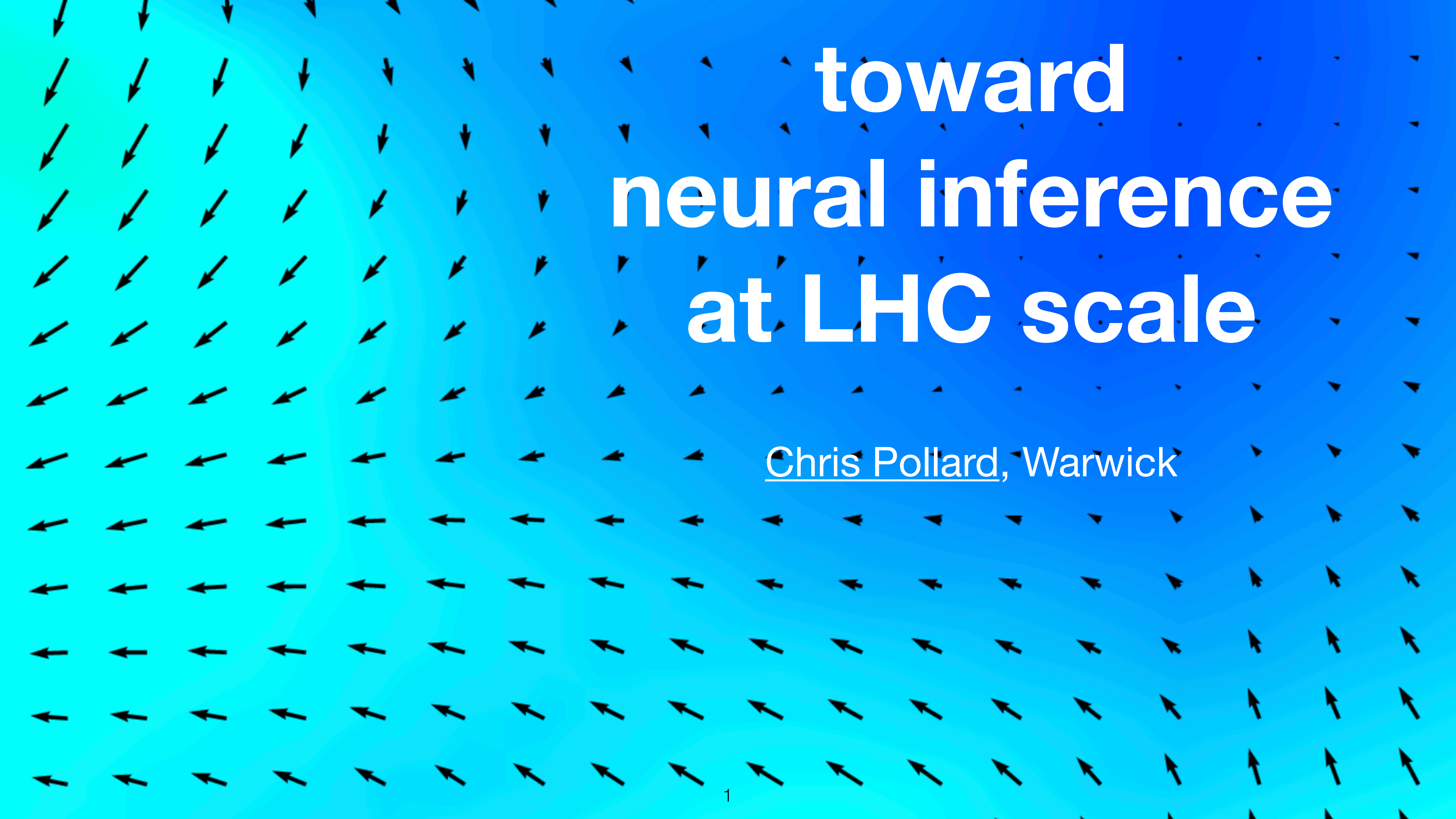


The background of the slide features a dense grid of small black arrows. These arrows are oriented in various directions, creating a complex, swirling pattern. The background color is a gradient of blue, transitioning from a lighter cyan on the left to a deeper blue on the right.

toward neural inference at LHC scale

Chris Pollard, Warwick







how does your brain efficiently break down an image?
it performs a quick “compression” of relevant features,
and it has seen many varieties of situations beforehand.



how does your brain efficiently break down an image?

it performs a quick “compression” of relevant features,

and it has seen many varieties of situations beforehand.

maintaining all information relevant to a task:

sufficiency.

expending effort ahead of time to enable quick inference:

amortization.

how does your brain efficiently break down language?

it performs a quick “compression”

and it has seen many variations

maintaining

experience



This is for informational purposes only. For medical advice or diagnosis, consult a professional.

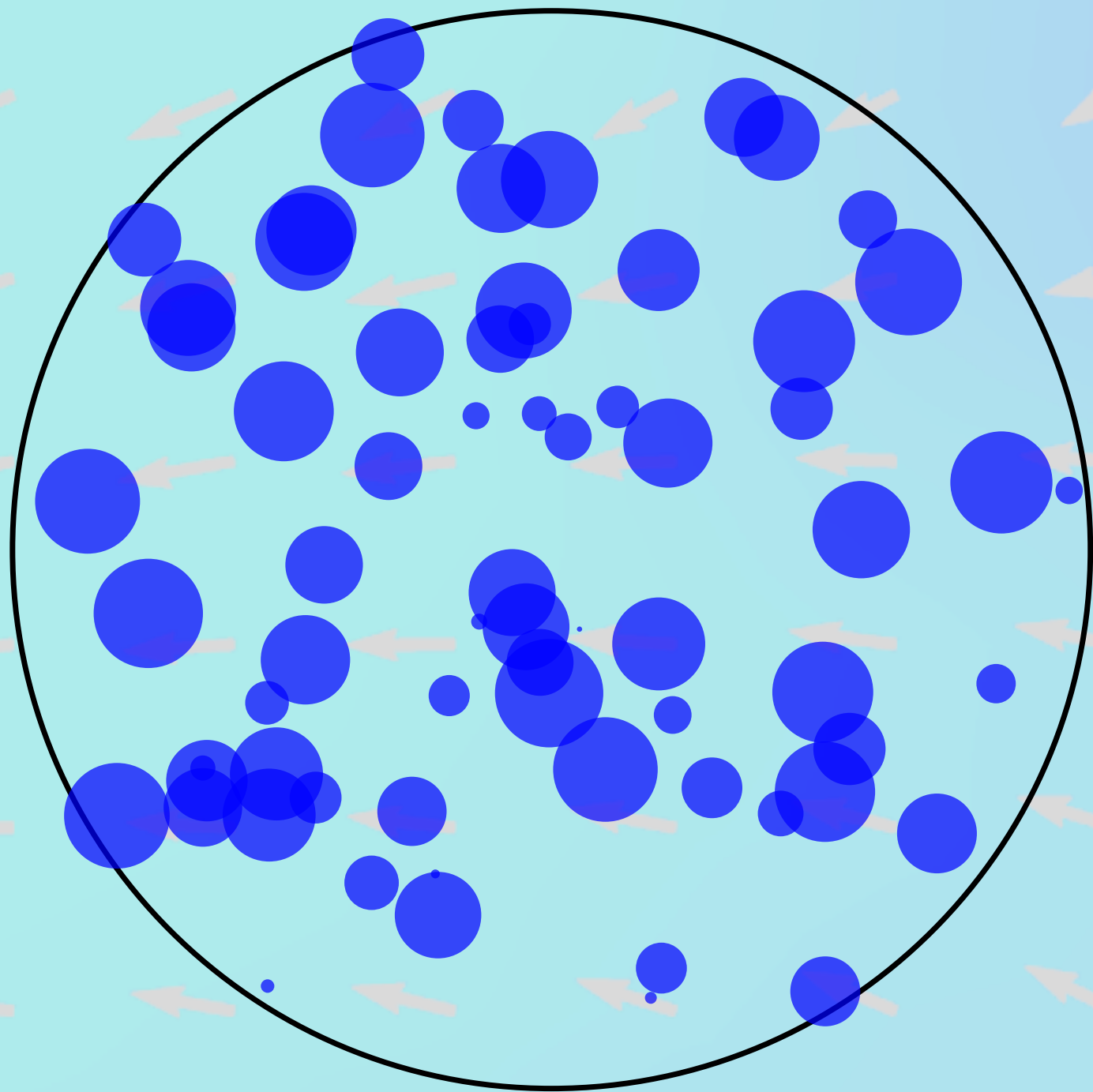
relevant to a task:

efficiency.

and of time to enable quick inference:
amortization.

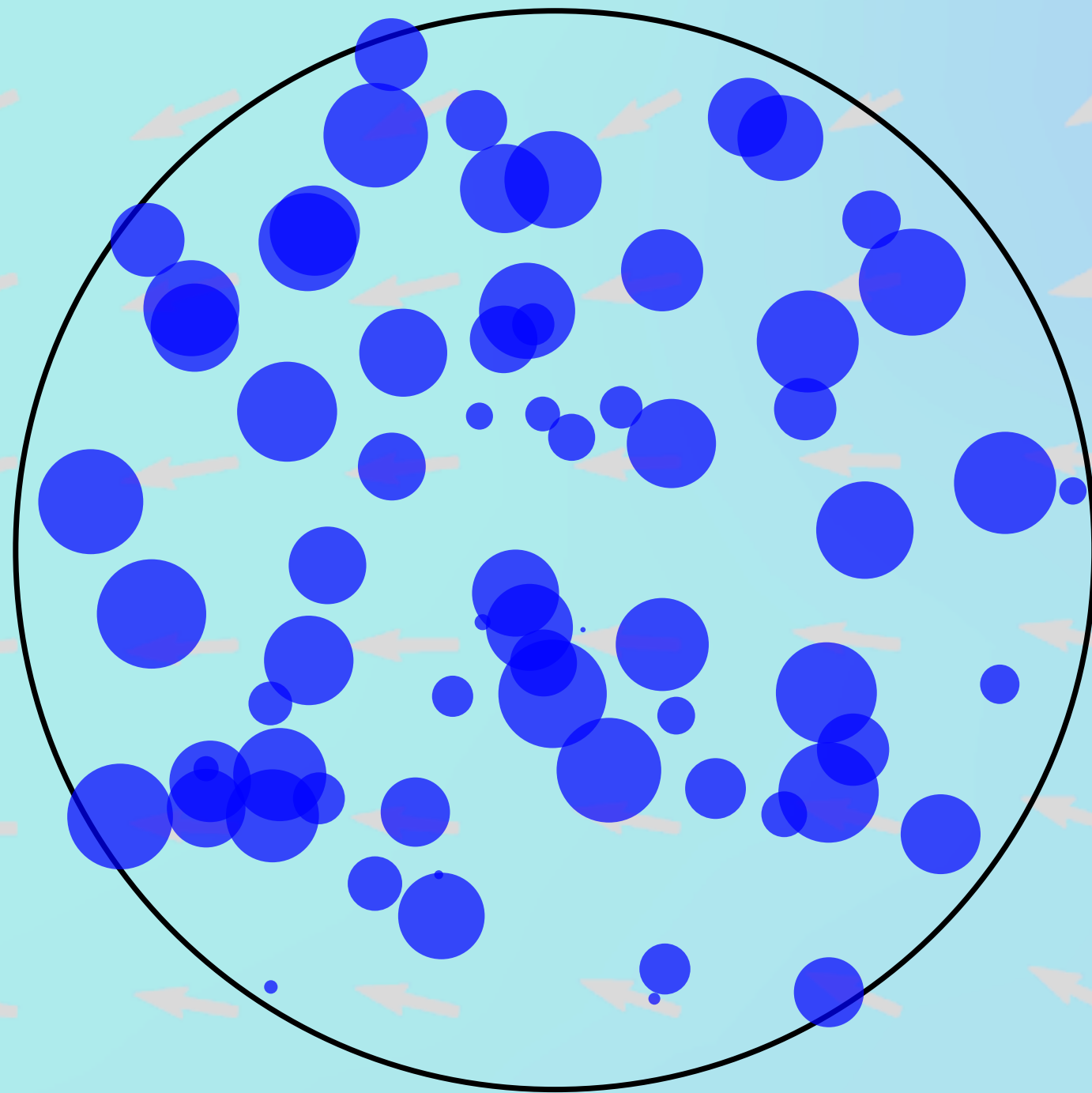
in collider physics we often deal with a simpler situation:

**e.g. jet constituents are *exchangeable*,
greatly reducing the function space over which
flavor taggers must search.**



in collider physics we often deal with a simpler situation:

**e.g. jet constituents are *exchangeable*,
greatly reducing the function space over which
flavor taggers must search.**



**set-transformers provide a good
inductive bias for jet-tagging tasks.**

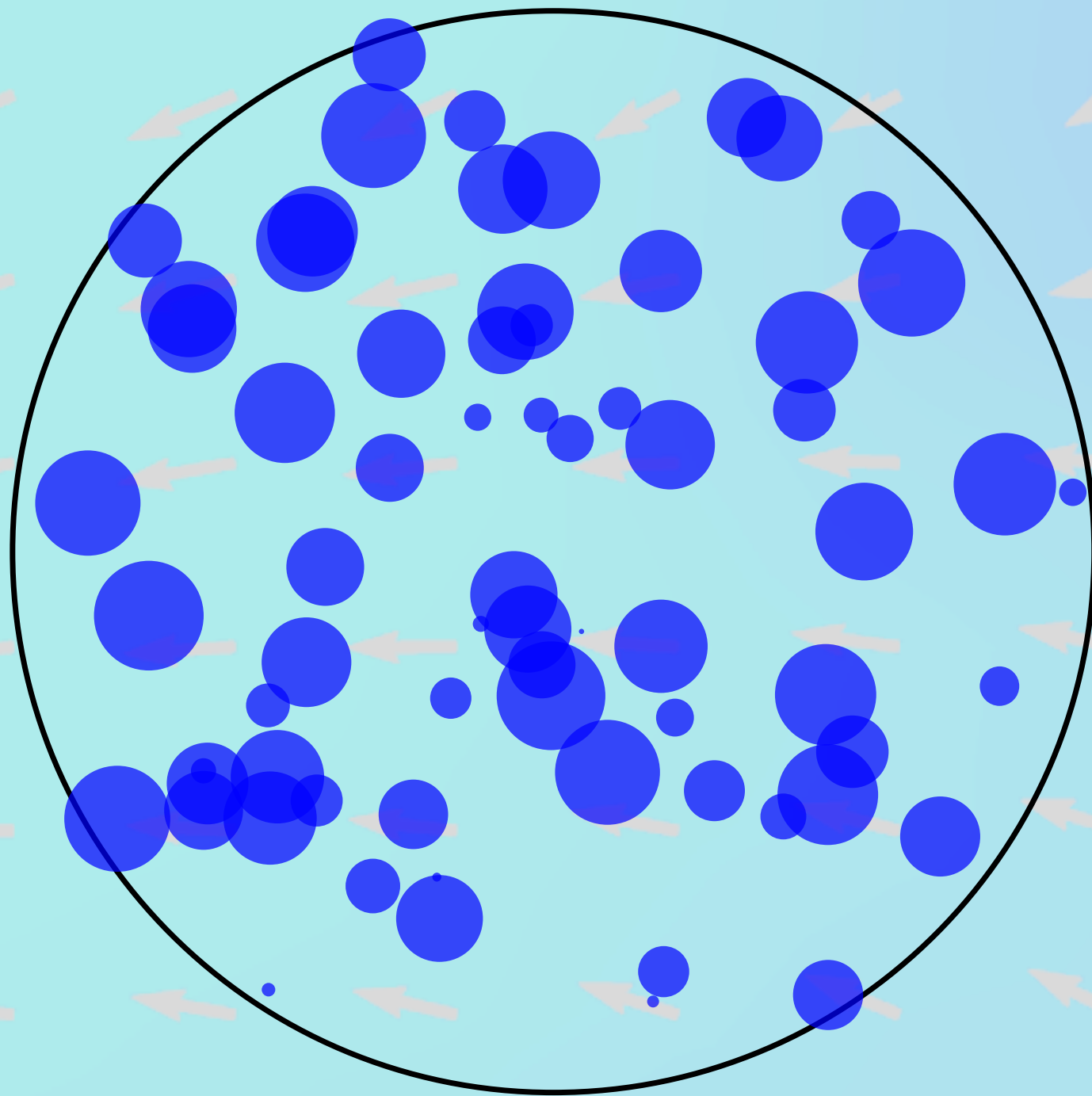
**large volumes of simulation allow
quick inference via amortization.**

when looking at entire datasets,

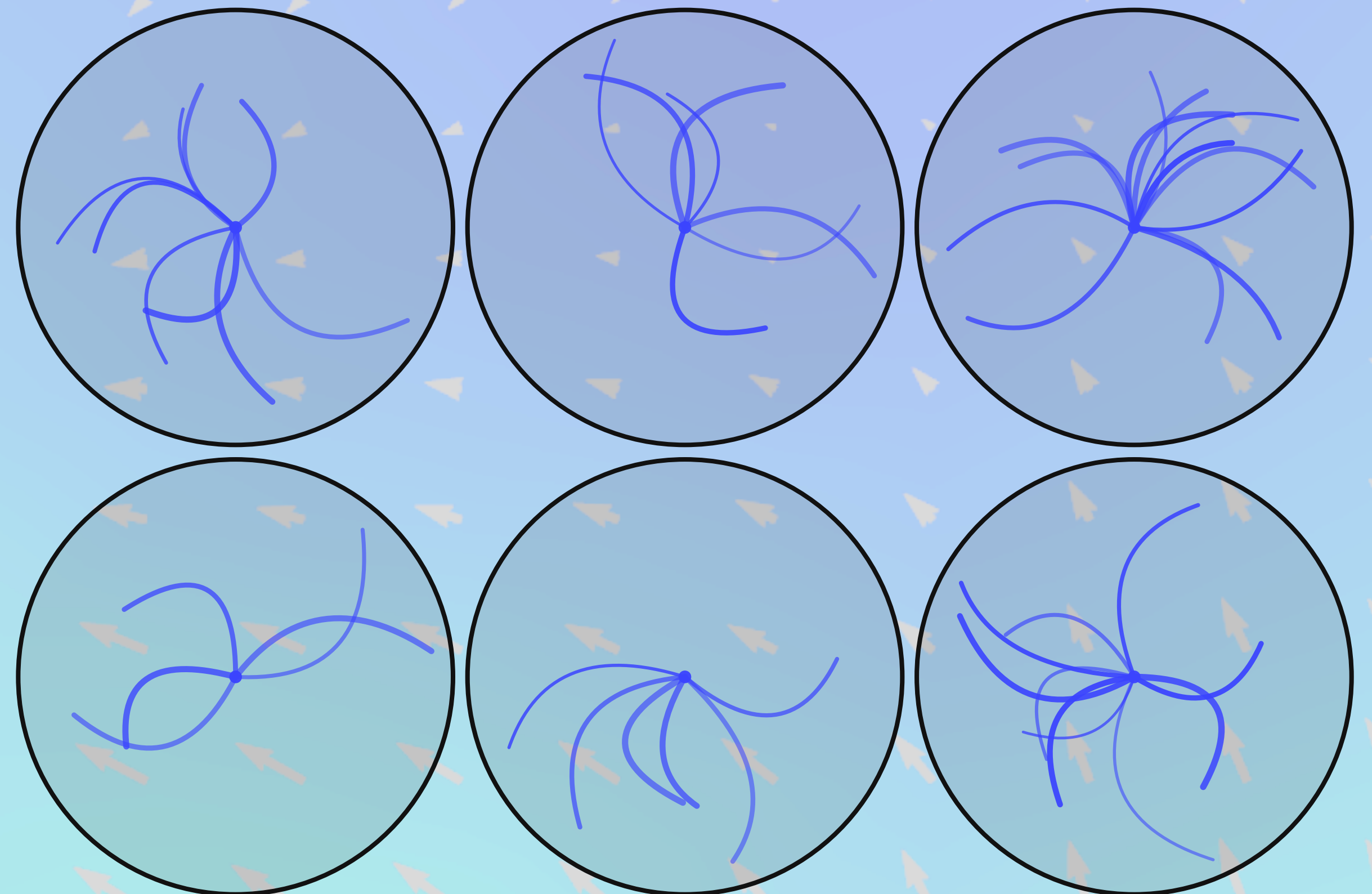
observations are not only *exchangeable*, they are (nearly)
independent and identically distributed (IID).

when looking at entire datasets,
observations are not only *exchangeable*, they are (nearly)
independent and identically distributed (IID).

exchangeable

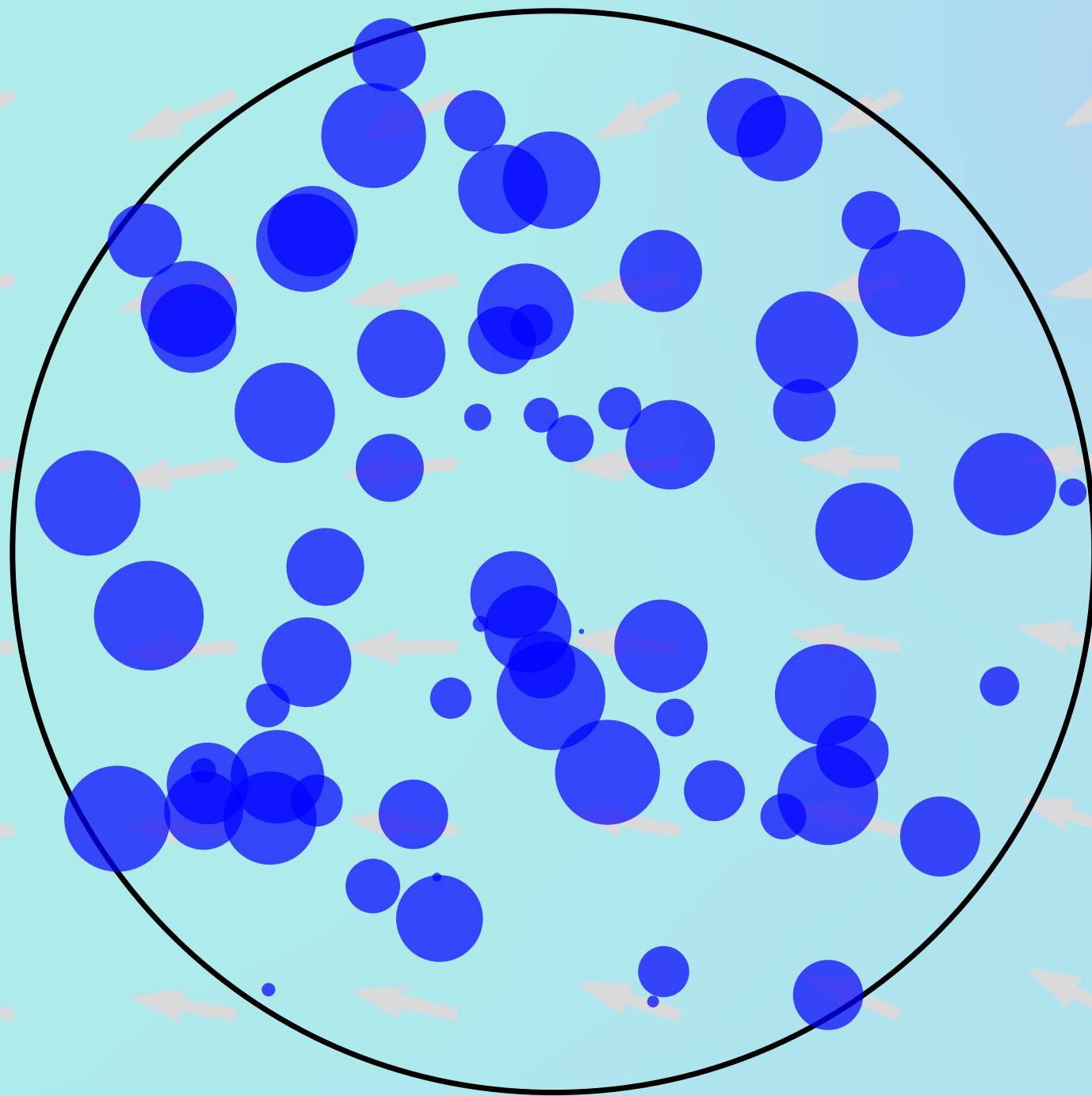


IID

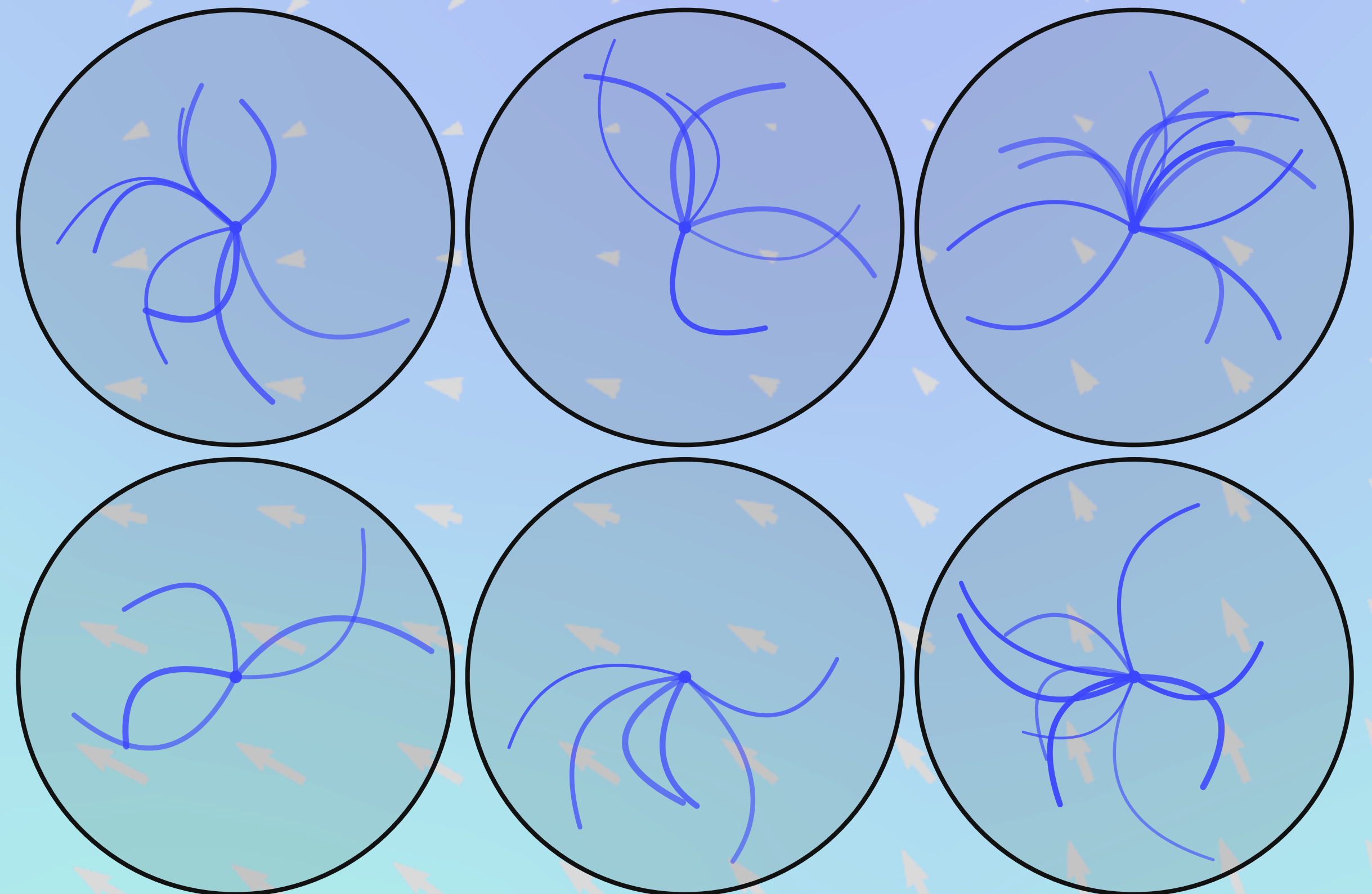


when looking at entire datasets,
observations are not only *exchangeable*, they are (nearly)
independent and identically distributed (IID).

exchangeable



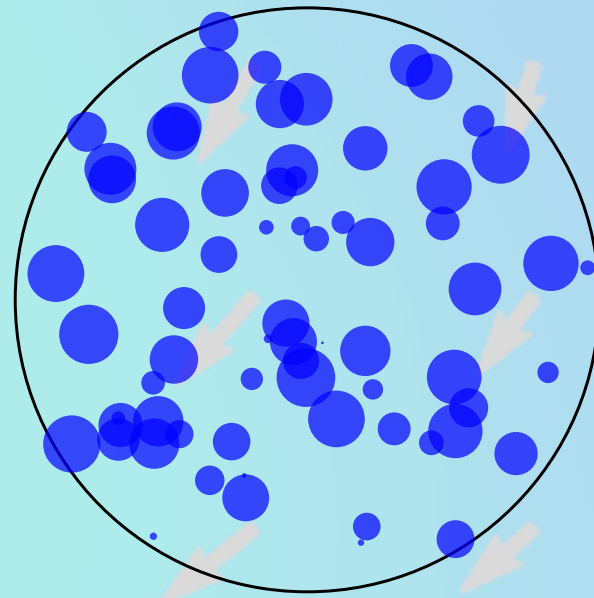
IID



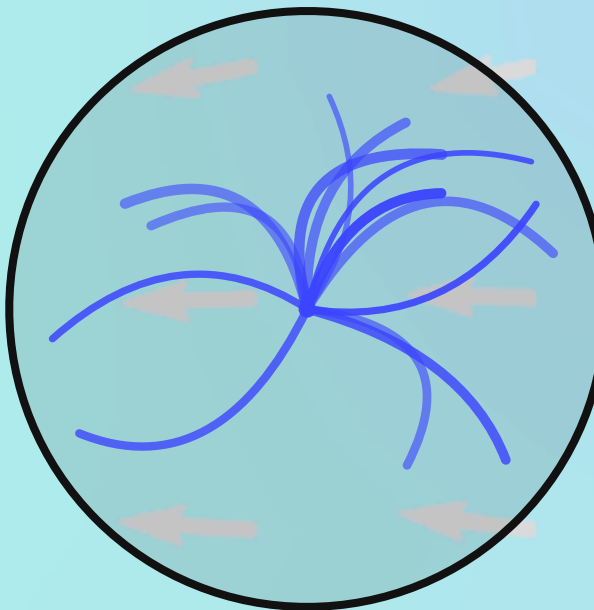
two axes: event complexity and dataset size

complexity of $x \uparrow$

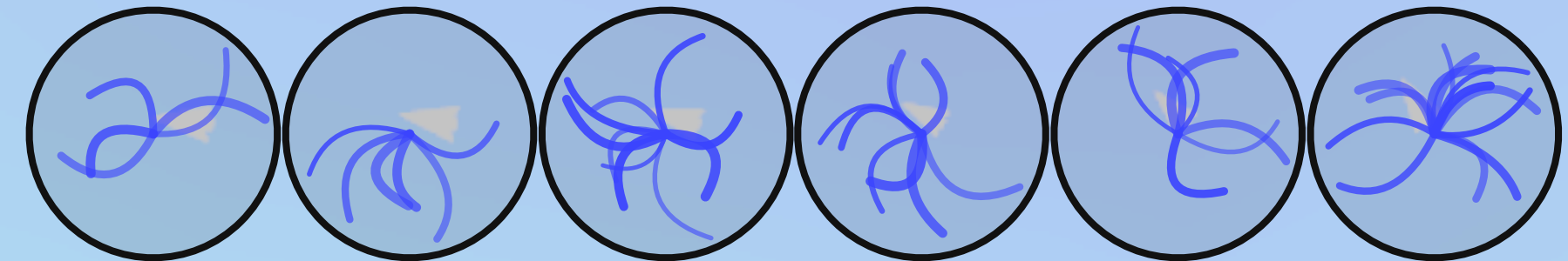
“raw”
data



coarse
summary



more informative



single
event

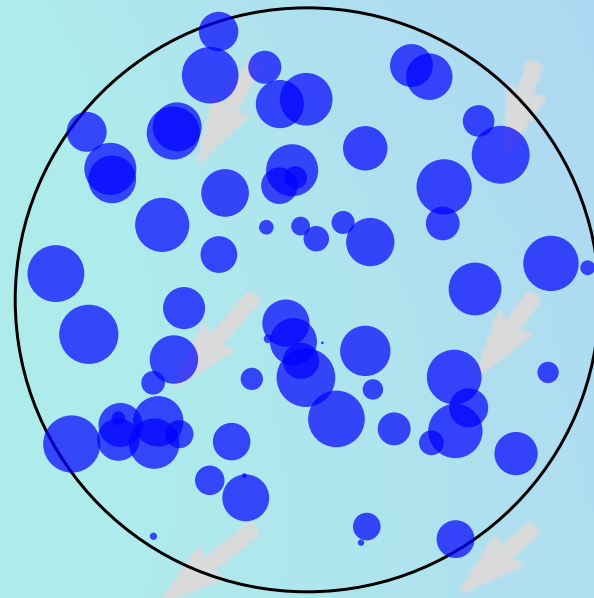
LHC
dataset

dataset size $N \rightarrow$

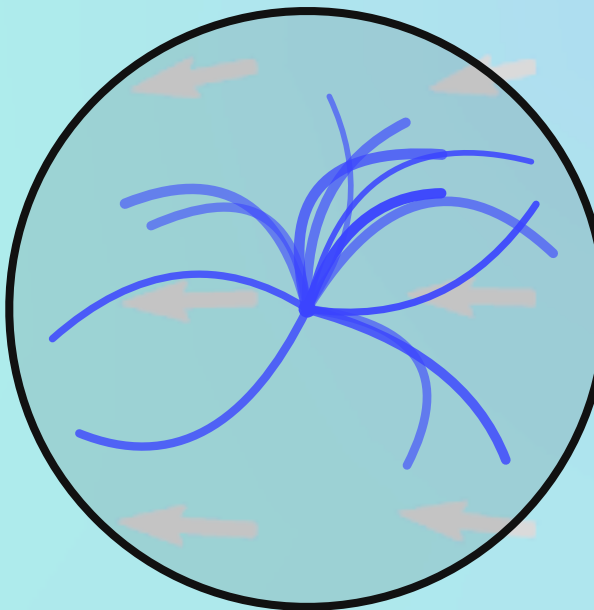
two axes: event complexity and dataset size

complexity of $x \uparrow$

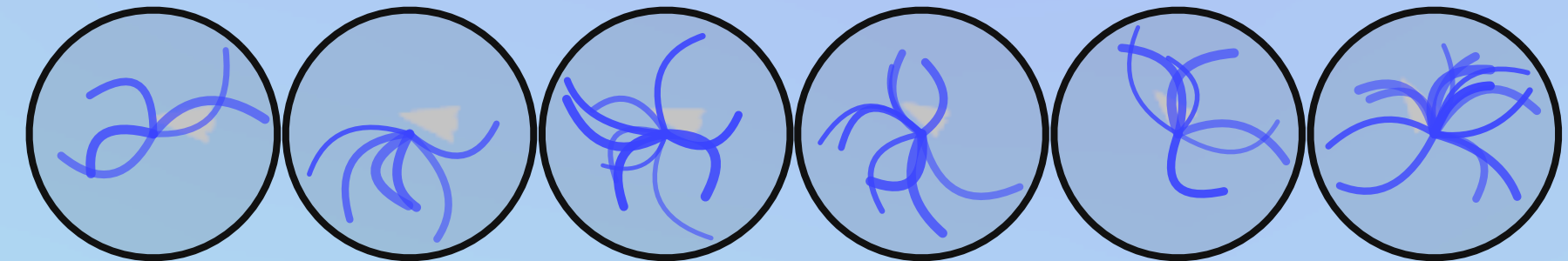
**“raw”
data**



**coarse
summary**



more informative



**single
event**

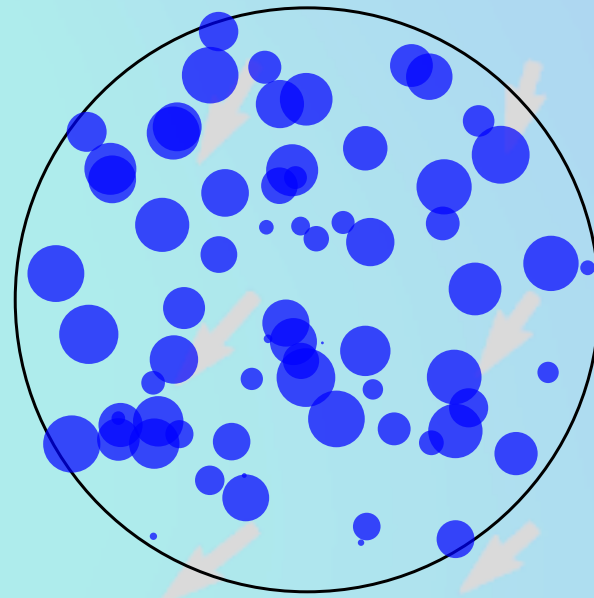
**LHC
dataset**

dataset size $N \rightarrow$

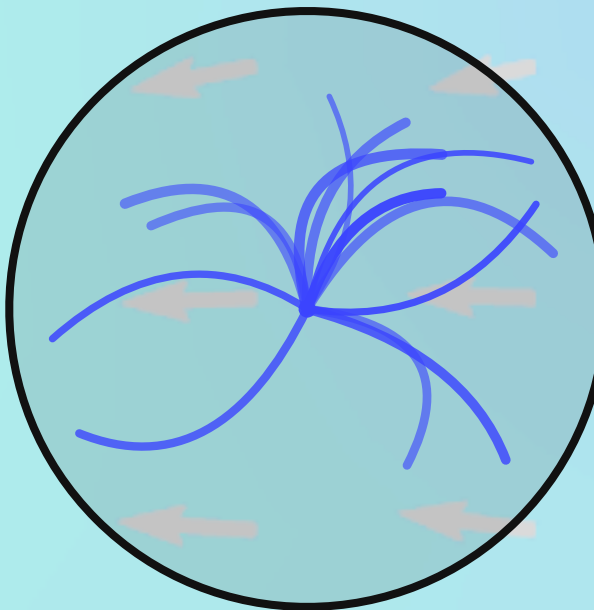
two axes: event complexity and dataset size

complexity of $x \uparrow$

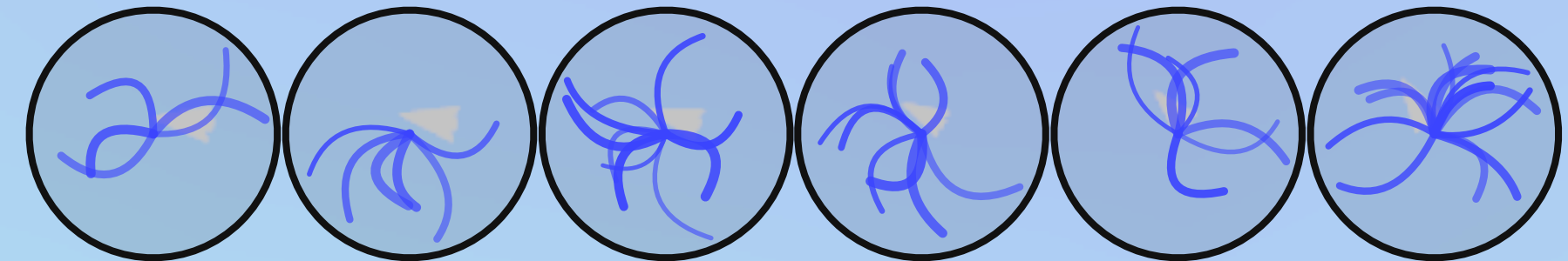
“raw”
data



coarse
summary



more expensive



single
event

LHC
dataset

dataset size $N \rightarrow$

to state the problem more formally,

- we aim for a procedure for inferring some **parameter ϕ** ,
given some observations $\{x\} \equiv \{x_1, x_2, \dots\}$,
even in the **presence of nuisance parameters**.
- for example, **given the data we've observed**, what are the constraints on
 - ◆ $\phi \equiv \alpha_s(Q^2)$ or $\phi \equiv \mu_{\text{new physics}}$?
- this procedure should **computationally scale gracefully to LHC-scale datasets**.

this procedure should **computationally scale gracefully to large datasets.**

◎ two main requirements:

◆ efficient “compression” of observations x , $t_i \equiv t(x_i)$

◆ efficient way to combine compressed observations t_i, t_j

◎ ***both*** above should preserve information about ϕ from x .

recent literature solves this problem under lenient conditions
if we are able to simulate $\phi \rightarrow p(\{x\})$.

- i.e. we need to be able to produce datasets with
various choices of the quantity we want to extract.

Published in Transactions on Machine Learning Research (02/2024)

Hierarchical Neural Simulation-Based Inference Over Event Ensembles

[publication link](#)

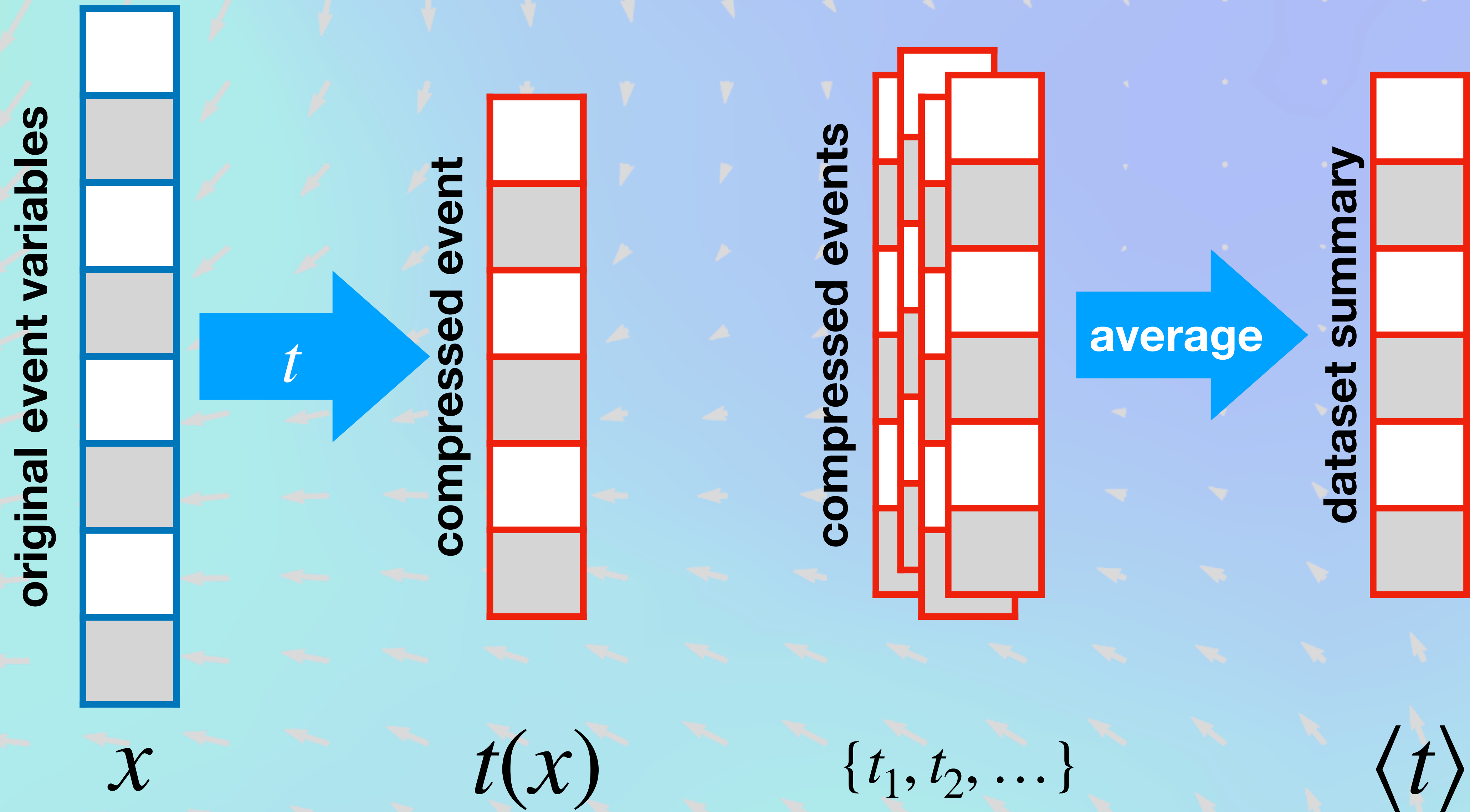
TMLR (2024)

- e.g. to measure $\alpha_S(Q^2)$, one needs to be able to simulate datasets given each choice of α_S .

How? Train two neural networks jointly on your simulation:

- one learns a **per-event compression** $t(x)$;
- the second learns $\rho(\langle t \rangle) \approx p(\phi \mid \langle t \rangle)$,
i.e. $p(\phi)$ given the **average** $t(x)$ over observations:
 - ◆ averaging is the “correct” inductive bias for IID data.
- this procedure asymptotically approaches the correct $\rho(\langle t \rangle) \approx p(\phi \mid \{x\})$, even in the presence of nuisance parameters!
 - ◆ **applying t and ρ is extremely fast!**

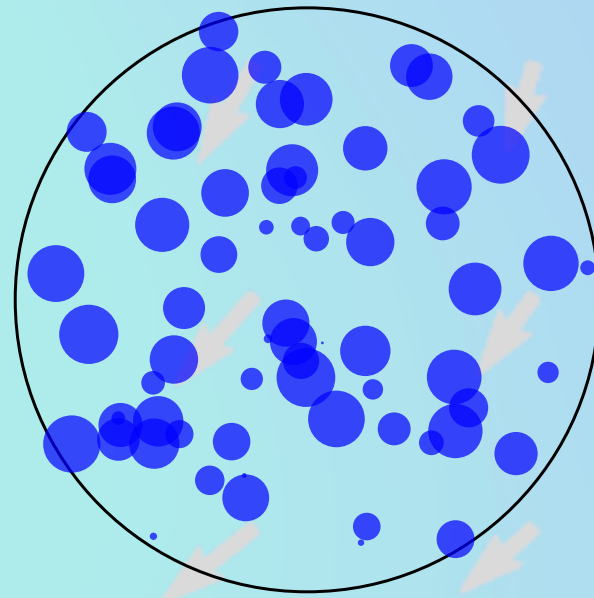
per-event operations and aggregation



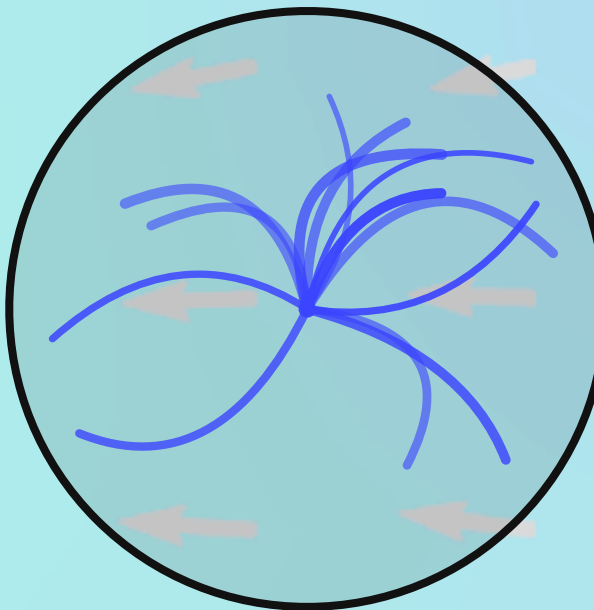
two axes: event complexity and dataset size

complexity of $x \uparrow$

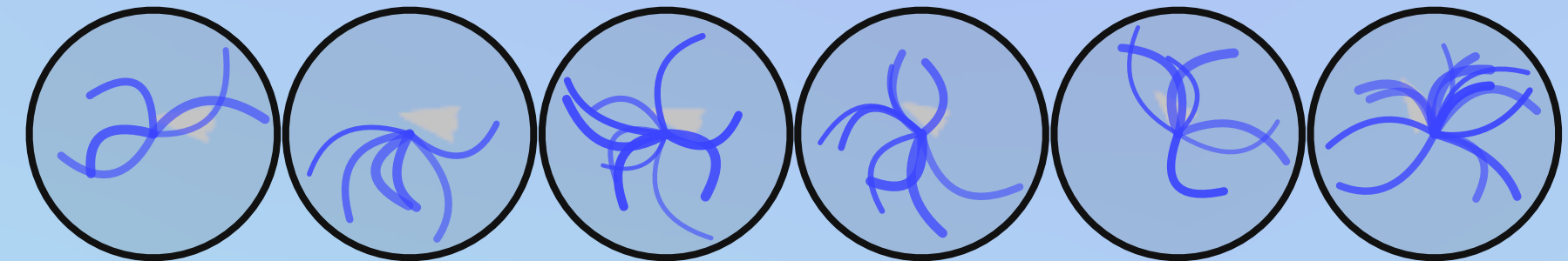
“raw”
data



coarse
summary



more informative



single
event

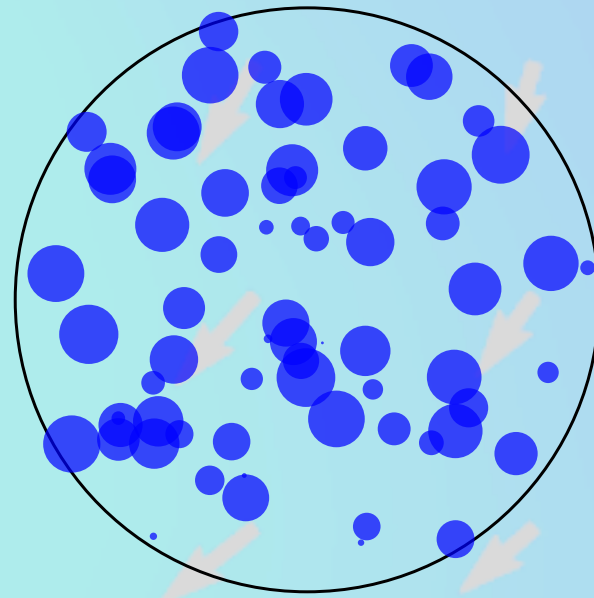
LHC
dataset

dataset size $N \rightarrow$

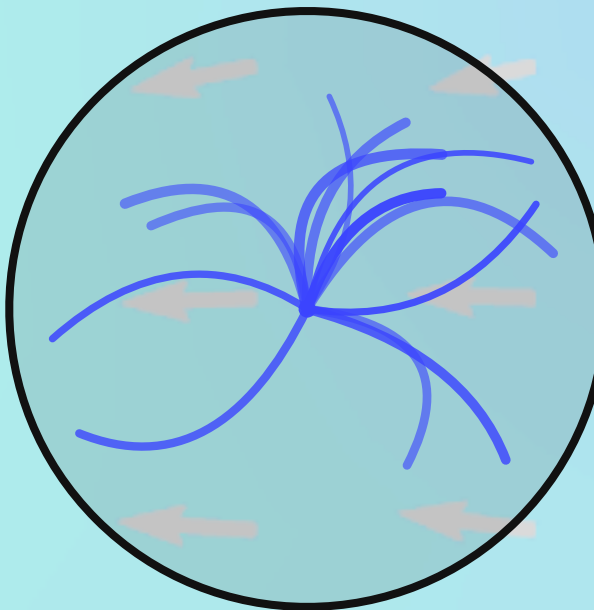
two axes: event complexity and dataset size

complexity of $x \uparrow$

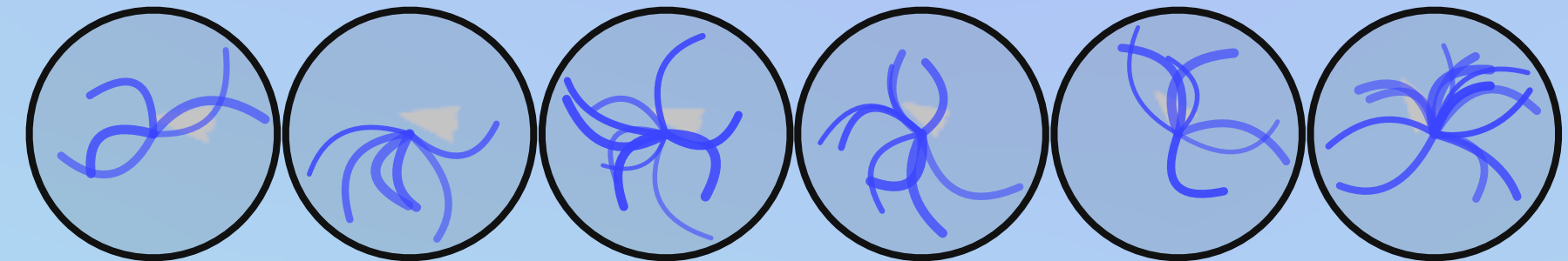
“raw”
data



coarse
summary



more informative



single
event

LHC
dataset

dataset size $N \rightarrow$

we recently showed $t(x)$ **learned from datasets of size 2**
generalizes to larger datasets:

- this vastly reduces the cost of deriving $t(x)$.
- a solution to a general “compression” problem for IID data that is **especially pertinent for collider physics.**

**It Just Takes Two:
Scaling Amortized Inference to Large Sets**

Antoine Wehenkel* 
Apple
awehenkel@apple.com

Michael Kagan 
SLAC National Accelerator Laboratory
makagan@slac.stanford.edu

Lukas Heinrich 
TU München
Lukas.Heinrich@cern.ch

Chris Pollard* 
University of Warwick
christopher.pollard@warwick.ac.uk

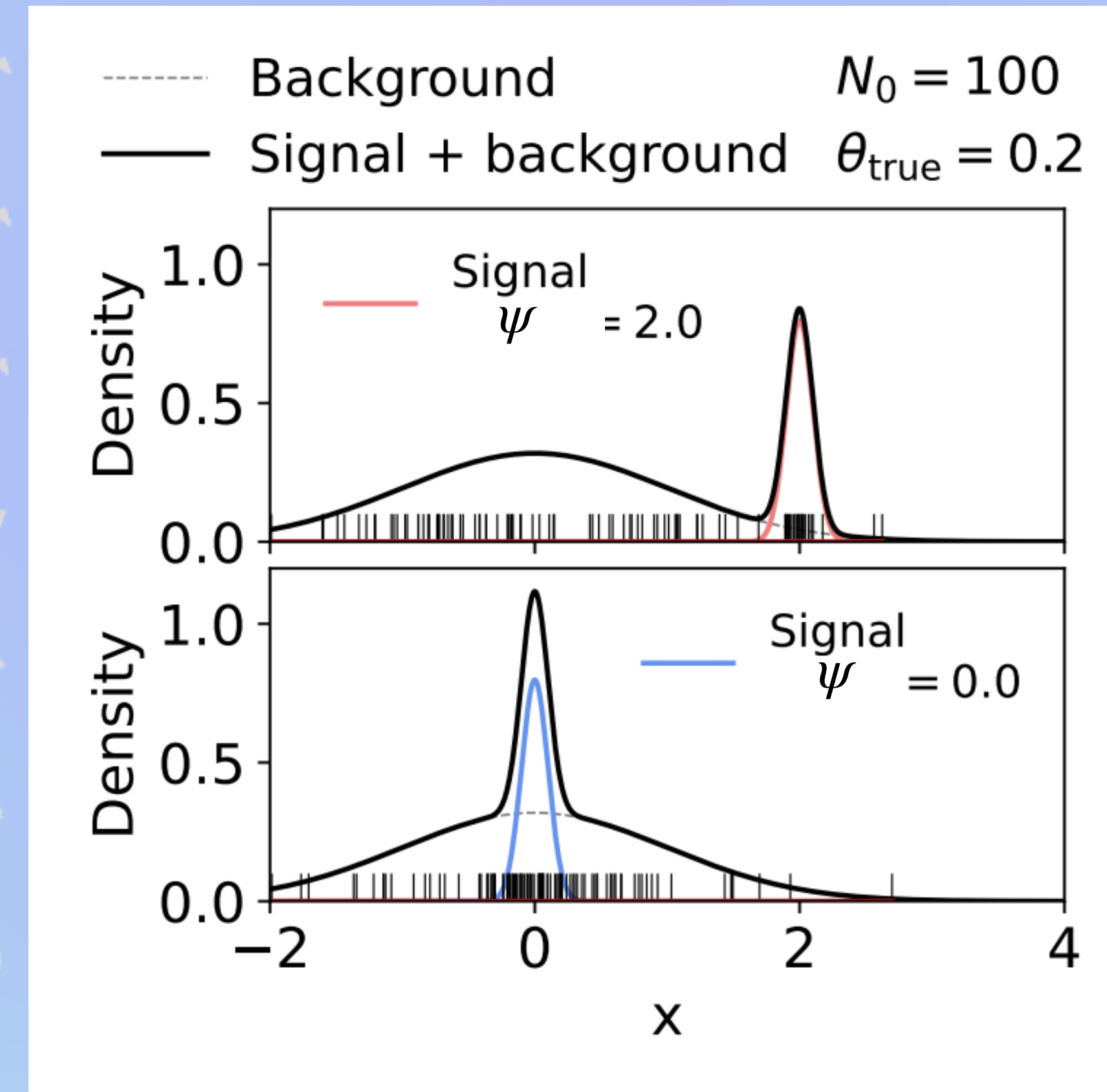
[preprint link](#)

**NeurIPS main track
(accepted)**

what might a “simulator” look like?

example: particle physics “bump hunt”

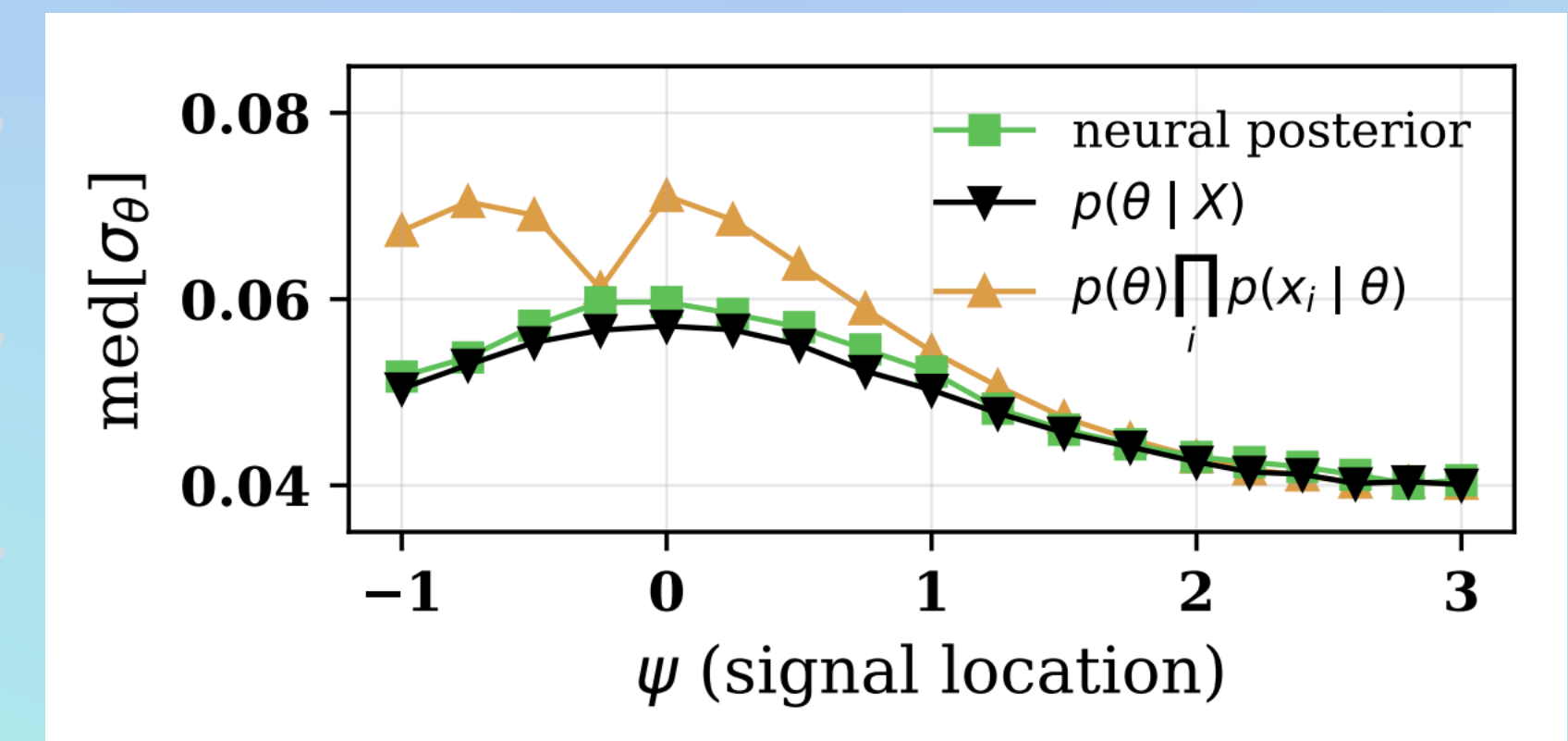
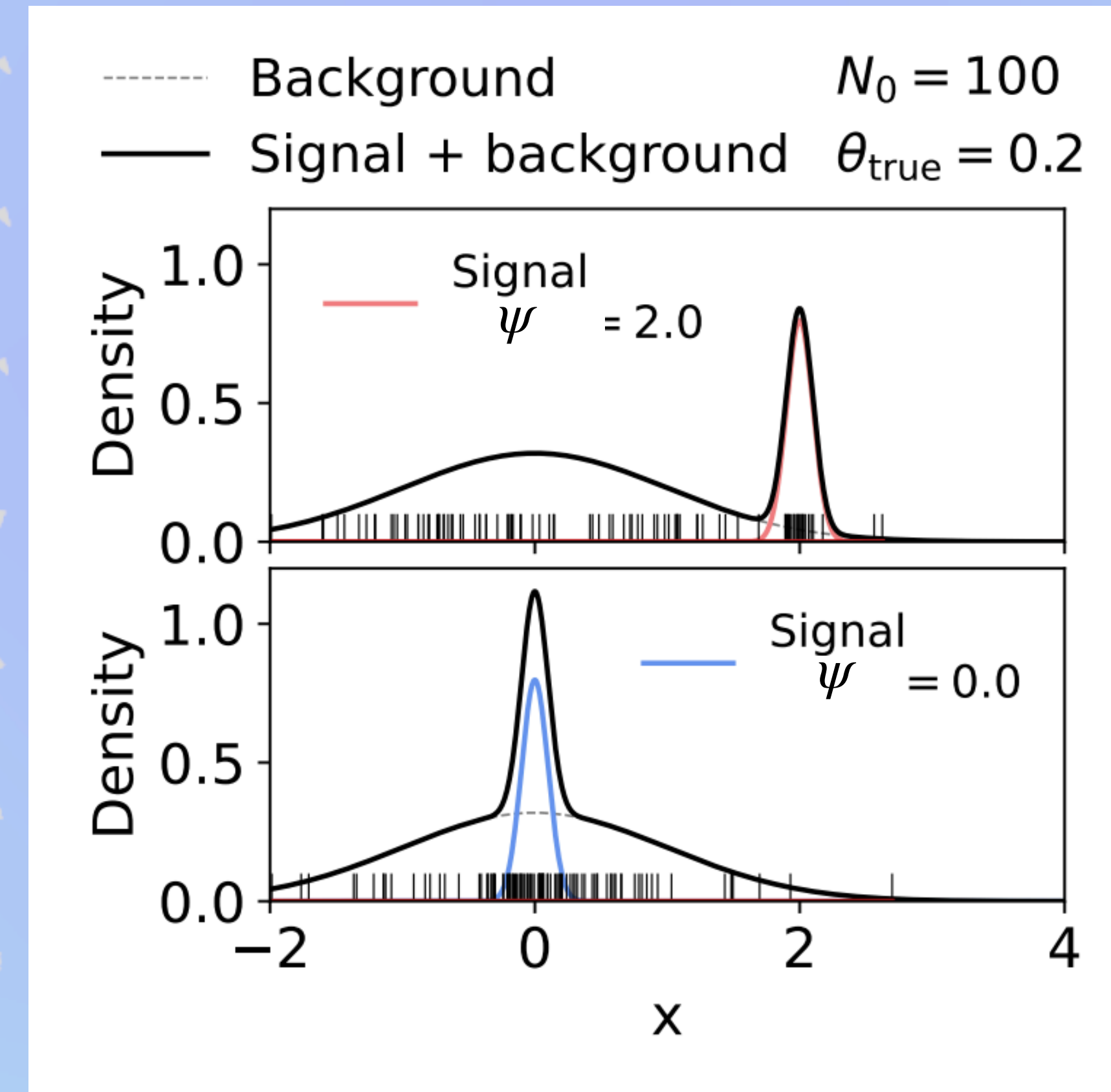
- ⊙ $\{x\}$ here are dijet masses
- ⊙ the goal is to extract the signal fraction (θ),
 - ◆ but we don't know exactly where the signal peak sits (global nuisance ψ).



what might a “simulator” look like?

example: particle physics “bump hunt”

- ◎ $\{x\}$ here are dijet masses
- ◎ the goal is to extract the signal fraction (θ),
 - ◆ but we don't know exactly where the signal peak sits (global nuisance ψ).
- ◎ the neural approximation emits the correct answer,
 - ◆ marginalizing properly against the global nuisance.
- ◎ checked against “brute force” MCMC.

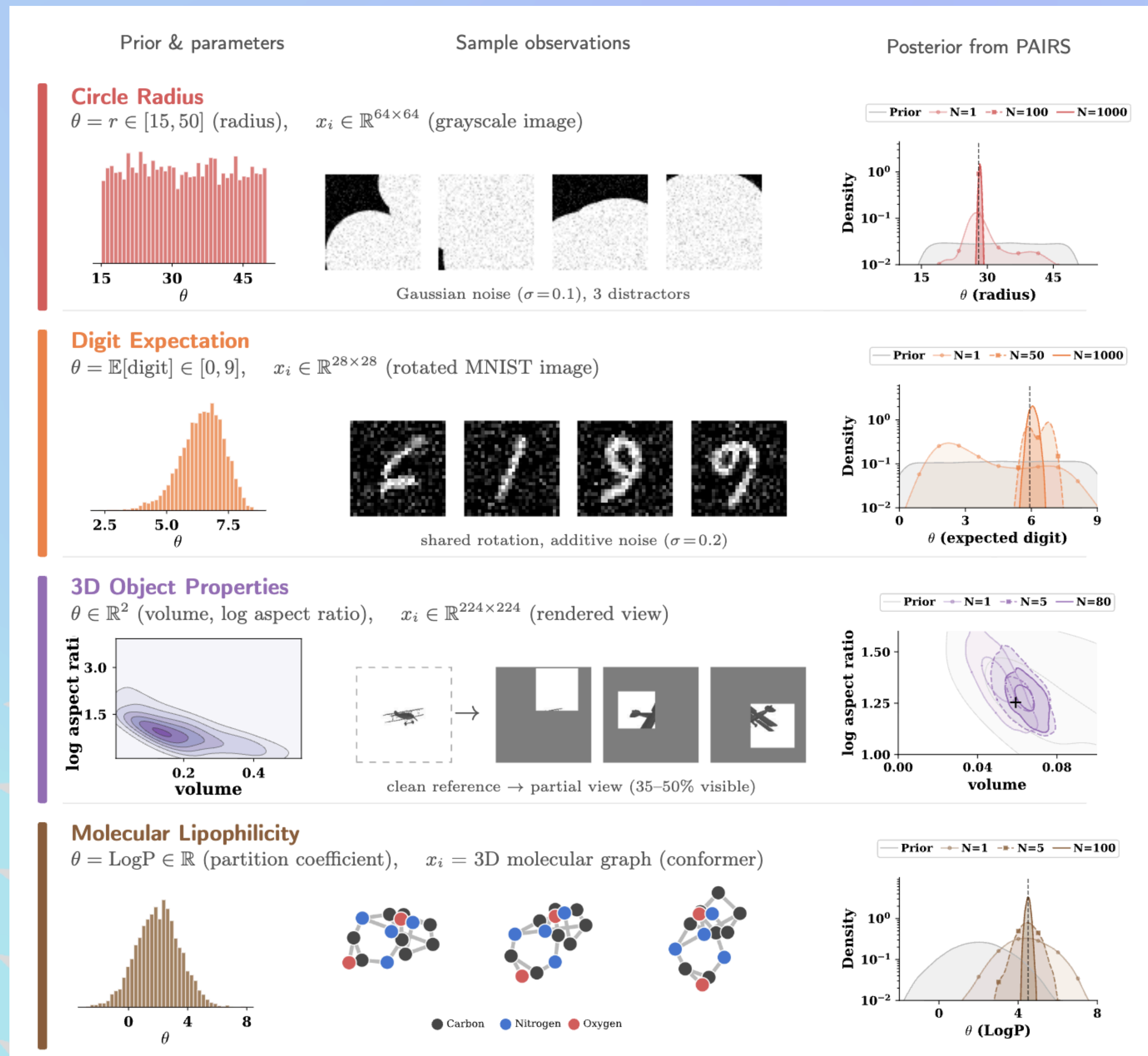


tested on a wide variety of tasks:

● image processing

● 3D view synthesis

● molecular chemistry

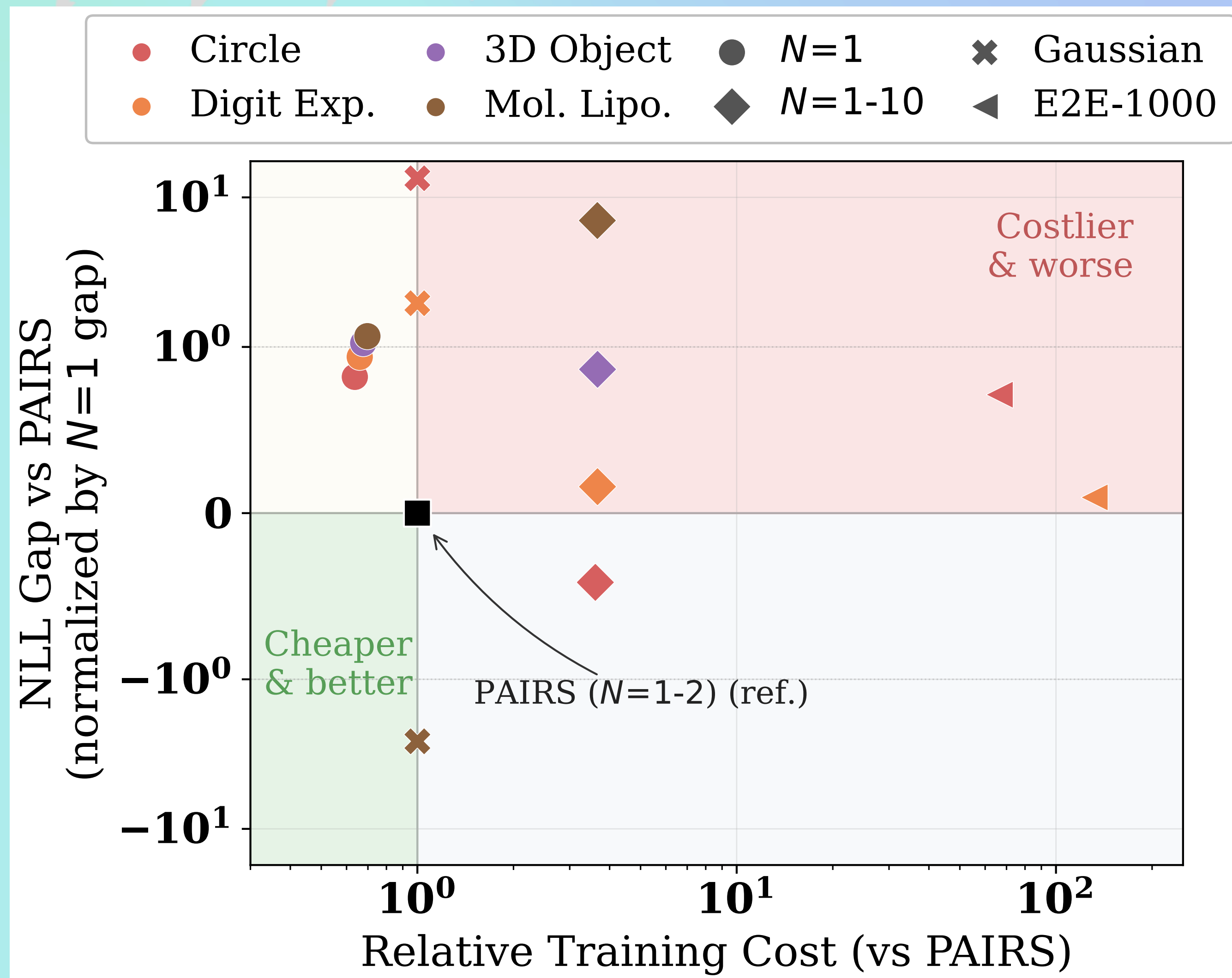


we find this procedure

● yields parameter constraints as good as end-to-end training,

● substantially reduces total computational cost,

● requires only two observations to be loaded in memory to derive $t(x)$.



to summarize,

● given a simulator from $\phi \rightarrow p(\{x\})$,

◆ one can derive a **compression** t of x
that is information-preserving w.r.t. ϕ .

◆ **averaging over** $t(x)$ **combines observations**
while **preserving information w.r.t. ϕ** .

● $\rho(\langle t \rangle) \approx p(\phi \mid \{x\})$ is **very fast to compute and update!**

potential applications in HEP, specifically:

- ◎ **derivation of optimal observables in the presence of nuisances,**
- ◎ **large-scale, compute efficient simulation-based inference (SBI),**
- ◎ **real-time analysis**
 - ◆ if t and ρ are pre-trained, one can e.g. infer a BSM signal strength on the fly.
- ◎ **online monitoring and calibration**



bonus

**inference of ϕ over a dataset, $\{x\}$,
via the *deep set* architecture**

