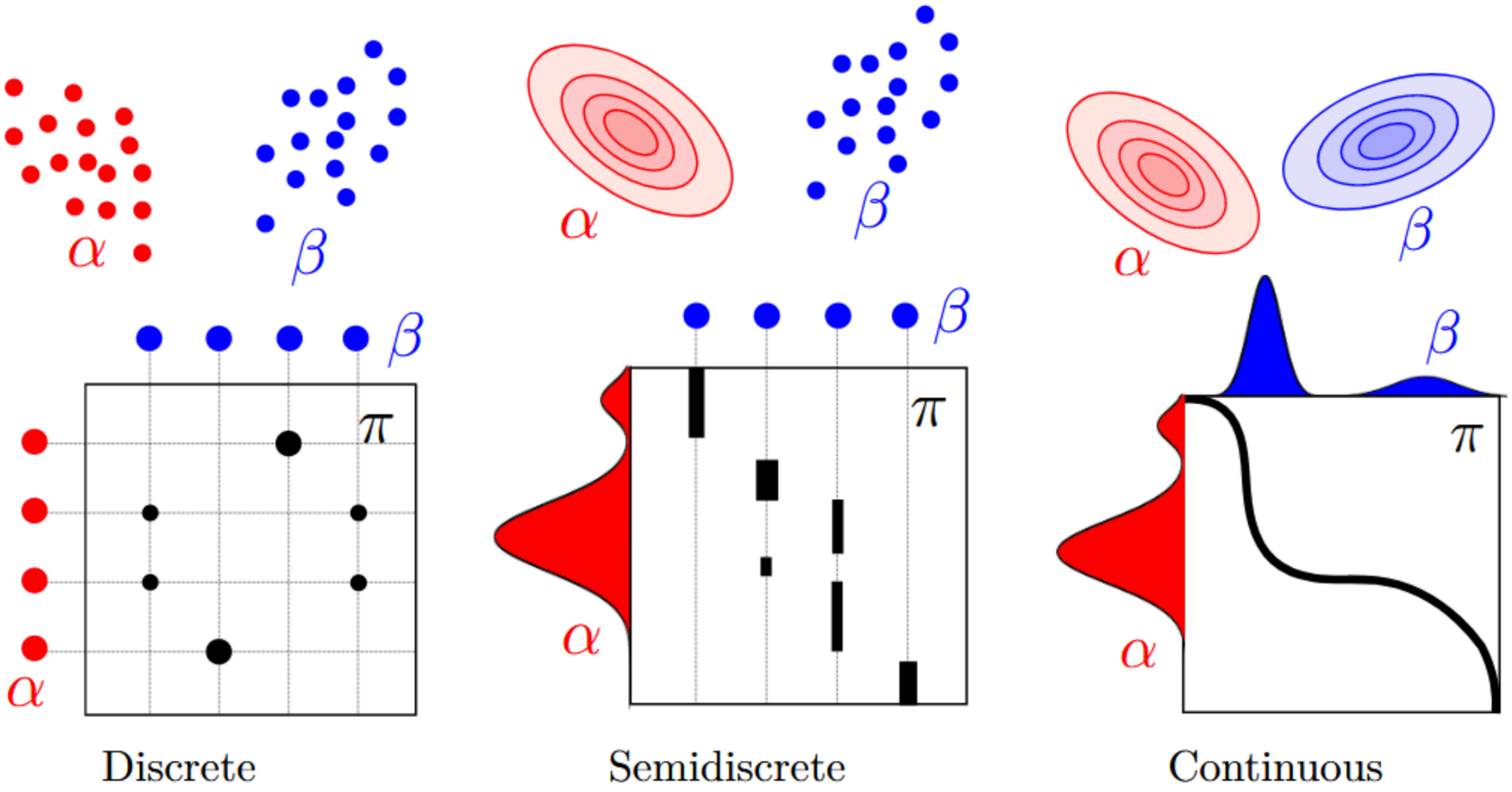
Faithful representation



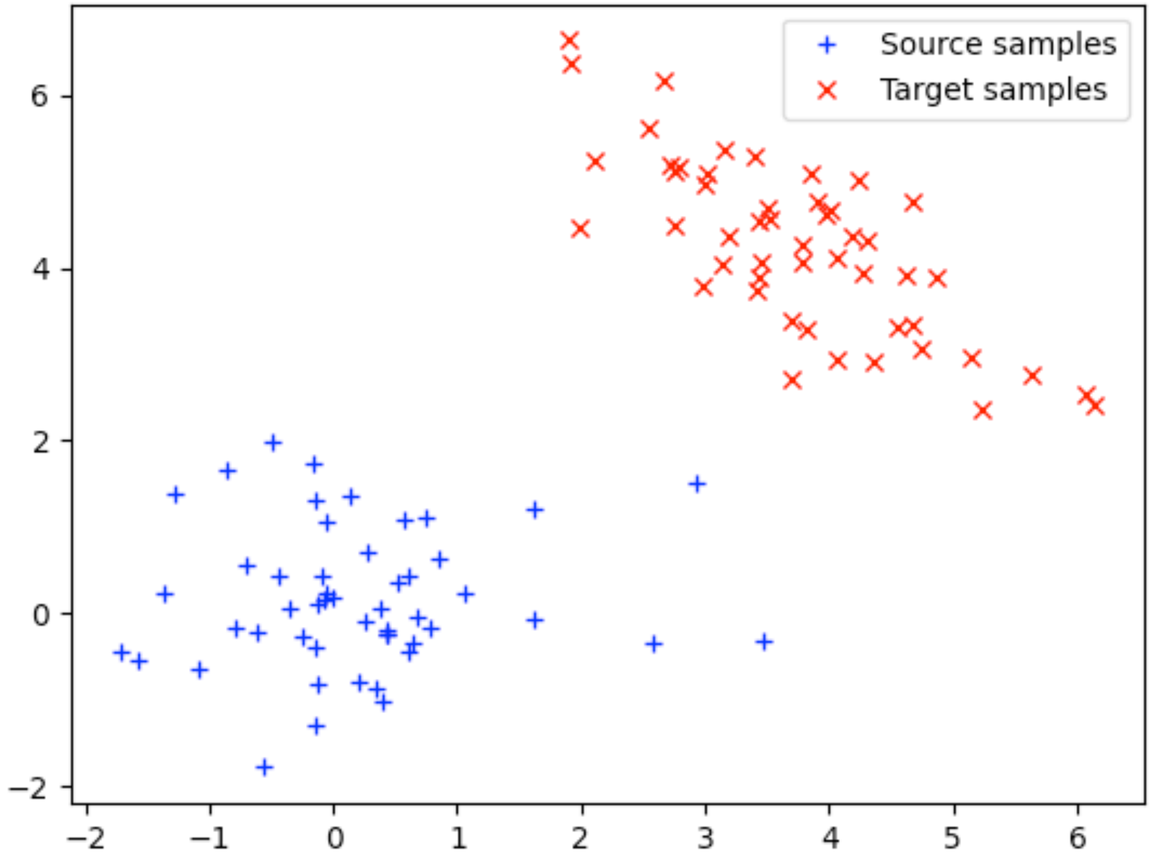
Slide stolen from Rikab Ghambir

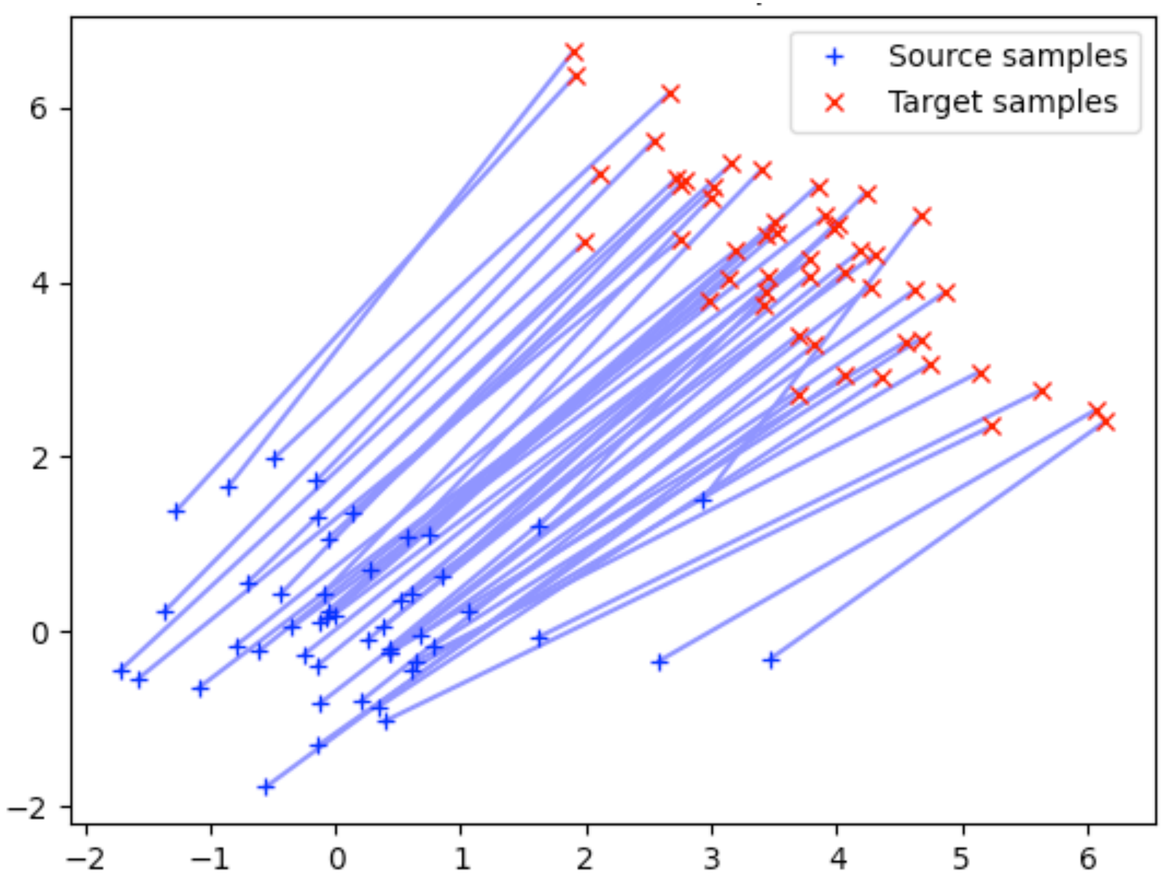
Optimal Transport - Earth Movers Distance

Optimal Transport - Earth Movers Distance



Optimal Transport - Earth Movers Distance





Optimal Transport - Earth Movers Distance

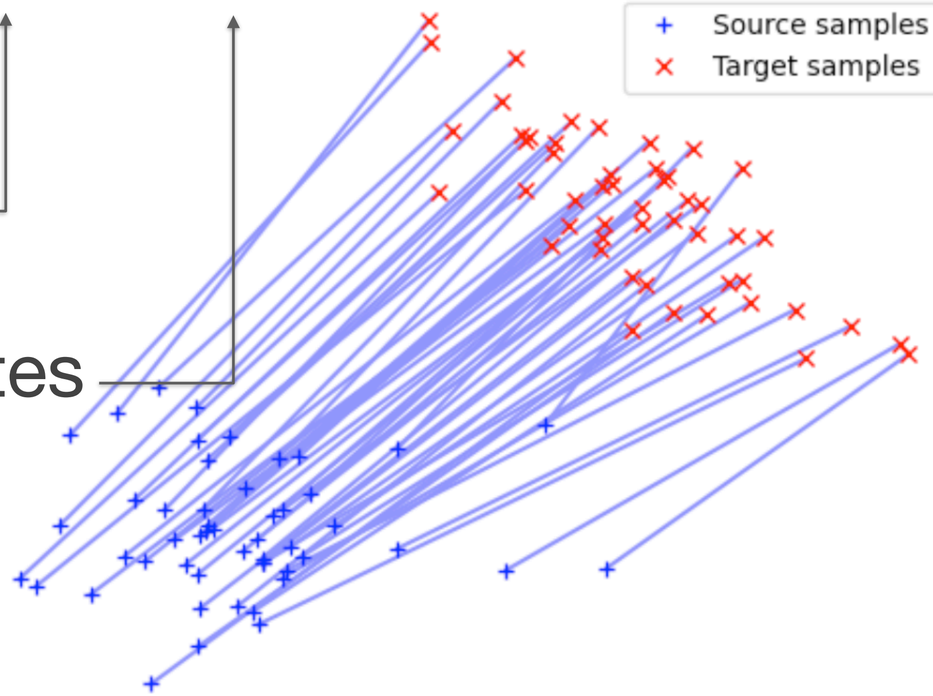
$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y)\sim\pi}\left[\|x - y\|\right]$$

Distribution 1

Distribution 2

Joint Distribution

Coordinates



Energy Movers Distance

$$\mathrm{EMD}(\mathcal{E}, \mathcal{E}') = \min_{\{f_{ij} \geq 0\}} \sum_{ij} f_{ij} \frac{\theta_{ij}}{R} + \left| \sum_i E_i - \sum_j E'_j \right|$$

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y)\sim\pi} \left[\|x - y\| \right]$$

Event 1

Event 2

Energies

Particle Coordinates

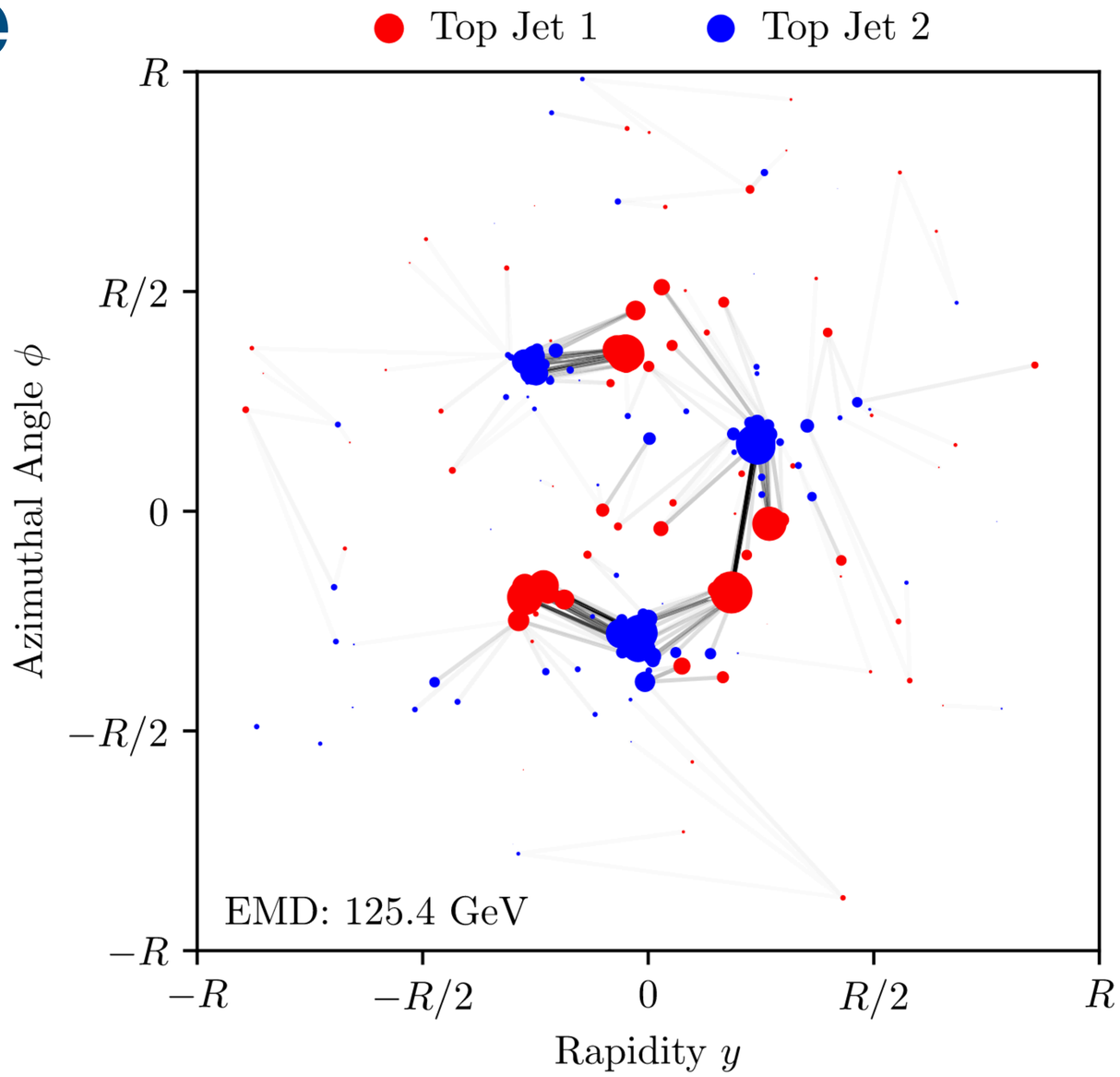
Energy Movers Distance

1. A proper metric

2. IRC safe

3. faithfully lifts the detector metric

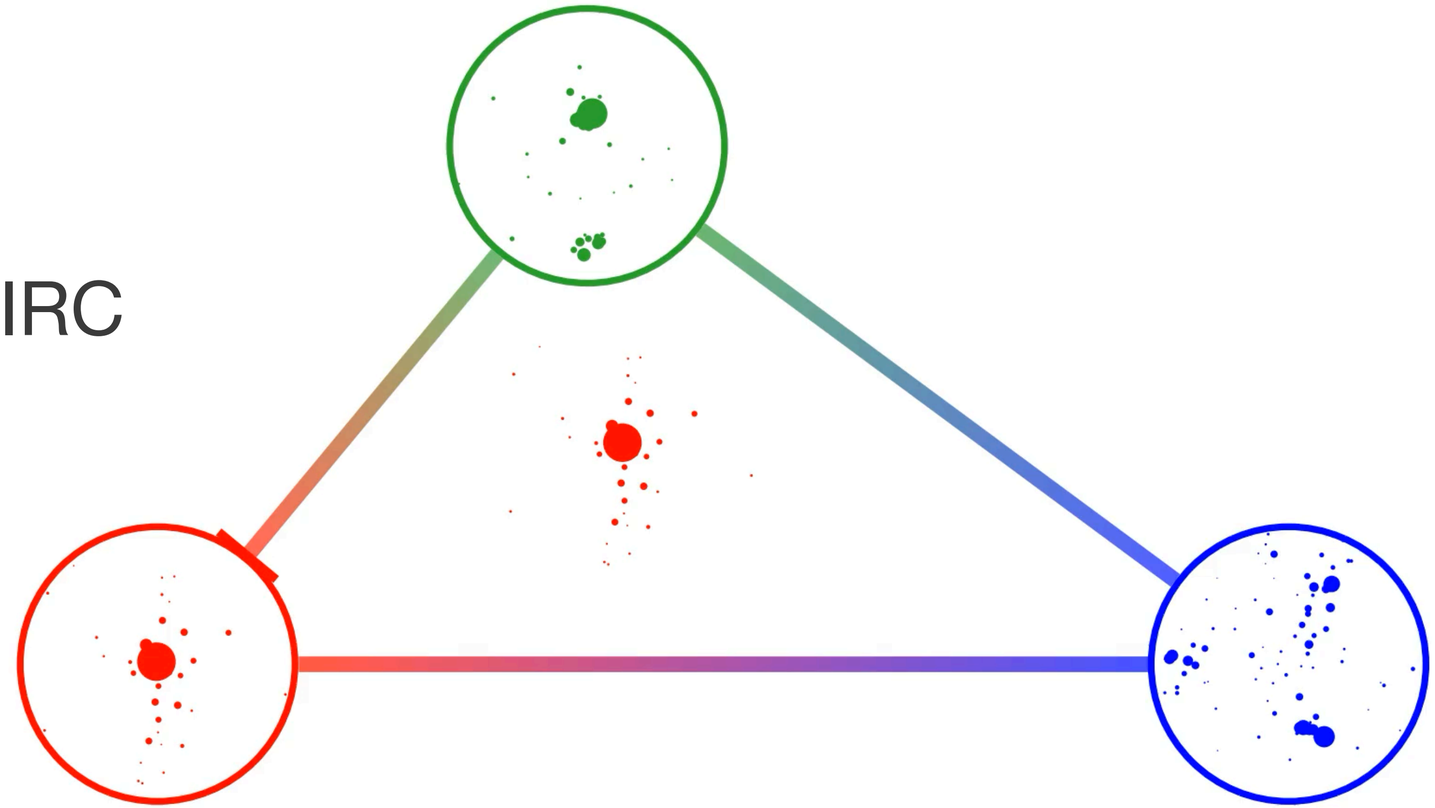
It is the only metric that has all 3!



Comparison?

Have a metric!

Can compare events (with all IRC safe info)



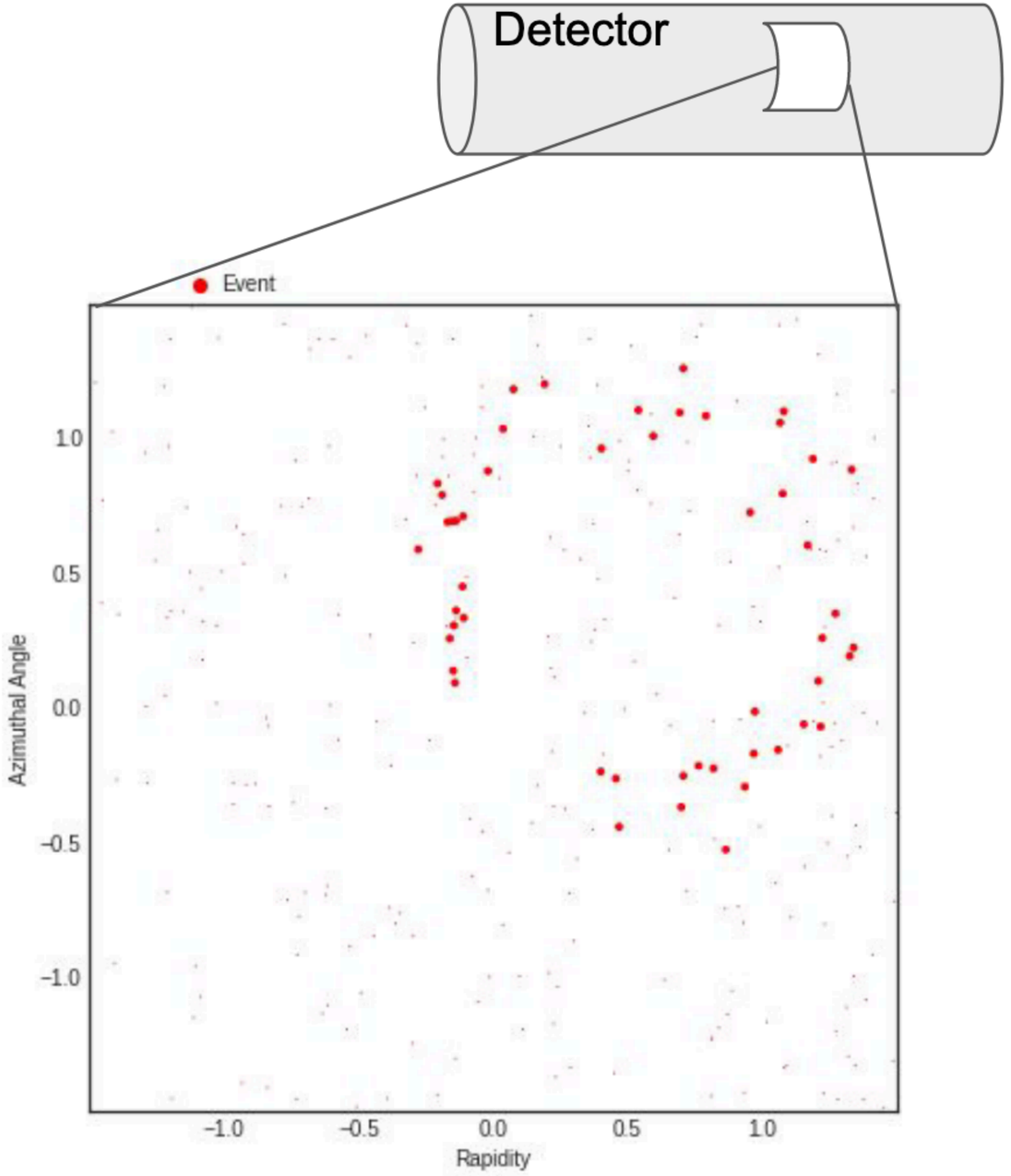
From https://energyflow.network/docs/emd/

Comparison?

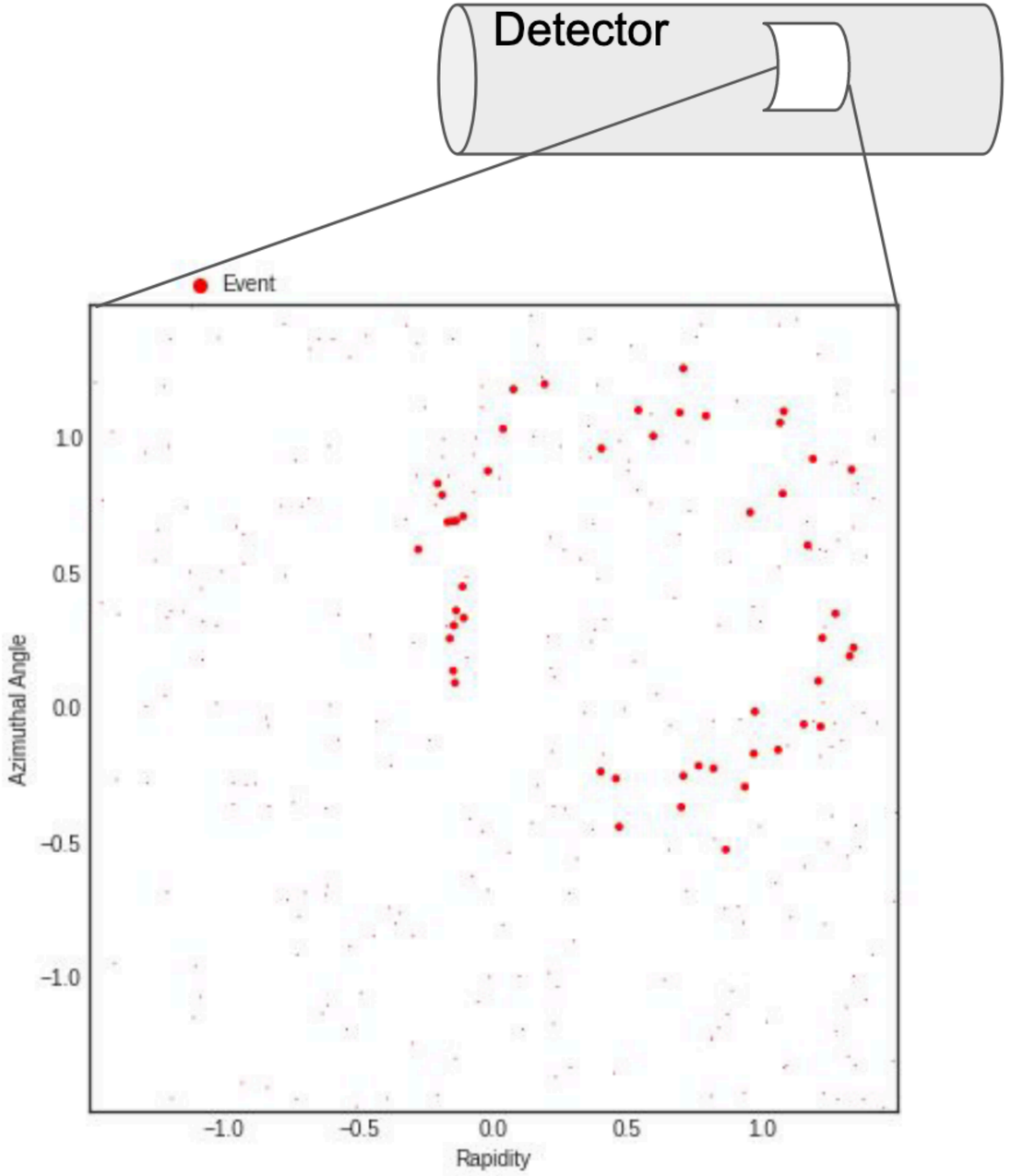
Have a metric!

Can compare events (with all IRC safe info)

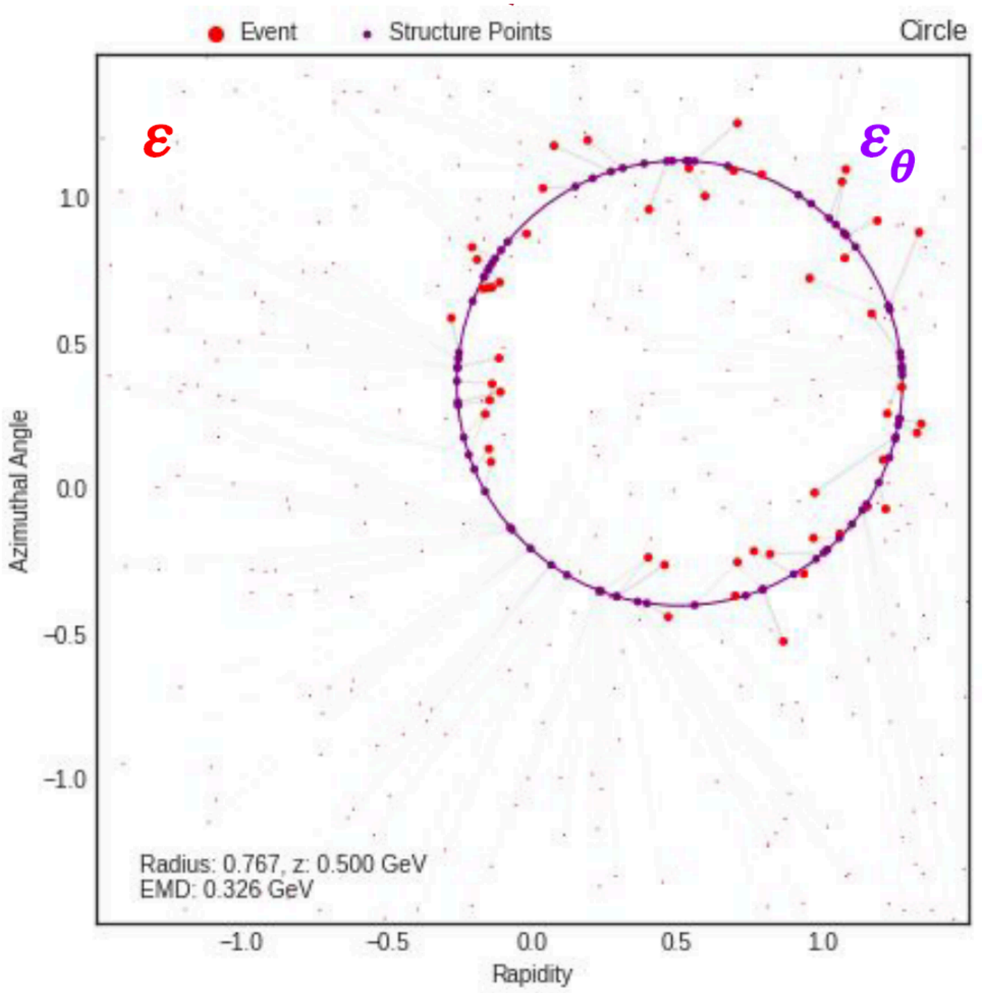
Can define observable as distance to some shape



Question: What shape is that?



Answer: It is a circle!

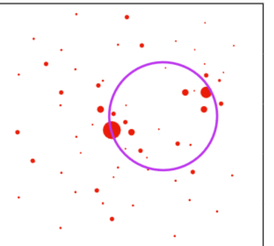
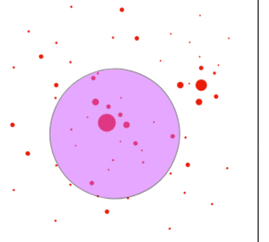
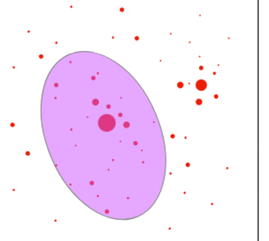
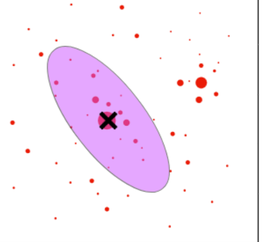
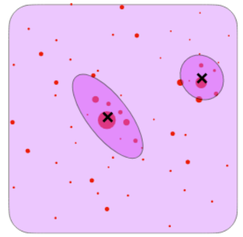


Observables : old & new

- Thrust
- Spherocity
- Broadening
- N-jettiness

Even jet clustering algorithms:

- XCone
- Cambridge Aachen
- (anti-) kt

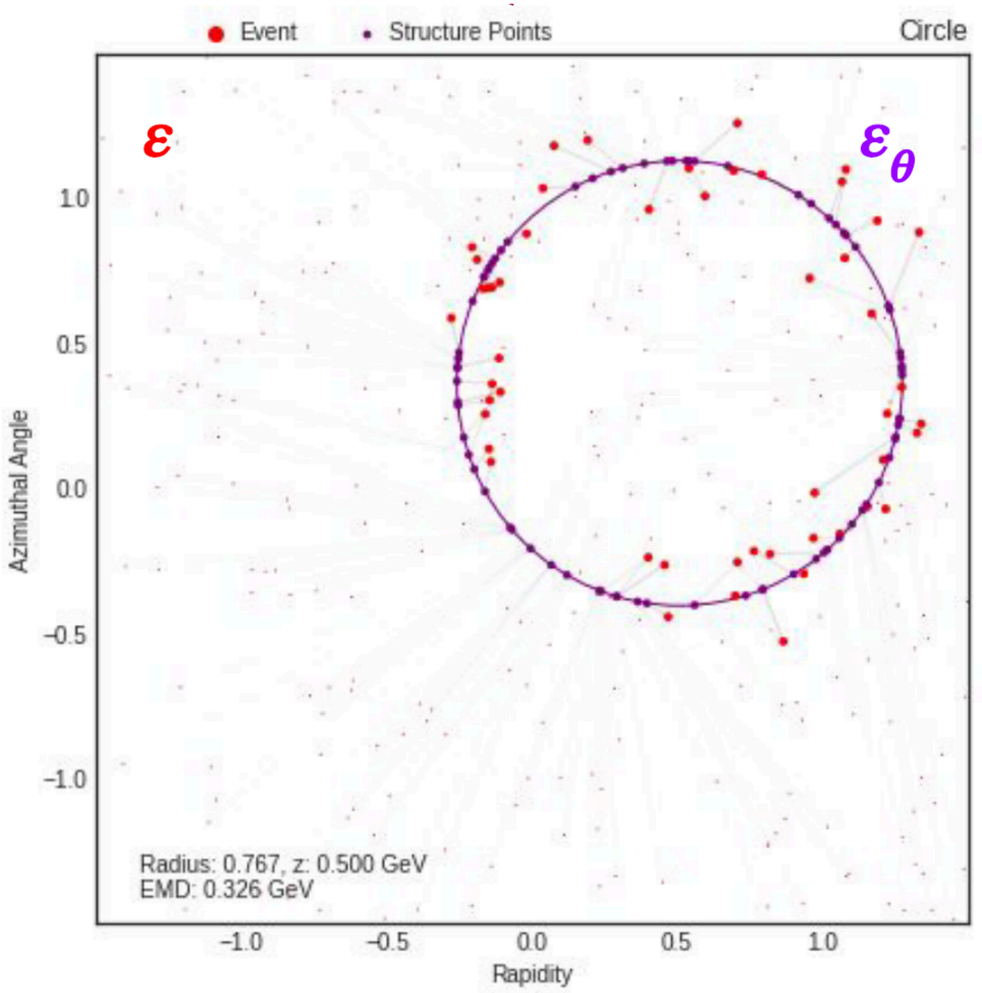
Shape	Specification	Illustration
Ringiness $\mathcal{O}_R$	**Manifold of Rings** $\mathcal{E}_{x_0,R_0}(x) = \frac{1}{2\pi R_0}$ for $\lvert x - x_0\rvert = R_0$ x_0 = Center, R_0 = Radius	
Diskiness $\mathcal{O}_D$	**Manifold of Disks** $\mathcal{E}_{x_0,R_0}(x) = \frac{1}{\pi R_0^2}$ for $\lvert x - x_0\rvert \leq R_0$ x_0 = Center, R_0 = Radius	
Ellipsiness $\mathcal{O}_E$	**Manifold of Ellipses** $\mathcal{E}_{x_0,a,b,\varphi}(x) = \frac{1}{\pi ab}$ for $x \in \text{Ellipse}_{x_0,a,b,\varphi}$ x_0 = Center, a, b = Semi-axes, φ = Tilt	
(Ellipse +Point)iness	**Composite Shape** $\mathcal{O}_E \oplus \tau_1$ Fixed to same center x_0	
N-(Ellipse +Point)iness +Pileup	**Composite Shape** $N \times (\mathcal{O}_E \oplus \tau_1) \oplus \mathcal{I}$	

arxiv:2302.12266

Problem!

Where exactly is the circle?

$$\min_{\theta} \text{EMD} = \min_{\theta} \min_{\pi} \mathbb{E}(\|x - y\|)$$

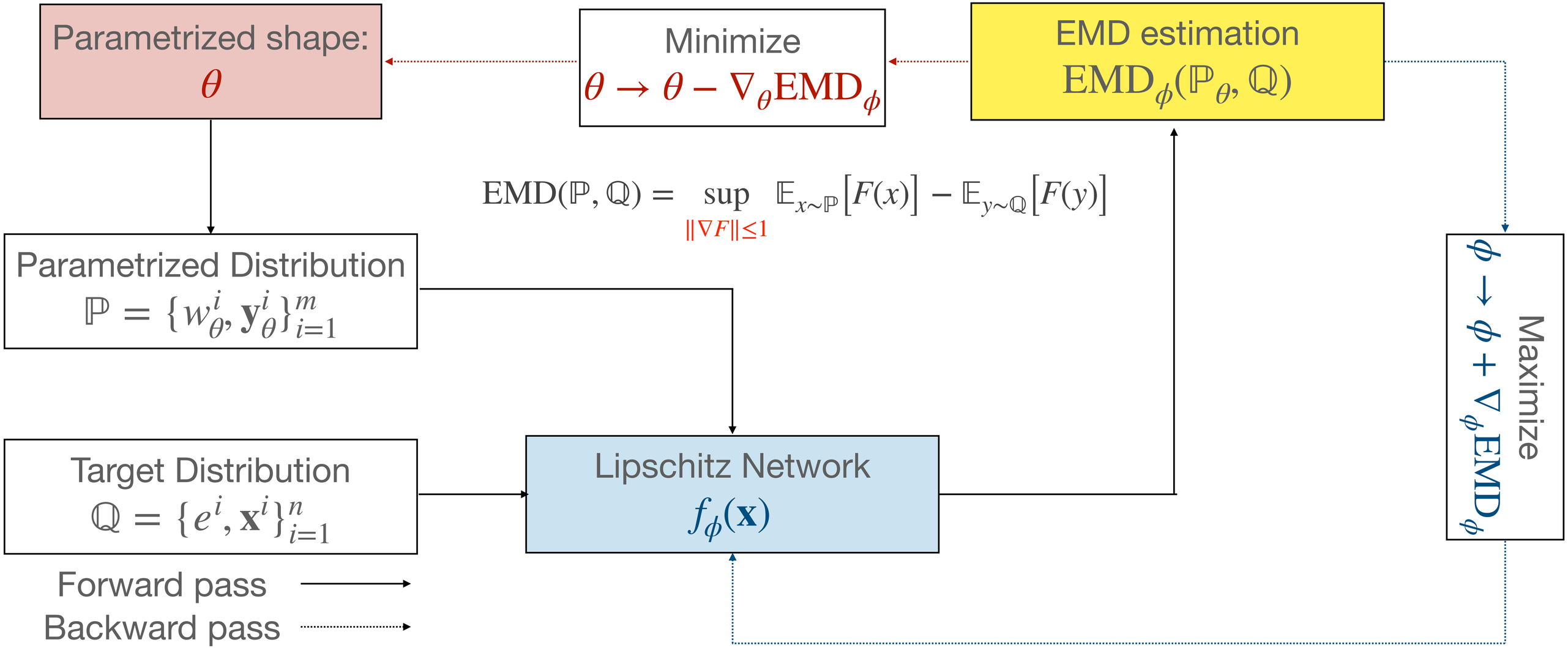


Kantorovich-Rubenstein Dual Formulation

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \sup_{\|\nabla F\| \leq 1} \mathbb{E}_{x \sim \mathbb{P}}\left[F(x)\right] - \mathbb{E}_{y \sim \mathbb{Q}}\left[F(y)\right]$$

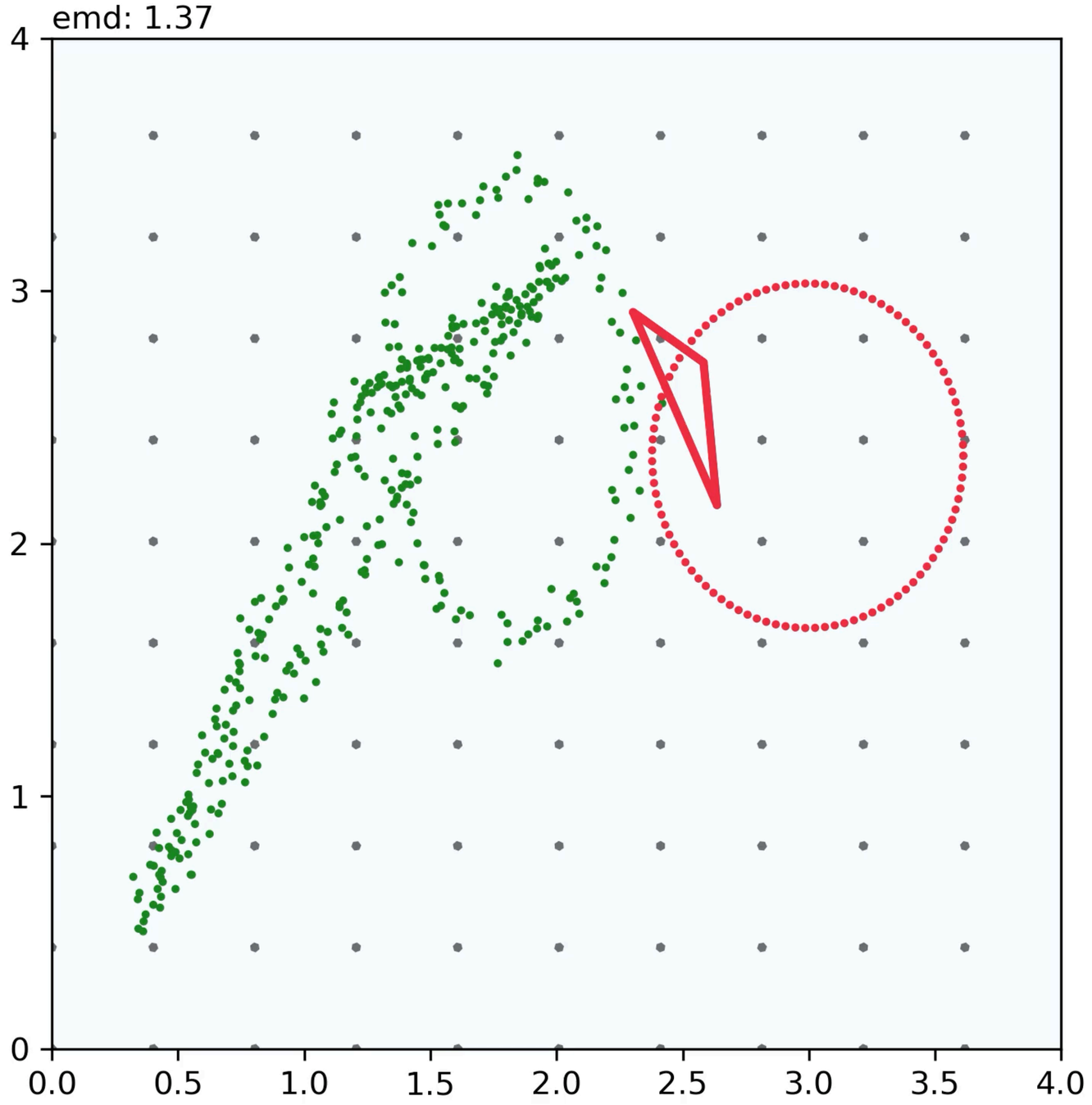
Kantorovich potential

Joint EMD Optimization



What shape is that?

Answer via geometric fitting!

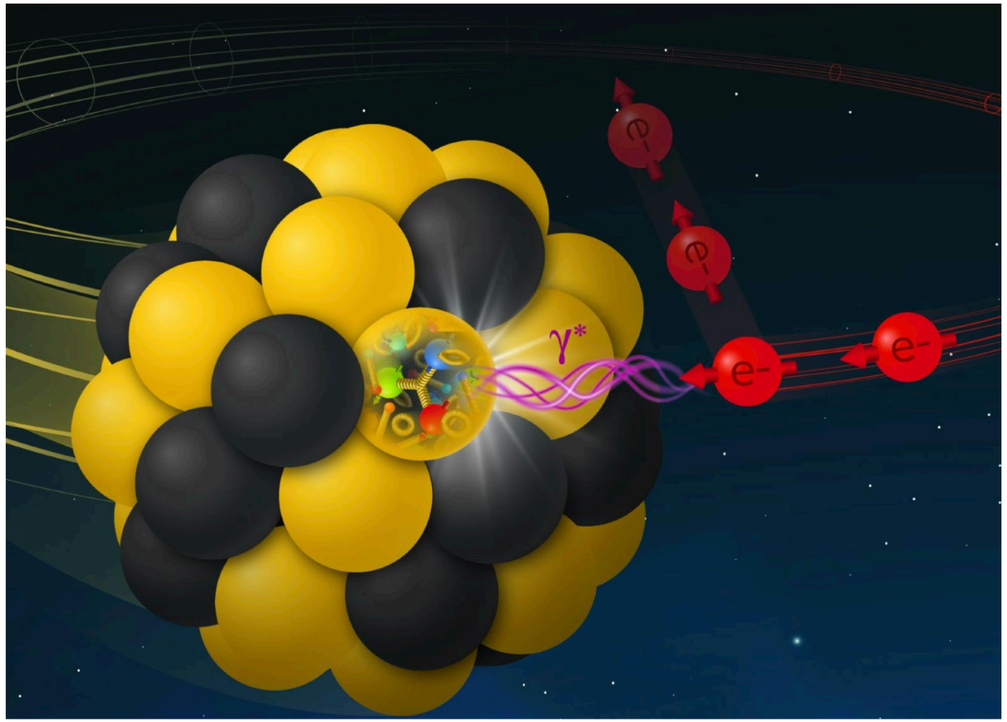


A new playing field

Unifying many observables!

Find new useful ones with this framework

A new playing field: the EIC
1. Electron Ion vs. Proton Proton
2. Lab Frame vs. Breit Frame



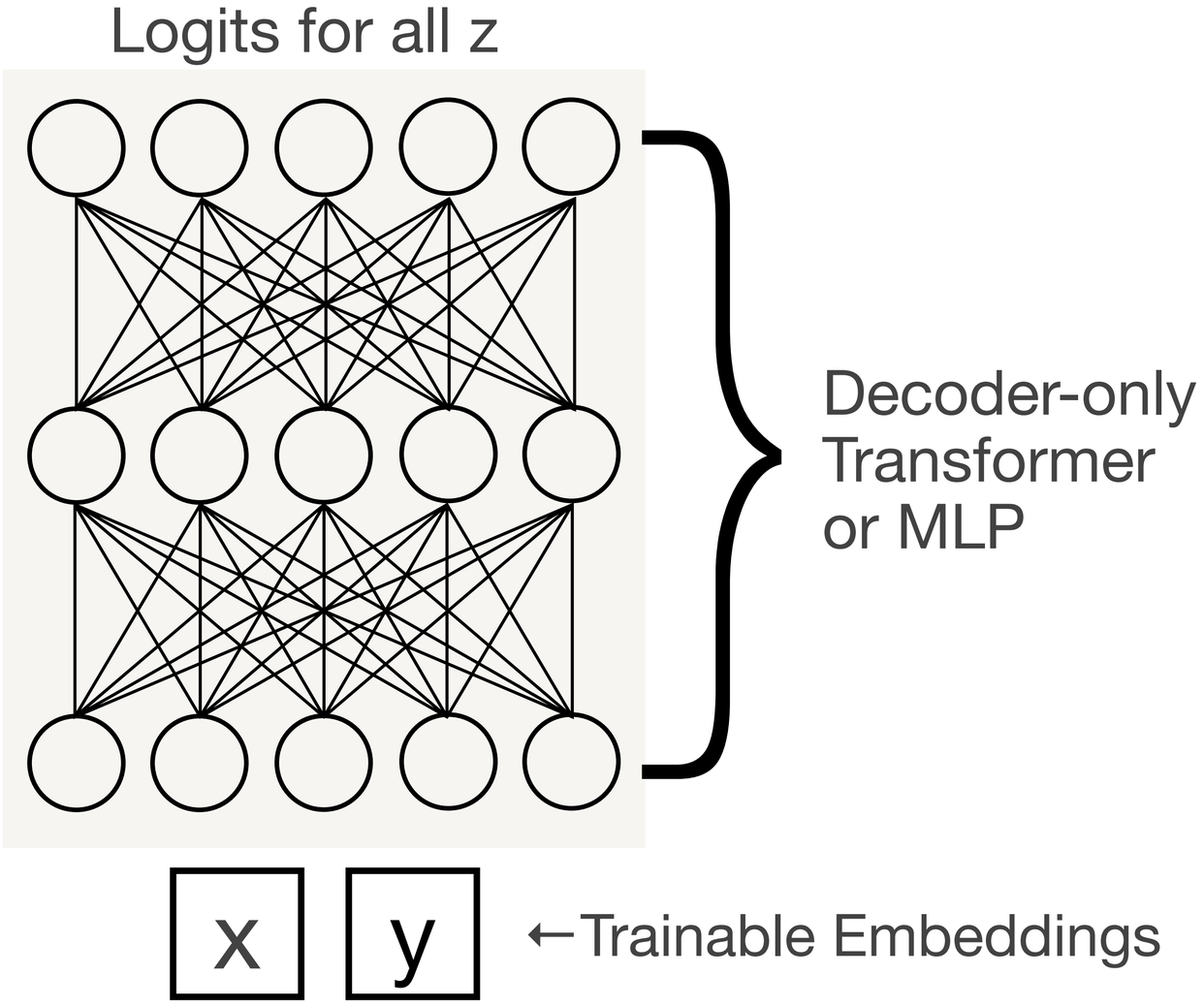
Find new observables that yield high information/discrimination!

Generalization and Emergent Capabilities: Grokking

The Task

Learn a binary operation

$$x \circ y = z$$



The Data

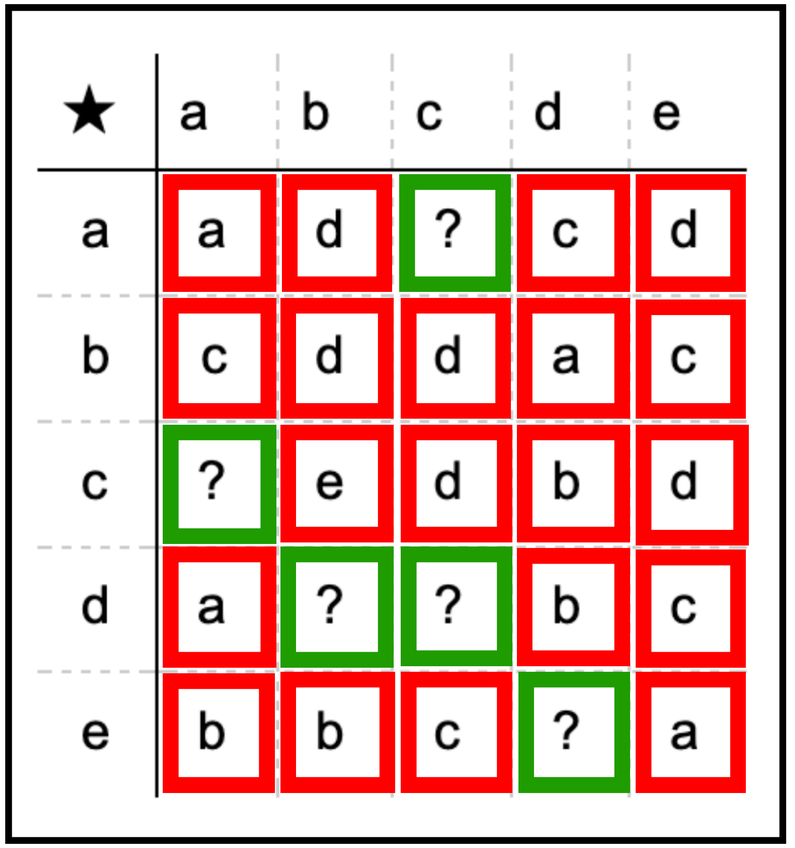
2D Table of operands and results

★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

Figure 1 of Power et al. "Grokking: Generalization beyond overfitting on small algorithmic datasets."

The Data

Split the table into
train & val datasets



★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

Figure 1 of Power et al. "Grokking: Generalization beyond overfitting on small algorithmic datasets."

Grokking and Generalization

★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

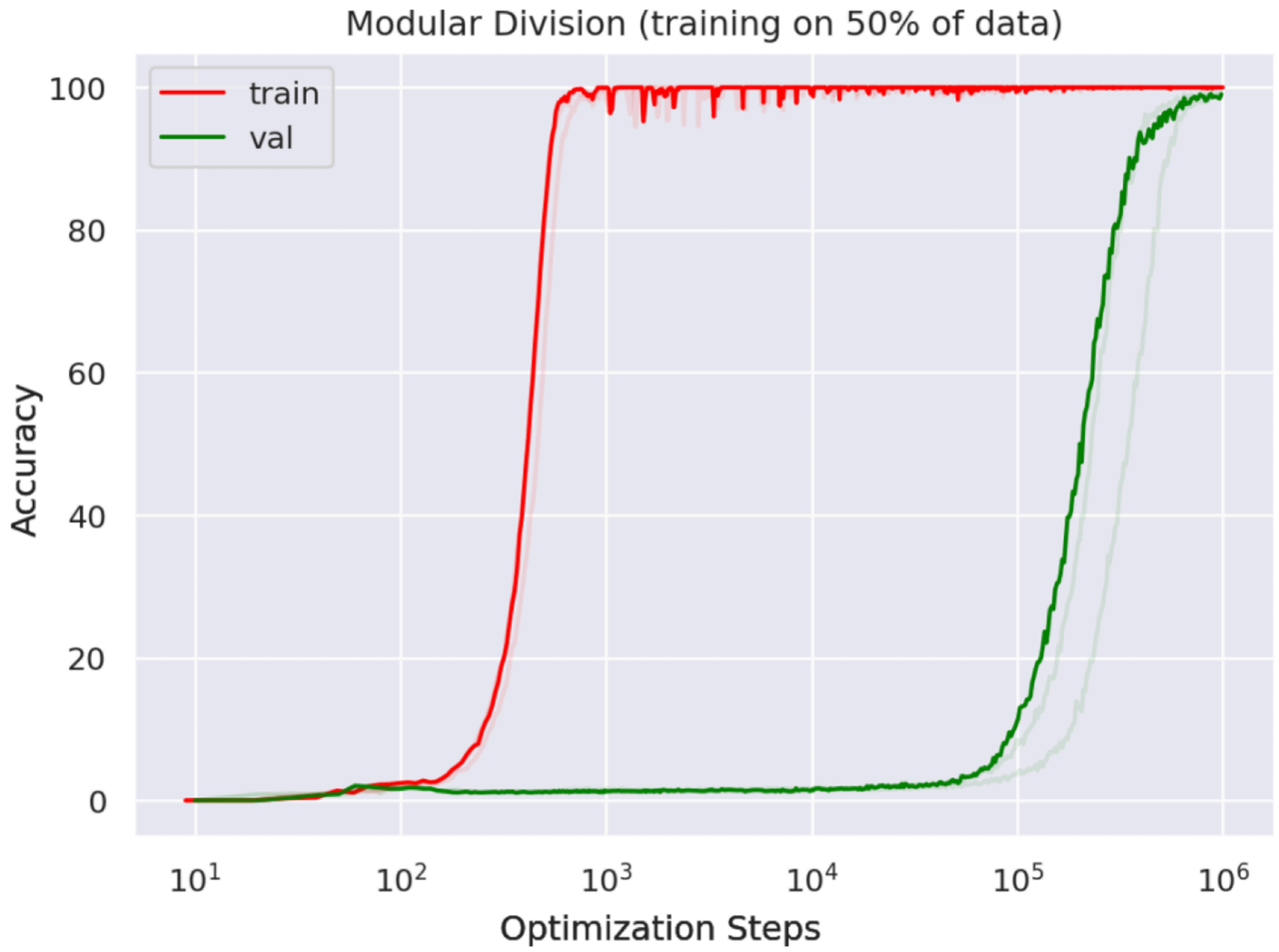
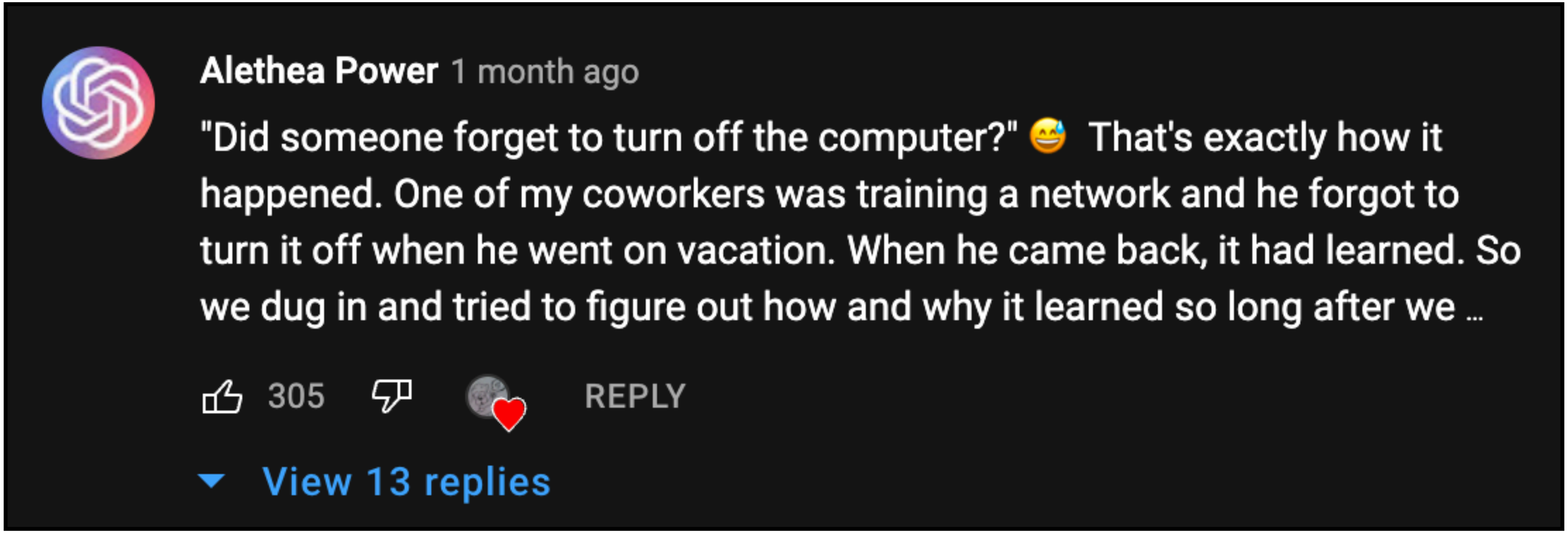


Figure 1 of Power et al. "Grokking: Generalization beyond overfitting on small algorithmic datasets."

How tf did they find that?



Training data dependence

★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

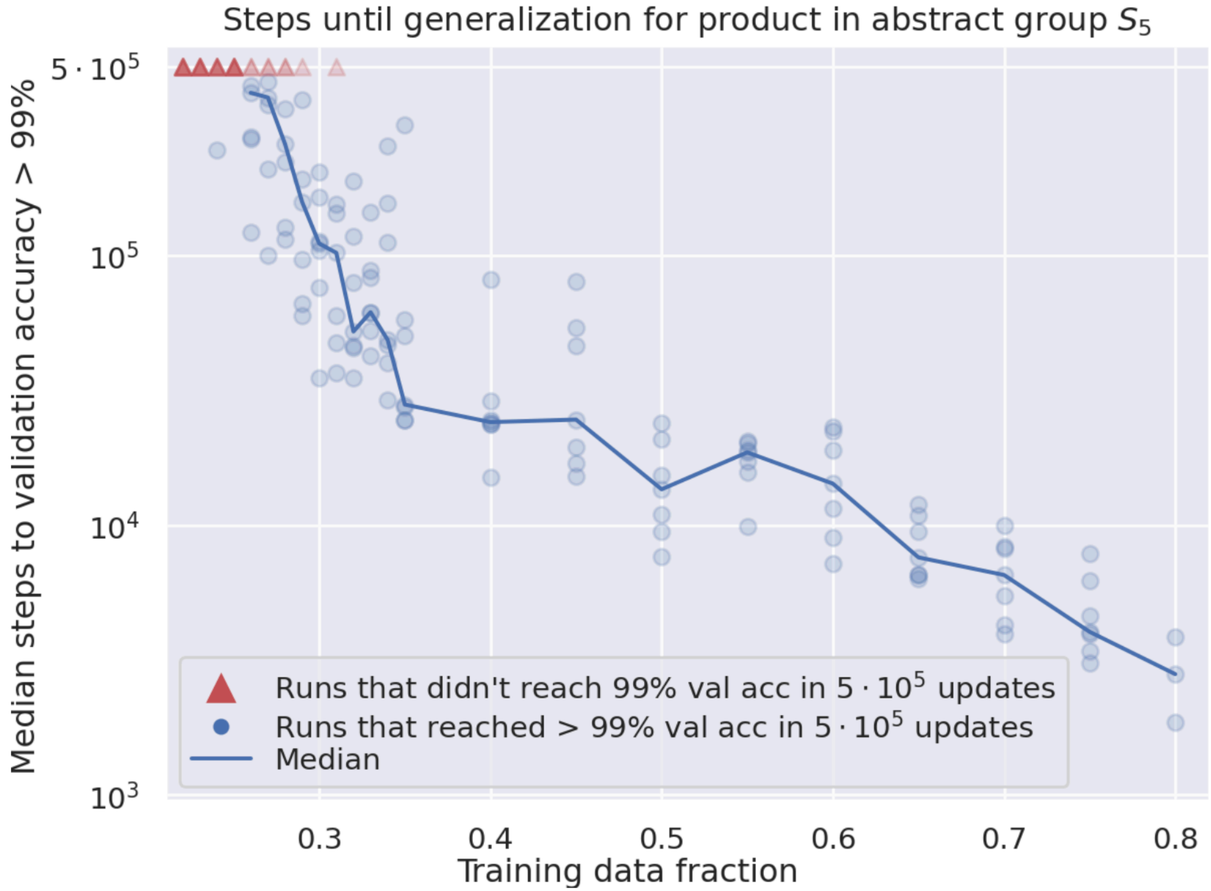


Figure 1 of Power et al. "Grokking: Generalization beyond overfitting on small algorithmic datasets."

Why is grokking interesting?

1. Because it is surprising!
2. Motivates investigation of 0-loss training & emergent capabilities
3. Do I just have to wait longer?

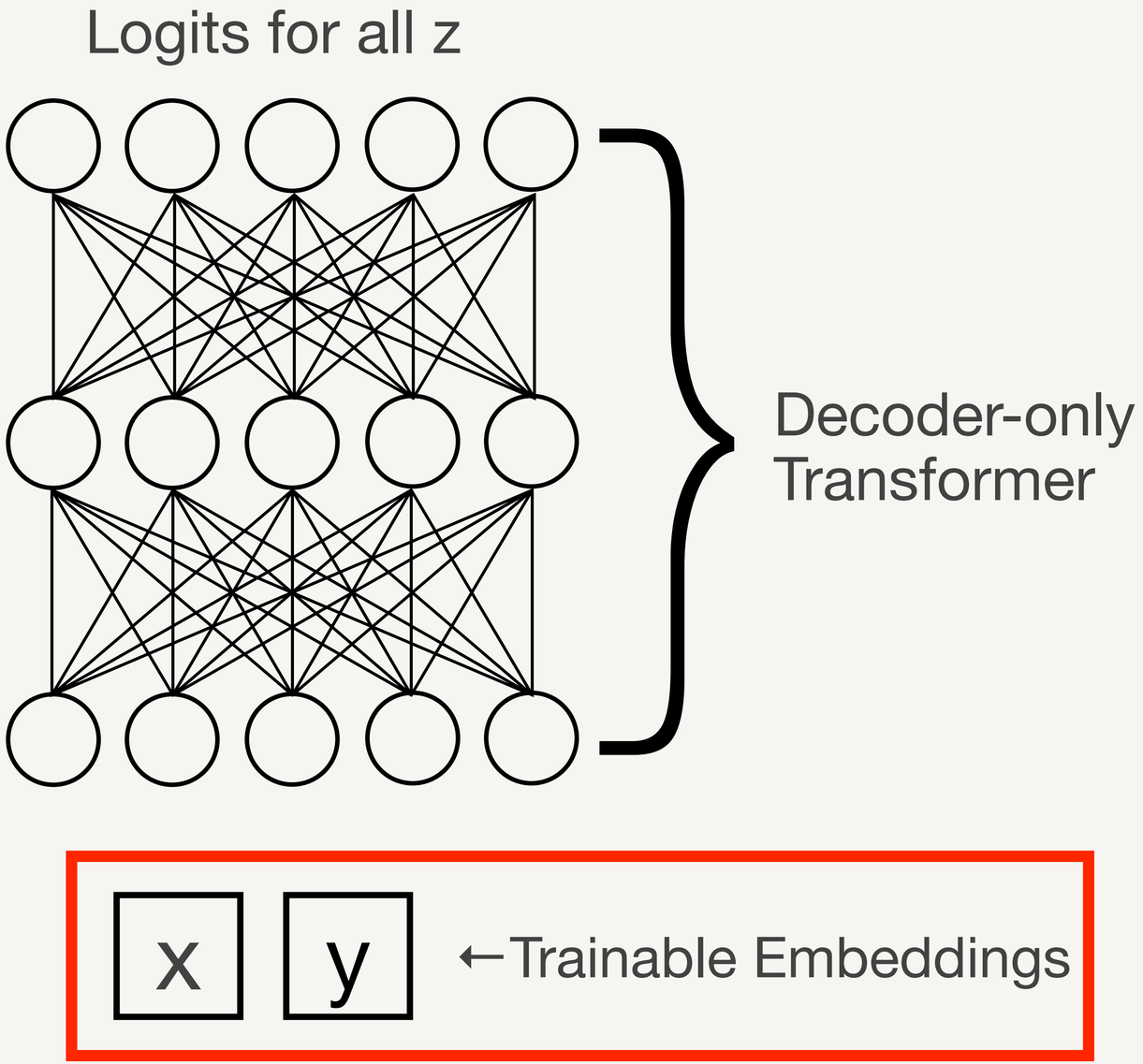
Questions

1. How do networks generalize on algorithmic datasets?
2. Why is the training set fraction so predictive of generalization time?
3. Why is generalization so delayed?

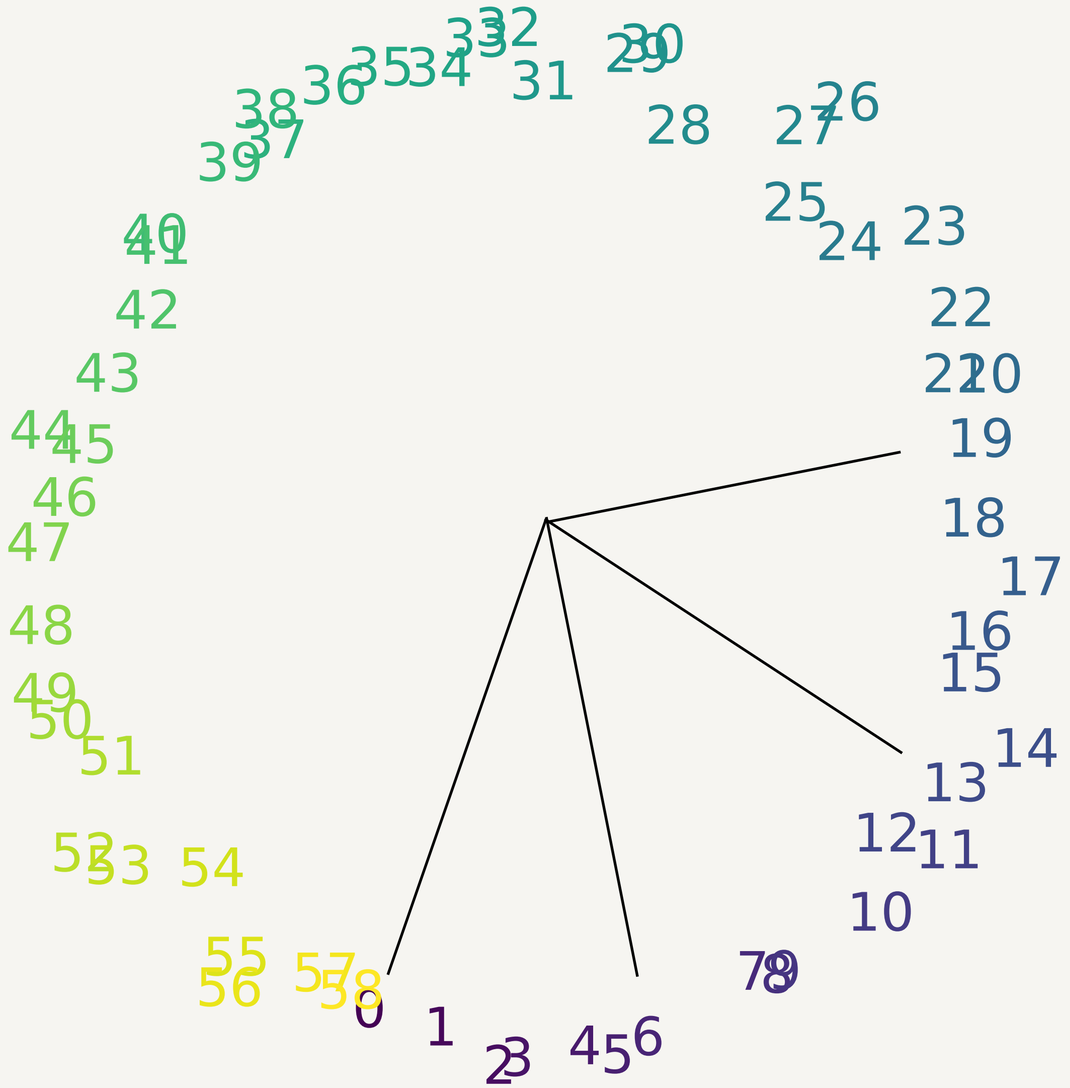
The Task

Learn a binary operation

$$x + y = z \mod 59$$

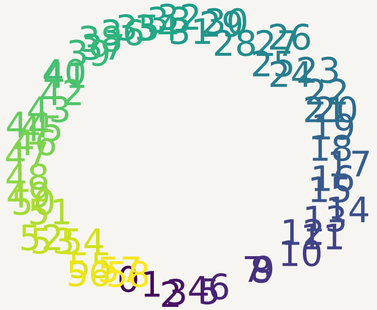


First 2 PCs of Trained Embeddings

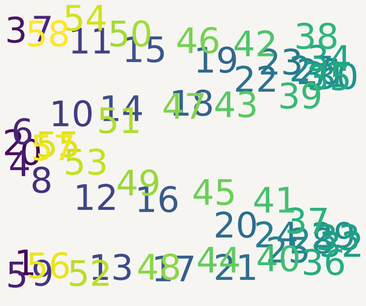


Different Principal Components

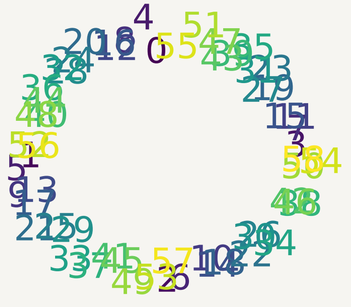
Axes 0 & 1



Axes 1 & 2

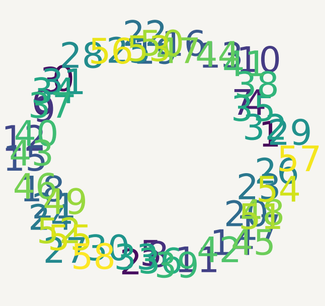


Axes 2 & 3

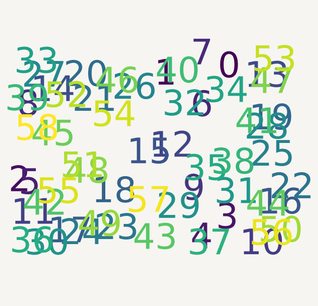


Axes 3 & 4

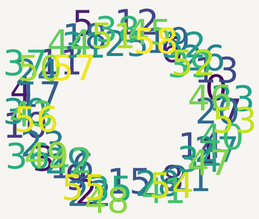
Axes 4 & 5



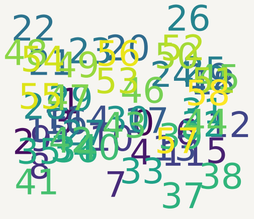
Axes 5 & 6



Axes 6 & 7



Axes 7 & 8

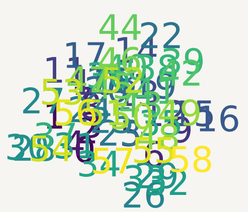


Axes 8 & 9

Axes 9 & 10

Axes 10 & 11

Axes 11 & 12

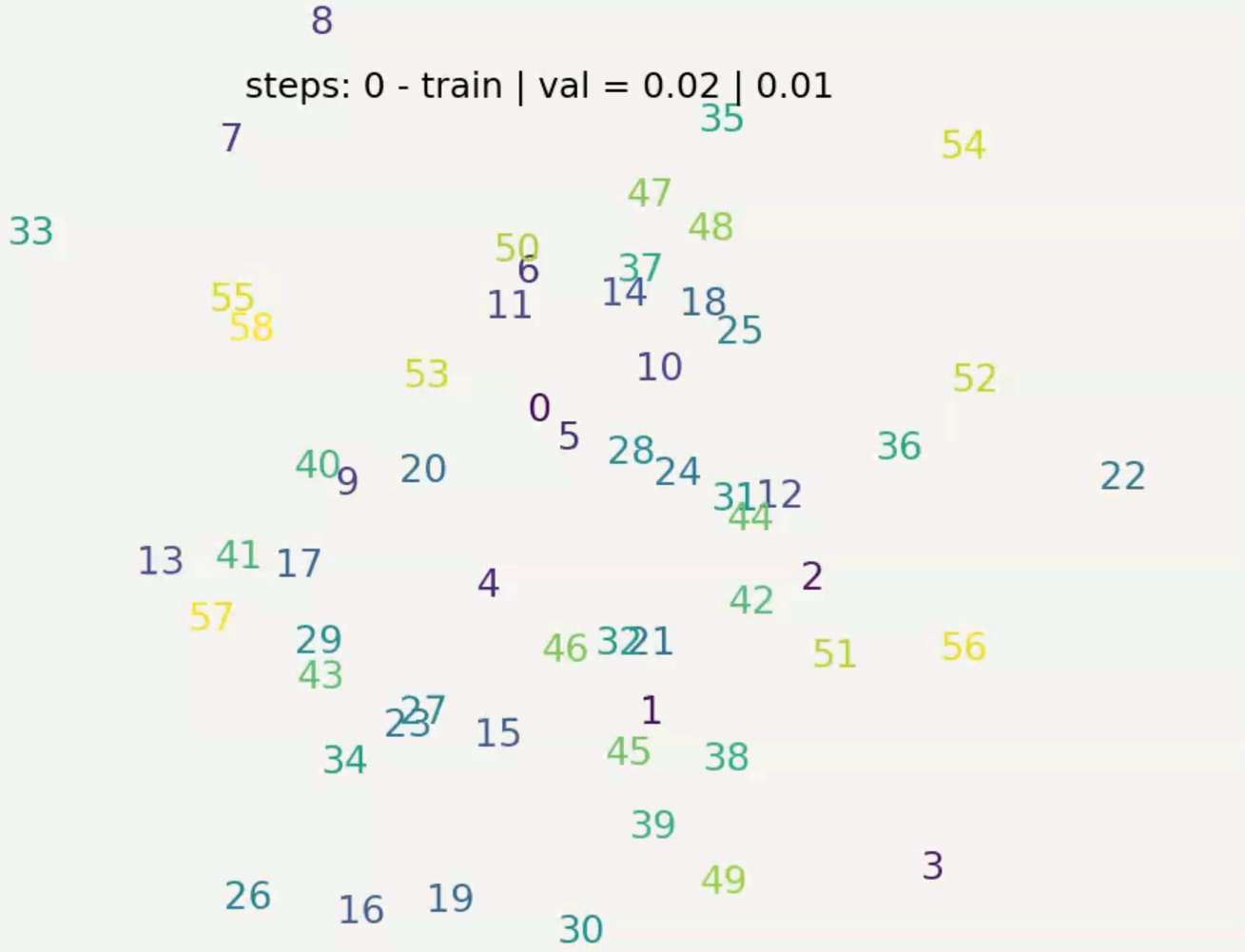


Axes 12 & 13

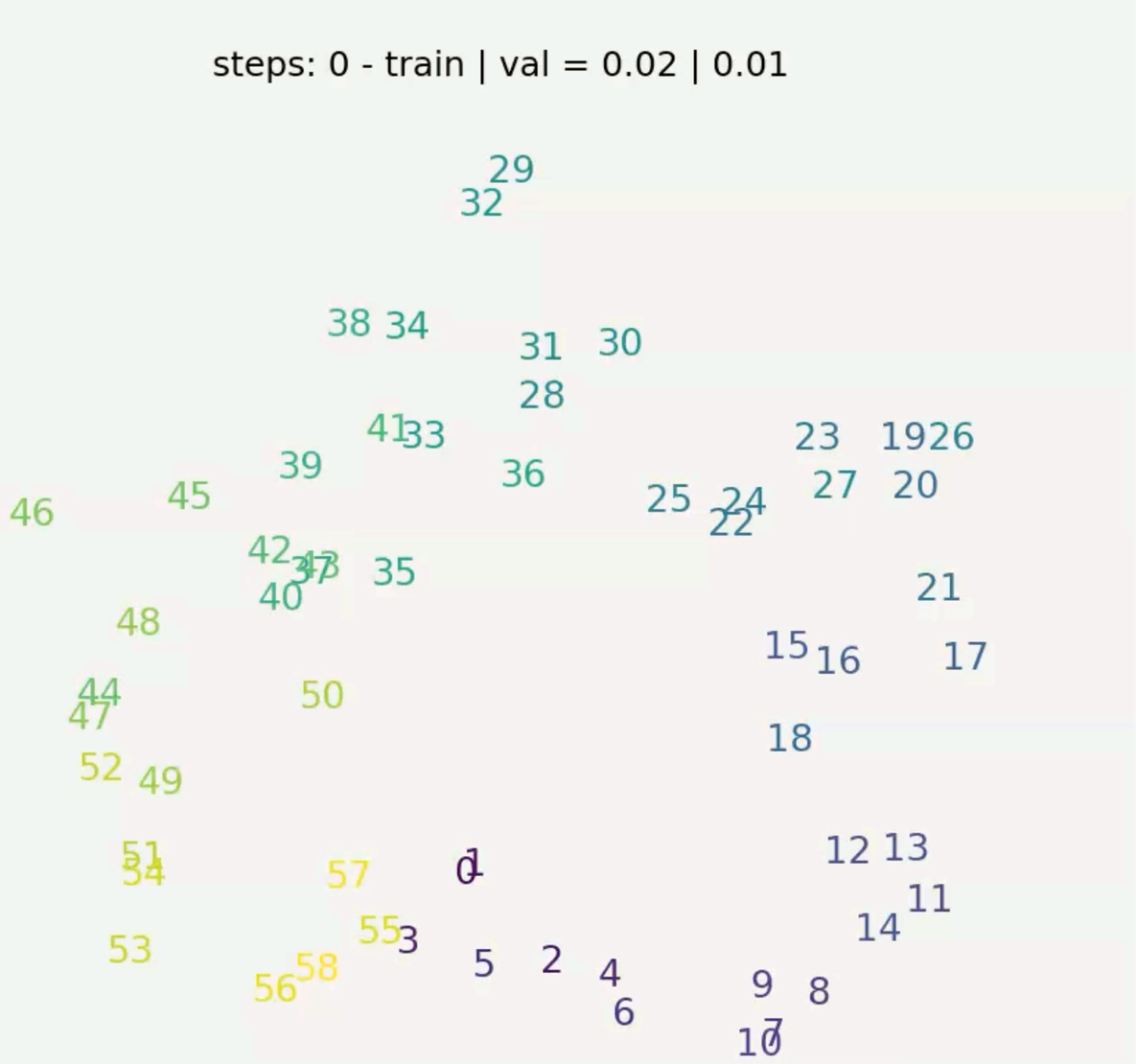
Axes 13 & 14

Axes 14 & 15

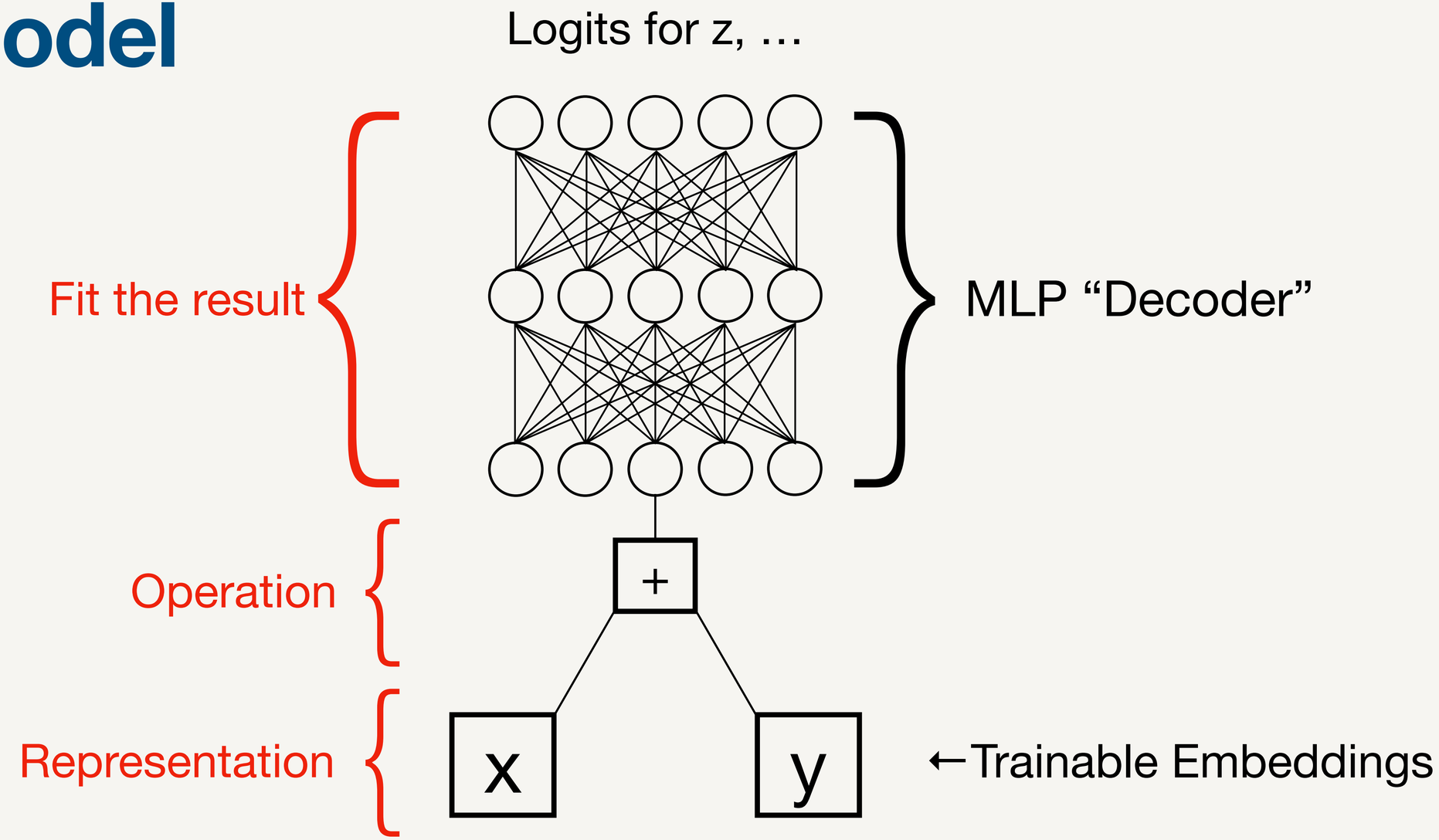
First 2 PCs @ current step during training



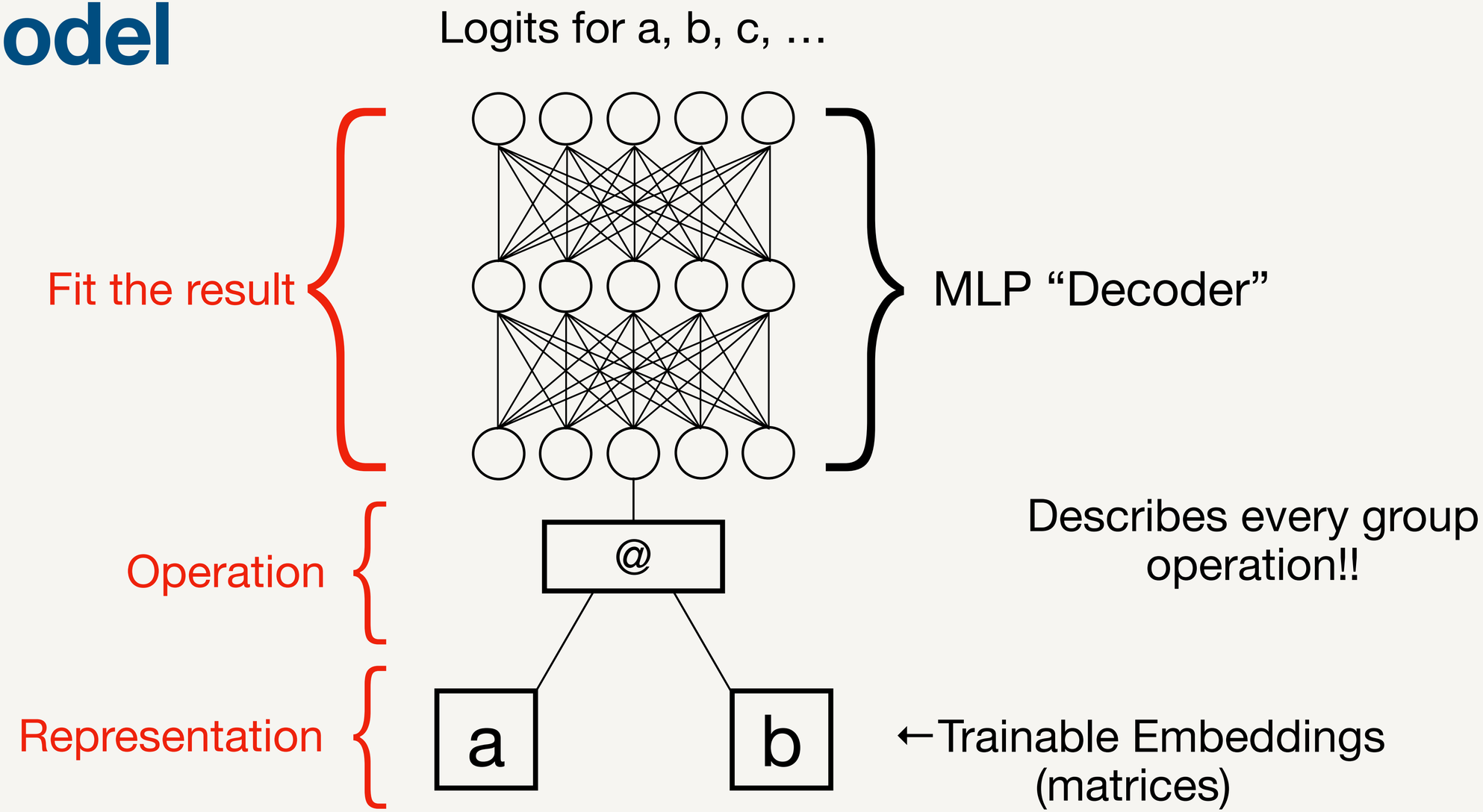
First 2 PCs @ last step during training



Toy Model


Logits for z, …
Fit the result
MLP “Decoder”
Operation
+
Representation
x
y
←Trainable Embeddings

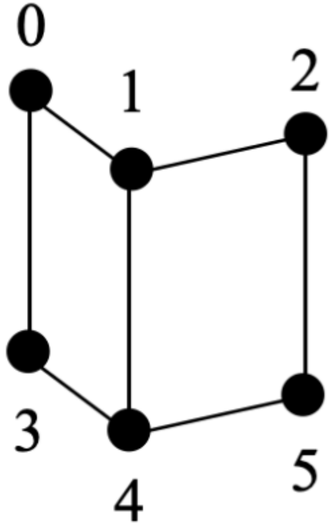
Toy Model

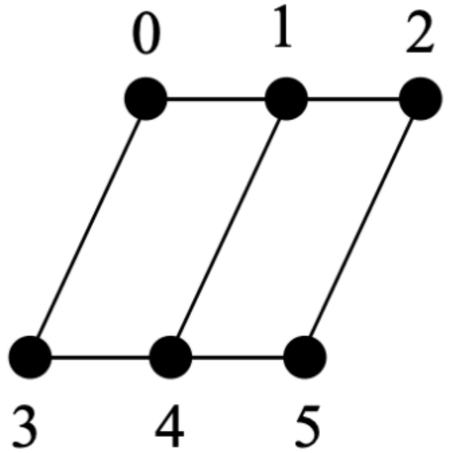

Logits for a, b, c, …
Fit the result
MLP “Decoder”
Operation
@
Describes every group operation!!
Representation
a
b
←Trainable Embeddings (matrices)

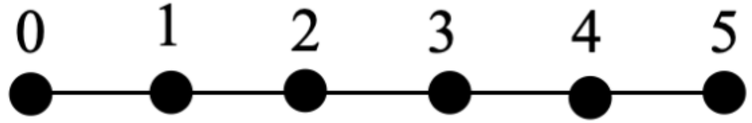
How can we generalize addition?

$$(i, j) \rightarrow i + j$$
$$(i + n, j - n) \rightarrow i + j$$

What is the best representation to achieve that?







An Effective Loss

$$P_0(D) = \{(i,j,m,n) \,|\, (i,j) \in D, (m,n) \in D, i + j = m + n\}$$

$$\ell_{\text{eff}} = \frac{\ell_0}{Z_0}, \quad \ell_0 \equiv \sum_{(i,j,m,n) \in P_0(D)} |\mathbf{E}_i + \mathbf{E}_j - \mathbf{E}_m - \mathbf{E}_n|^2, \quad Z_0 \equiv \sum_k |\mathbf{E}_k|^2,$$

$$\frac{dE_i}{dt} = -\eta \frac{d\ell_{\text{eff}}}{dE_i}$$

Time to linear structure

$$\ell_{\text{eff}} = \frac{\ell_0}{Z_0}, \quad \ell_0 \equiv \sum_{(i,j,m,n)\in P_0(D)} |\mathbf{E}_i + \mathbf{E}_j - \mathbf{E}_m - \mathbf{E}_n|^2, \quad Z_0 \equiv \sum_k |\mathbf{E}_k|^2,$$

$$\ell_{\text{eff}} = \frac{1}{2} R^T H R, R = [E_0, E_1, \cdots, E_{p-1}]$$

$$\frac{dR}{dt} = -HR \qquad H_{ij} = \frac{1}{Z_0} \frac{\partial^2 \ell_0}{\partial E_i \partial E_j}$$

Time to linear structure

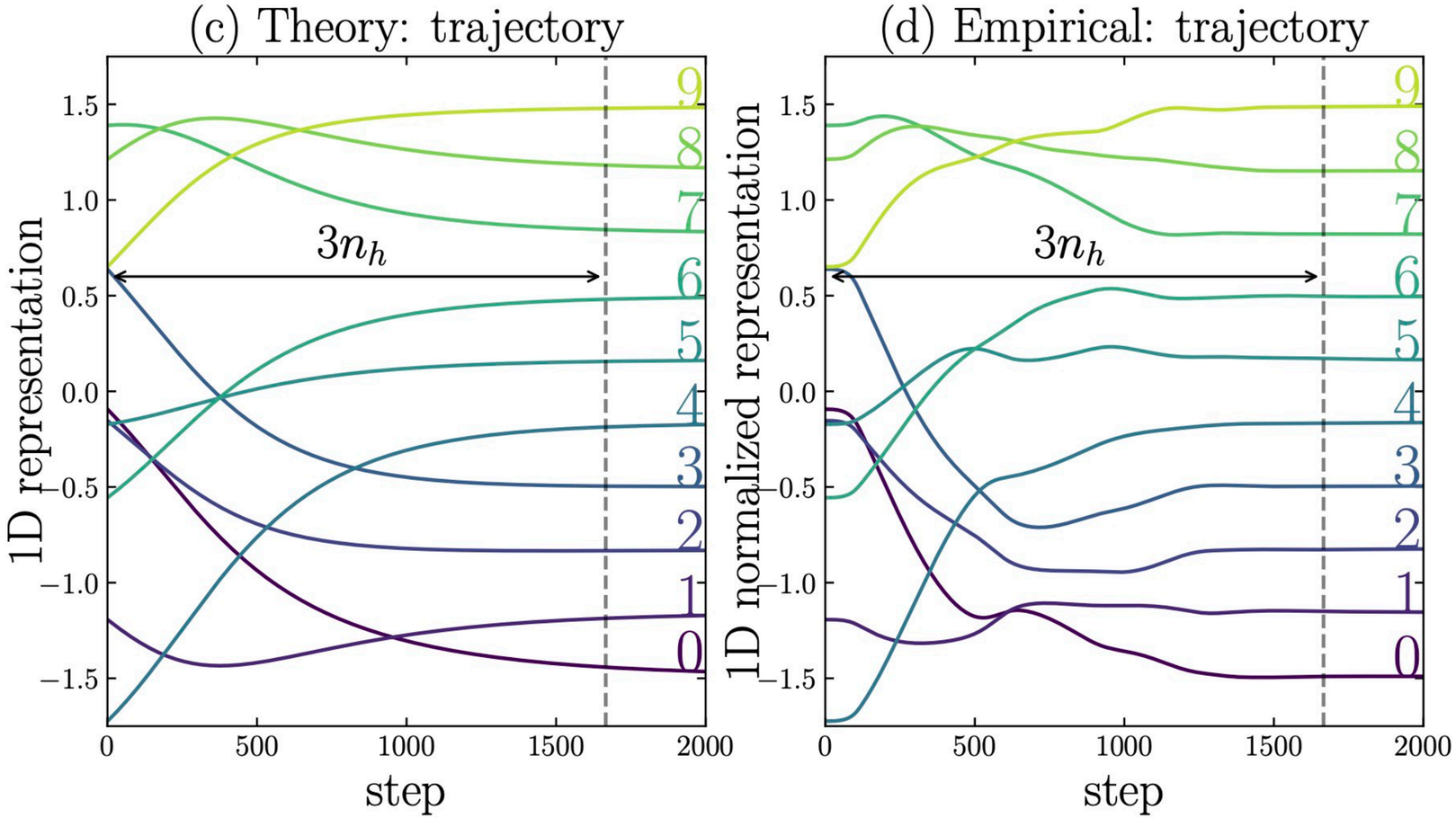
$$\frac{dR}{dt} = -HR$$

$$R(t=0) = \sum_i a_i \vec{v}_i$$

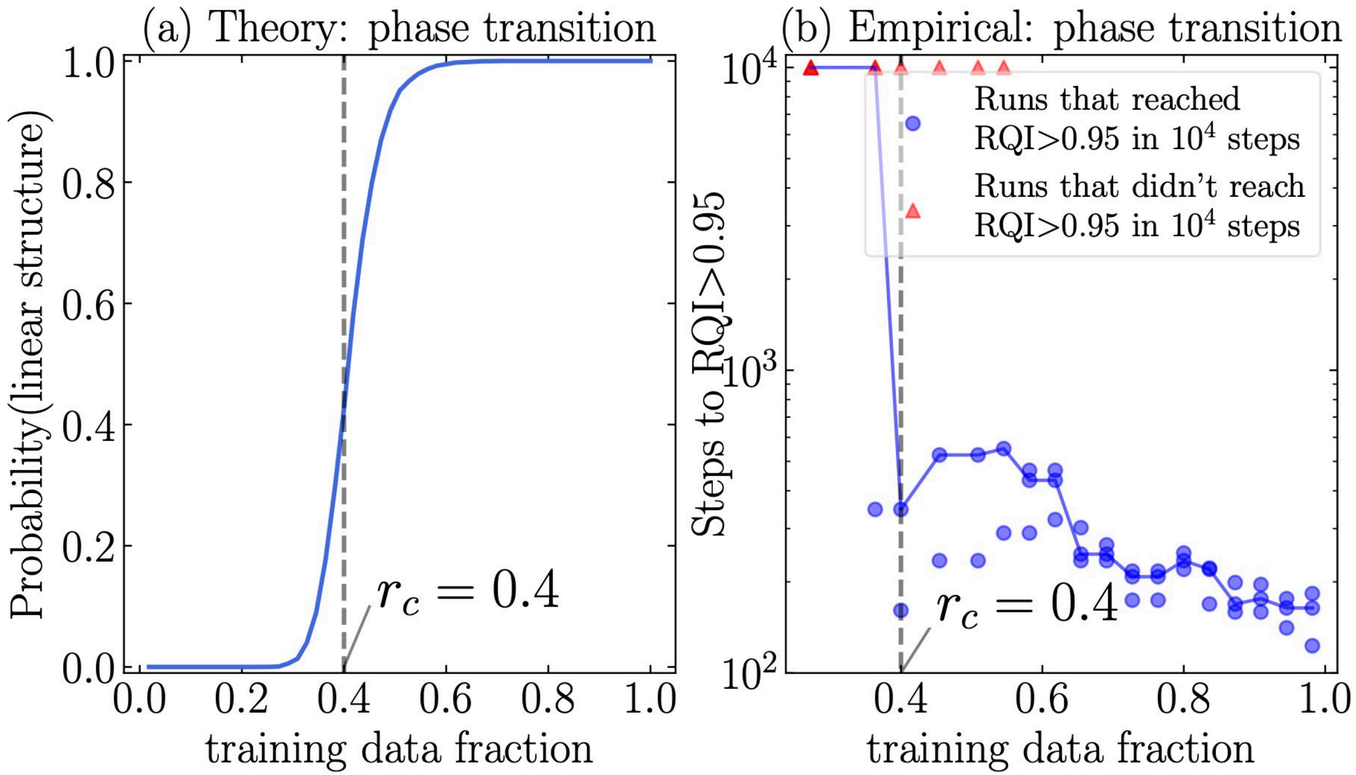
$$H = \text{diag}(\lambda_i)$$

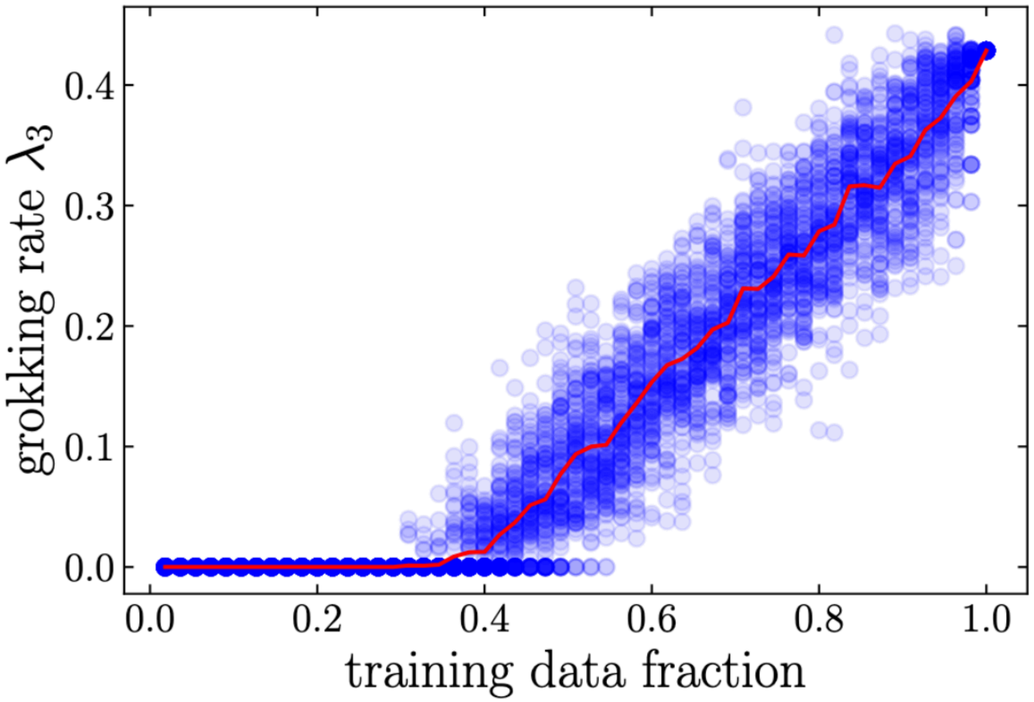
$$\rightarrow R(t) = \sum_i a_i \vec{v}_i e^{-\lambda_i t}$$

Predictor of Training dynamics



Predictor of Generalization (Time)





Answers

1. How do networks generalize on algorithmic datasets?

   → Via structured representations

2. Why is the training set fraction so predictive of generalization time?

   → Critical minimum + faster parallelograms

3. Why is generalization so delayed?

   → Close to the critical training data fraction

Final thoughts

Crucial dependence on Representation

→ Less explored territory of learning

How can we speed up structured representations?