Neural Inference:

what does it mean?
why should I care?
why can’t I have it right now?

Chris Pollard, Warwick

before we can talk about neural inference,

we have to believe machine-learning is doing what it claims!

Multilayer Feedforward Networks are Universal Approximators

KURT HORNIK

Technische Universität Wien

MAXWELL STINCHCOMBE AND HALBERT WHITE

University of California, San Diego

(*Received* 16 *September* 1988; *revised and accepted* 9 *March* 1989)

Abstract—*This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.*

NNs can approximate any continuous function $f : \mathbb{R}^m \to \mathbb{R}^n$

to *arbitrary precision,*

given a finite number of free parameters
(+ enough training data and time).

neural inference:

extracting parameters of interest from observations
using NN approximations of a likelihood(-ratio) or posterior.

neural inference:

extracting parameters of interest from observations
using NN approximations of a likelihood(-ratio) or posterior.

how can we achieve this?

mostly it's a matter of constructing (and minimizing)
the correct loss function.

with a *simulator* s that,

given a parameter ϕ,

produces observables x

$$s : \phi \to p(x) \equiv p(x \mid \phi),$$

with a *simulator* s that,
given a parameter ϕ,
produces observables x

$$s : \phi \rightarrow p(x) \equiv p(x \,|\, \phi),$$

if we have many simulated samples (ϕ, x),
then we know how to learn the log-likelihood ratio

$$D^{\phi}_{\phi'}(x) \equiv \log \frac{p(x \,|\, \phi)}{p(x \,|\, \phi')}.$$

we know how to learn the log-likelihood ratio

$$D^{\phi}_{\phi'}(x) \equiv \log \frac{p(x\,|\,\phi)}{p(x\,|\,\phi')}.$$

imagine the case that ϕ only has two possible values: A, B.

$$D^{A}_{B}(x) \equiv \log \frac{p(x\,|\,A)}{p(x\,|\,B)}$$

is simply what a *discriminator* between A and B learns to emit.

w… w to learn the log-likelihood ratio

“density ratio estimation”
https://arxiv.org/abs/1506.02169

$$D^{\phi}_{\phi'}(x) \equiv \log \frac{p(x\,|\,\phi)}{p(x\,|\,\phi')}.$$

imagine the case that ϕ only has two possible values: A, B.

$$D^{A}_{B}(x) \equiv \log \frac{p(x\,|\,A)}{p(x\,|\,B)}$$

is simply what a *discriminator* between A and B learns to emit.

given a simulator $p(x \mid \phi)$, we also know, how to learn

$$p(\phi \mid x) \propto p(x \mid \phi)\, p(\phi)$$

for some family of posteriors p_η parameterized by η:

given a simulator $p(x \mid \phi)$, we also know, how to learn

$$p(\phi \mid x) \propto p(x \mid \phi)\, p(\phi)$$

for some family of posteriors p_η parameterized by η:

$p(\phi)$ is the distribution over ϕ used in training,

and we learn the function $f : x \to \eta$ by minimizing the loss

$$L(\phi, x) = -\log p_{\eta=f(x)}(\phi).$$

"variational inference"
"neural posterior estimation"
https://arxiv.org/abs/1601.00670

gi[illegible]ator $p(x \,|\, \phi)$, we also know, how to learn

$$p(\phi \,|\, x) \propto p(x \,|\, \phi)\, p(\phi)$$

for some family of posteriors p_η parameterized by η:

$p(\phi)$ is the distribution over ϕ used in training,

and we learn the function $f : x \to \eta$ by minimizing the loss

$$L(\phi, x) = -\log p_{\eta=f(x)}(\phi).$$

summary:

if we believe NNs work the way they should…

then given a simulator $p(x\,|\,\phi)$

we can learn an *arbitrarily precise approximation* of

$$\log \frac{p(x\,|\,\phi)}{p(x\,|\,\phi')} \quad \text{or} \quad p(\phi\,|\,x).$$

summary:

if we believe NNs work the way they should…

then given a simulator $p(x\,|\,\phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

$$\log \frac{p(x\,|\,\phi)}{p(x\,|\,\phi')} \quad \text{or} \quad p(\phi\,|\,x).$$

summary:

if we believe NNs work the way they should…

then given a simulator $p(x \mid \phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

$$\log \frac{p(x \mid \phi)}{p(x \mid \phi')} \quad \text{or} \quad p(\phi \mid x).$$

leads to *best possible* constraints

summary:

if we believe NNs work the way they should…

then given a simulator $p(x \mid \phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

the entire analysis is "basically done."

$$\log \frac{p(x \mid \phi)}{p(x \mid \phi')} \quad \text{or} \quad p(\phi \mid x).$$

leads to *best possible* constraints

“simulation-based inference”
https://arxiv.org/abs/1911.01429

summary:

if we believe NNs work the way they should…

then given a simulator $p(x \mid \phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

the entire analysis is “basically done.”

$$\log \frac{p(x \mid \phi)}{p(x \mid \phi')} \quad \text{or} \quad p(\phi \mid x).$$

leads to *best possible* constraints

too good to be true?

too good to be true?

at least three* open “issues”
with neural simulation-based inference (nSBI):

 *convergence of function approximations*

 *global nuisance parameters*

 *model mis-specification*

*that I could quickly think of
when assembling this talk

convergence of function approximations

adobe stock

convergence of function approximations

the likelihood-ratio over an entire dataset
may require very, very good approximations
of the per-event likelihood-ratio to behave properly.

$$L(x \mid \phi) = \prod_i L(x_i \mid \phi)$$

(for independent observations)

convergence of function approximations

the likelihood-ratio over an entire dataset
may require very, very good approximations
of the per-event likelihood-ratio to behave properly.

$$L(x \mid \phi) = \prod_i L(x_i \mid \phi)$$

(for independent observations)

this may be *difficult*,
but at least it can be *verified* on toy datasets.

https://openart.ai/community/NOz2NbLjiXRdF0pRDPuL

global
nuisance parameters

global nuisance parameters

in the presence of global nuisance parameters,
per-event likelihoods do not easily compose.

we need to be able to profile against
or marginalize over global nuisance parameters.

global nuisance parameters

two main approaches:

1) learn the likelihood/posterior *parameterized* in the nuisances and profile/marginalize after the fact, or

2) perform dataset-wide (ensemble) learning.

global nuisance parameters

works but can be painful

two main approaches:

1) learn the likelihood/posterior *parameterized* in the nuisances and profile/marginalize after the fact, or

2) perform dataset-wide (ensemble) learning.

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Submitted to: Rep. Prog. Phys.

CERN-EP-2024-298
December 3, 2024

Measurement of off-shell Higgs boson production in the $H^* \to ZZ \to 4\ell$ decay channel using a neural simulation-based inference technique in 13 TeV pp collisions with the ATLAS detector

The ATLAS Collaboration

A measurement of off-shell Higgs boson production in the $H^* \to ZZ \to 4\ell$ decay channel is presented. The measurement uses 140 fb^{-1} of proton–proton collisions at $\sqrt{s} = 13$ TeV collected by the ATLAS detector at the Large Hadron Collider and supersedes the previous result in this decay channel using the same dataset. The data analysis is performed using a neural simulation-based inference method, which builds per-event likelihood ratios using neural networks. The observed (expected) off-shell Higgs boson production signal strength in the $ZZ \to 4\ell$ decay channel at 68% CL is $0.87^{+0.75}_{-0.54}$ ($1.00^{+1.04}_{-0.95}$). The evidence for off-shell Higgs boson production using the $ZZ \to 4\ell$ decay channel has an observed (expected) significance of 2.5σ (1.3σ). The expected result represents a significant improvement relative to that of the previous analysis of the same dataset, which obtained an expected significance of 0.5σ. When combined with the most recent ATLAS measurement in the $ZZ \to 2\ell 2\nu$ decay channel, the evidence for off-shell Higgs boson production has an observed (expected) significance of 3.7σ (2.4σ). The off-shell measurements are combined with the measurement of on-shell Higgs boson production to obtain constraints on the Higgs boson total width. The observed (expected) value of the Higgs boson width at 68% CL is $4.3^{+2.7}_{-1.9}$ ($4.1^{+3.5}_{-3.4}$) MeV.

arXiv:2412.01548

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Submitted to: Rep. Prog. Phys.

CERN-EP-2024-305
December 3, 2024

An implementation of neural simulation-based inference for parameter estimation in ATLAS

The ATLAS Collaboration

Neural simulation-based inference is a powerful class of machine-learning-based methods for statistical inference that naturally handles high-dimensional parameter estimation without the need to bin data into low-dimensional summary histograms. Such methods are promising for a range of measurements, including at the Large Hadron Collider, where no single observable may be optimal to scan over the entire theoretical phase space under consideration, or where binning data into histograms could result in a loss of sensitivity. This work develops a neural simulation-based inference framework for statistical inference, using neural networks to estimate probability density ratios, which enables the application to a full-scale analysis. It incorporates a large number of systematic uncertainties, quantifies the uncertainty due to the finite number of events in training samples, develops a method to construct confidence intervals, and demonstrates a series of intermediate diagnostic checks that can be performed to validate the robustness of the method. As an example, the power and feasibility of the method are assessed on simulated data for a simplified version of an off-shell Higgs boson couplings measurement in the four-lepton final states. This approach represents an extension to the standard statistical methodology used by the experiments at the Large Hadron Collider, and can benefit many physics analyses.

arXiv:2412.01600

ATLAS recently published a measurement of off-shell Higgs production using this approach:

→ it does work!

→ very high CPU costs

→ currently limited to relatively small datasets.

global nuisance parameters

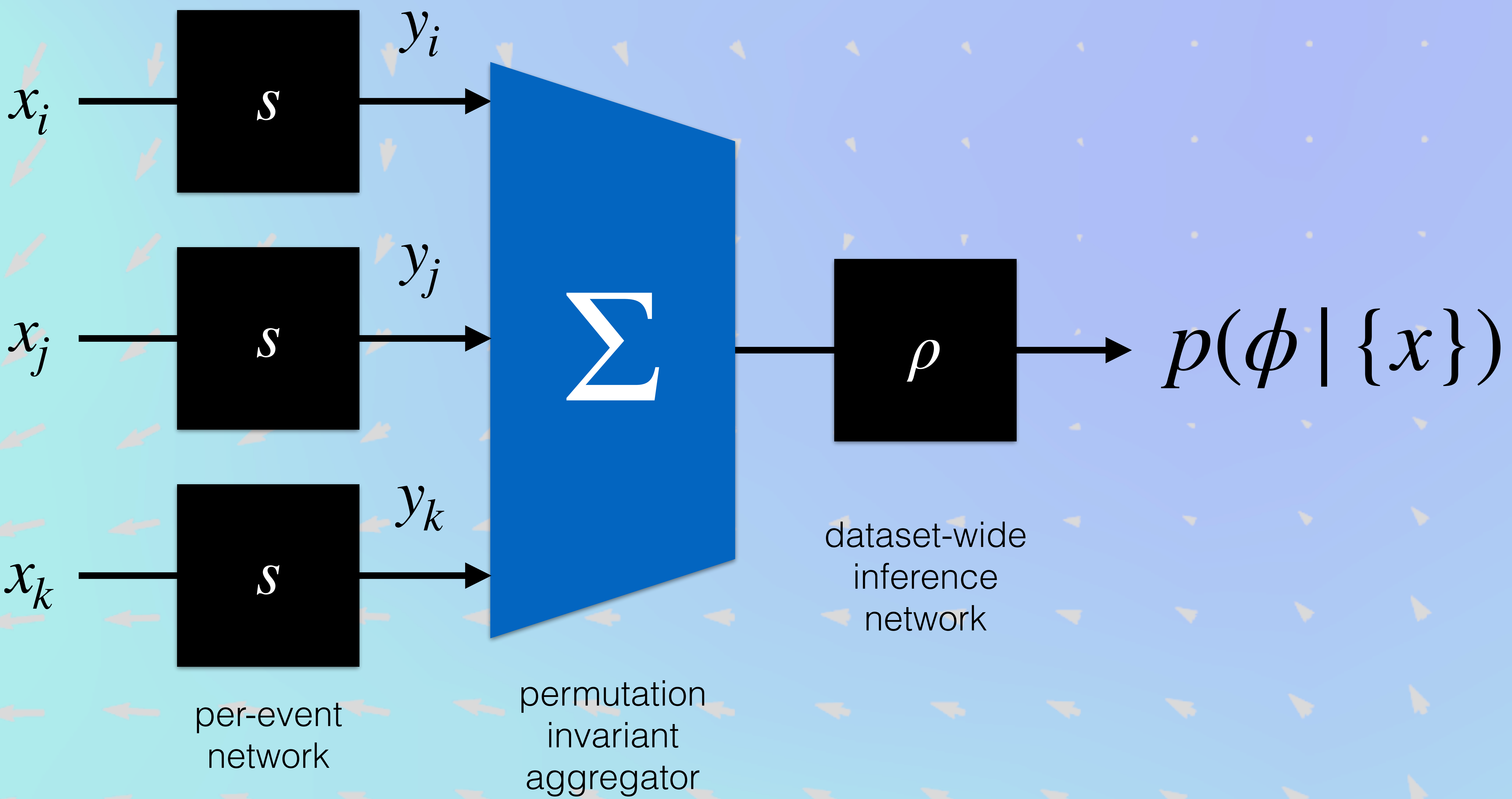
works but can be painful

two main approaches:

1) learn the likelihood/posterior *parameterized* in the nuisances and profile/marginalize after the fact, or

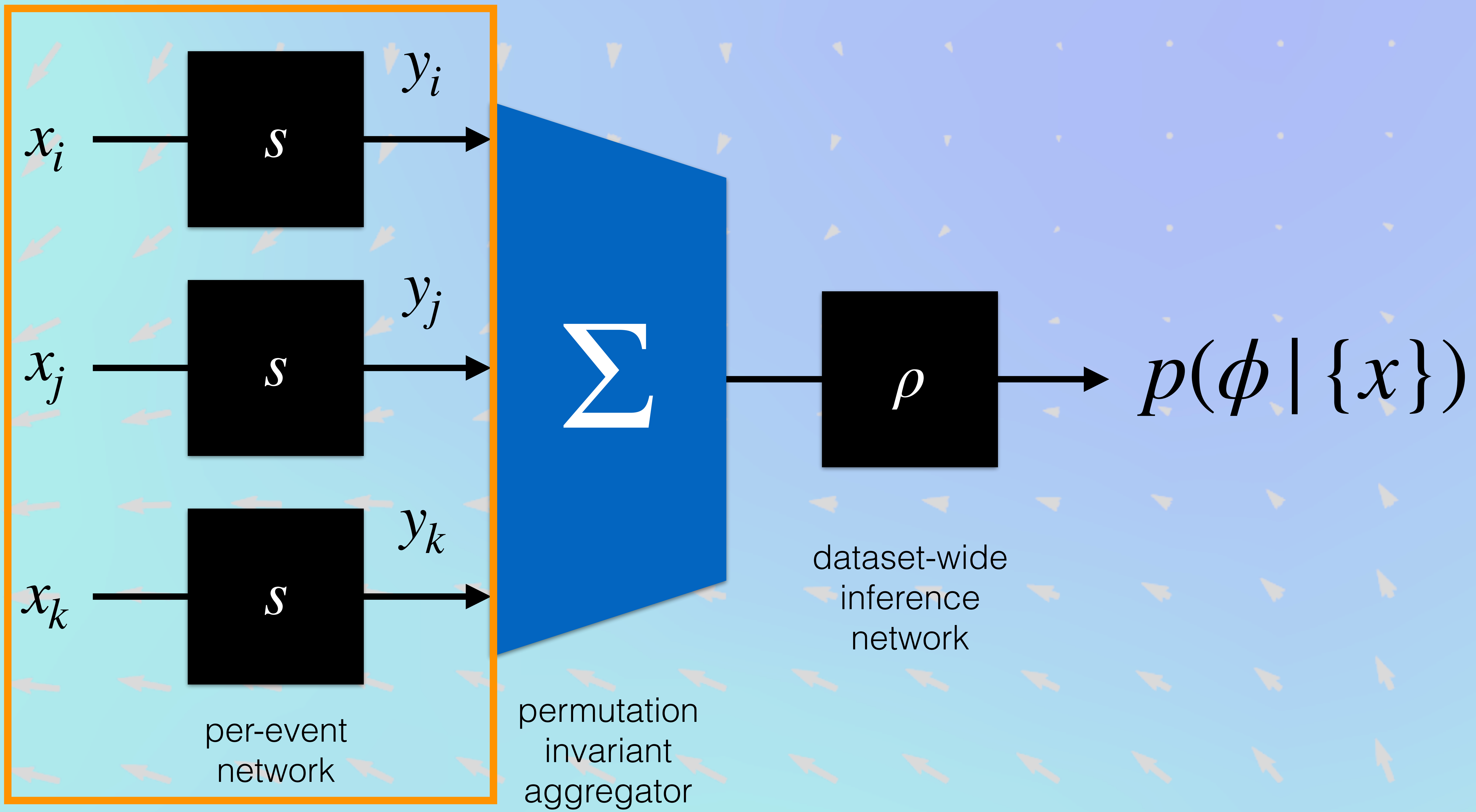
2) perform dataset-wide (ensemble) learning.

has some very nice properites

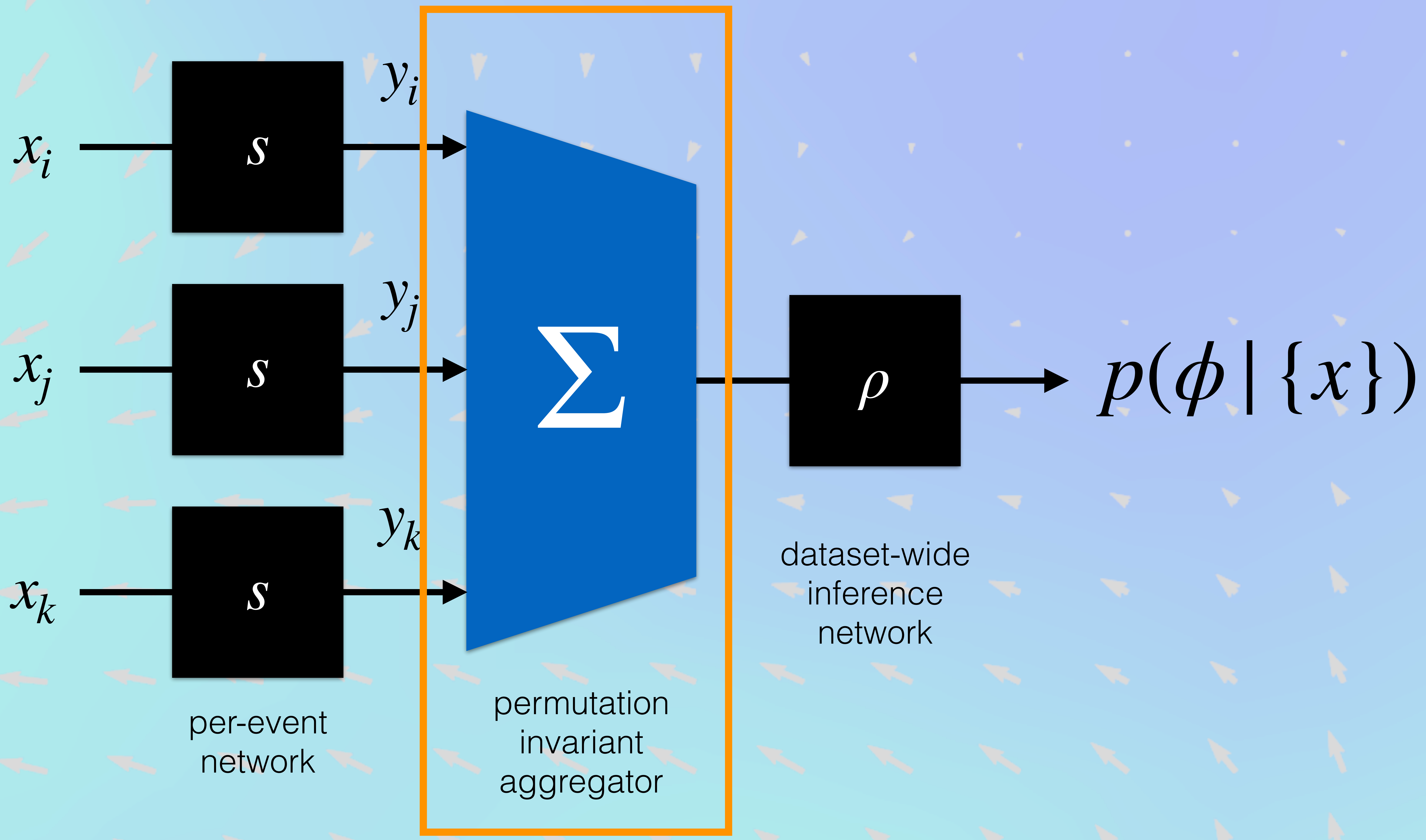


inference over an *ensemble*, $\{x\}$,
via the *deep set* architecture

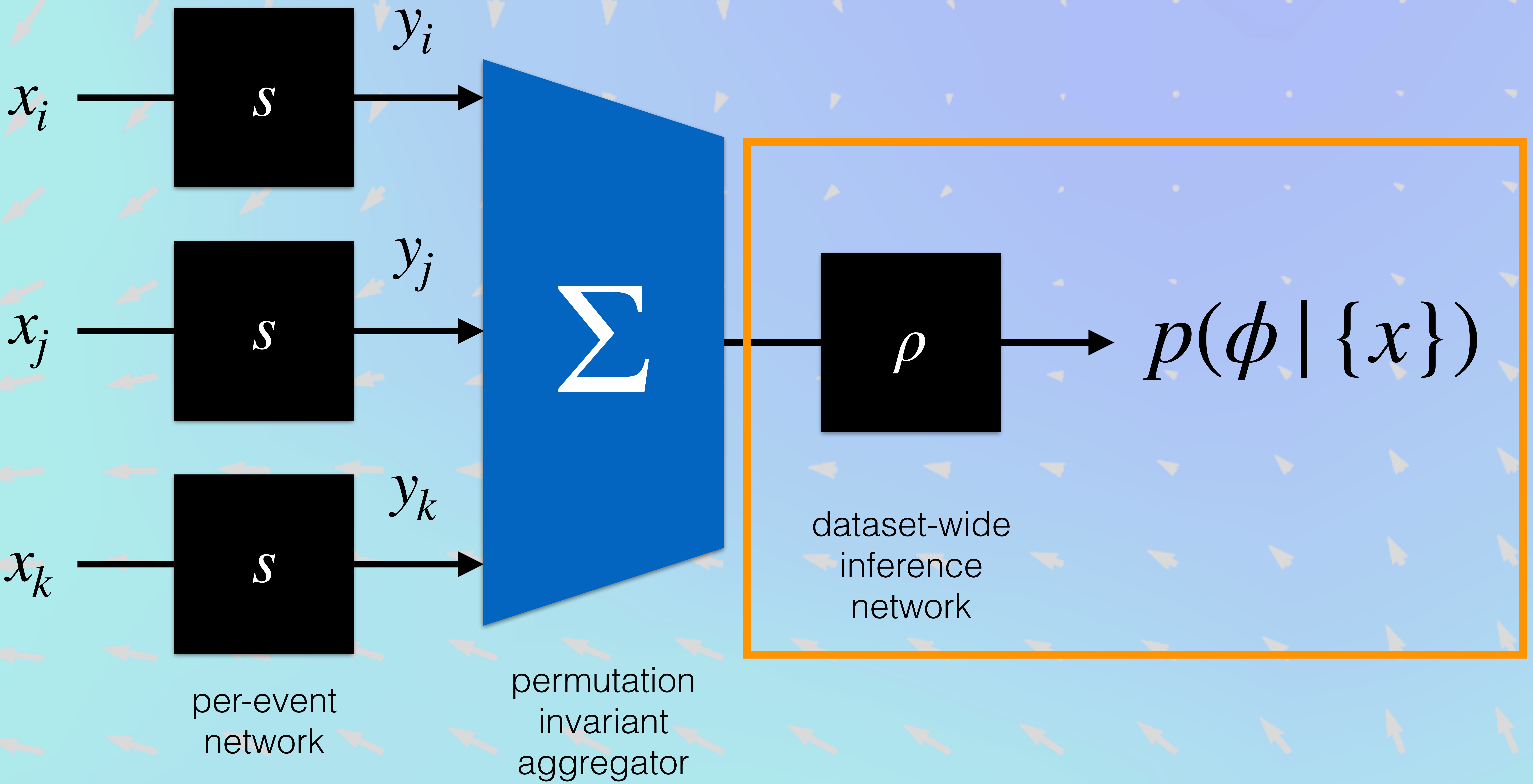
derive a vector summary observable per event



sum over these per-event summaries



perform variational inference on this dataset-wide summary

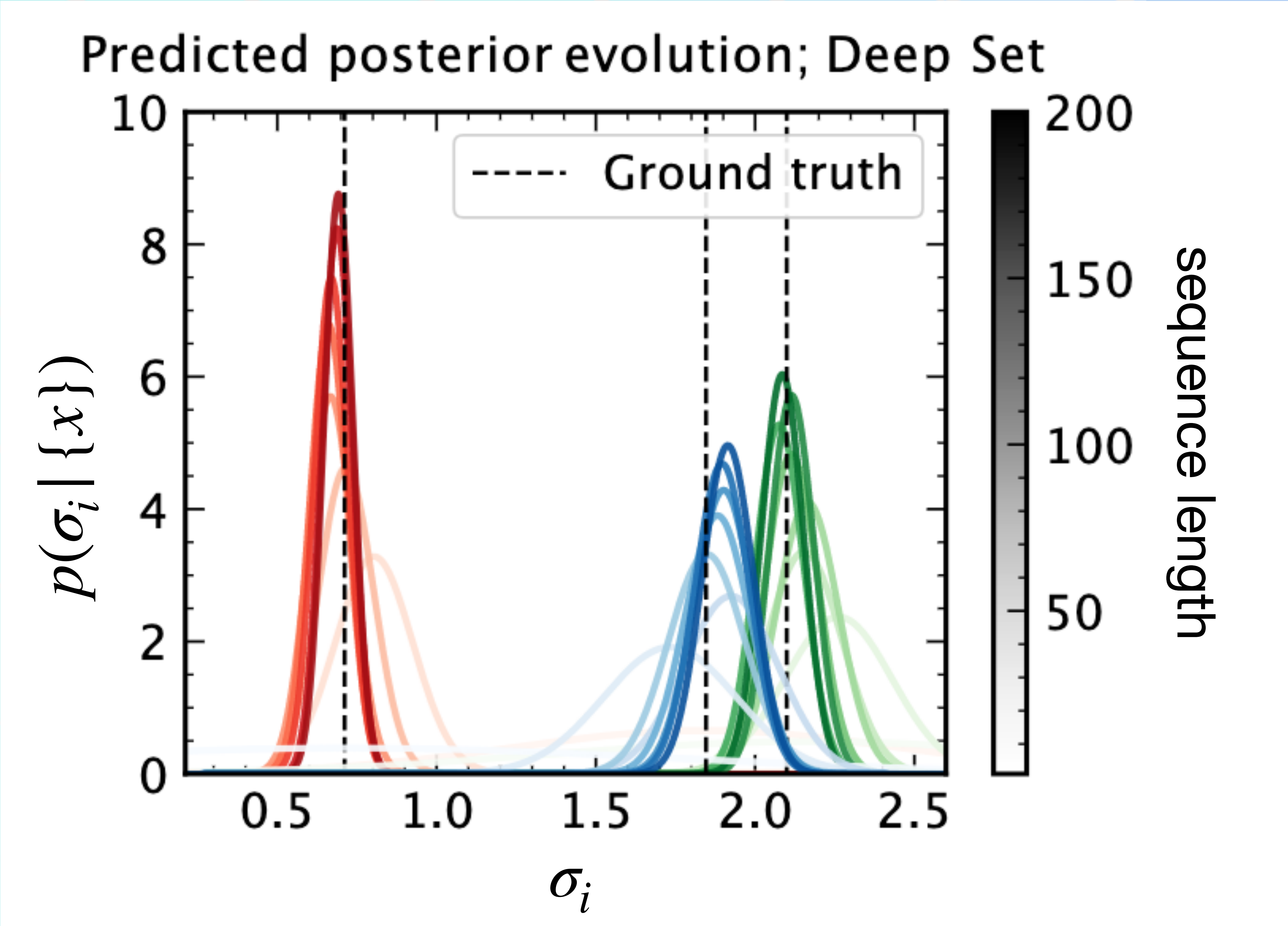


very simple case: inferring diagonal components of $\vec{\sigma}$ from observations

$$x \sim \mathcal{N}(\overrightarrow{\mu}, \mathrm{diag}(\vec{\sigma}^2))$$

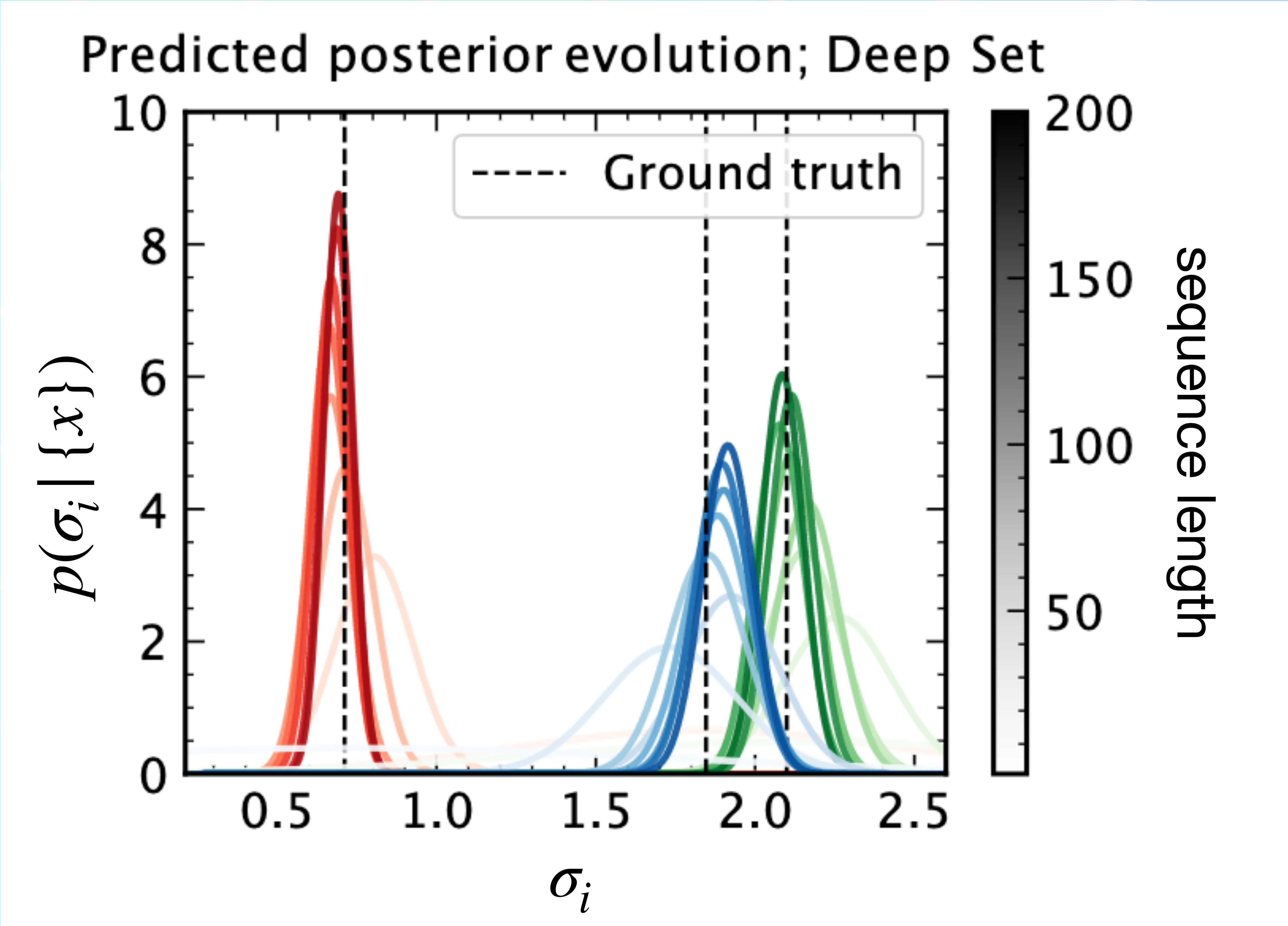
very simple case: inferring diagonal components of $\vec{\sigma}$ from observations

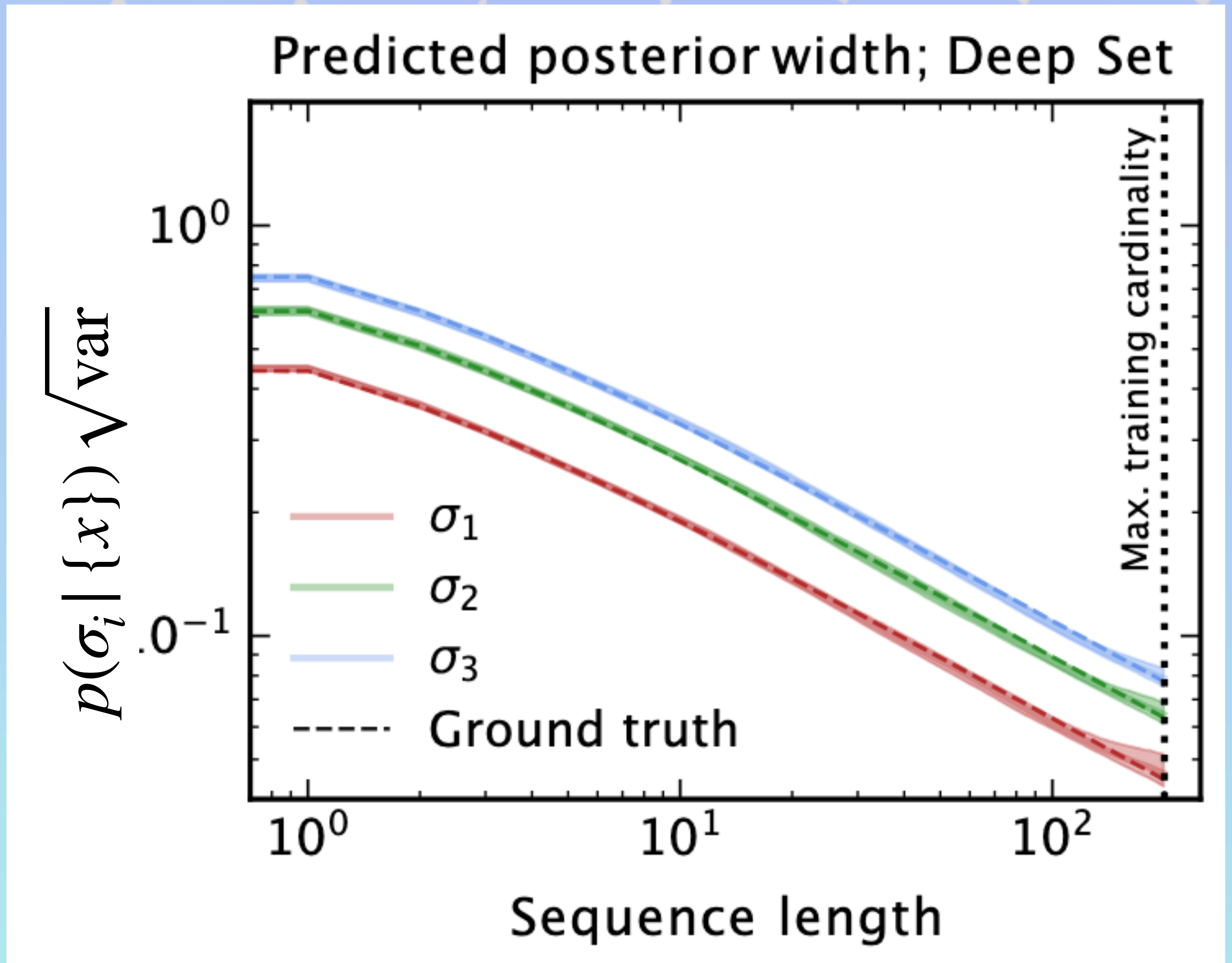
$$x \sim \mathcal{N}(\vec{\mu}, \mathrm{diag}(\vec{\sigma}^2))$$



very simple case: inferring diagonal components of $\vec{\sigma}$ from observations

$$x \sim \mathcal{N}(\vec{\mu}, \mathrm{diag}(\vec{\sigma}^2))$$





particle-physics “inspired” example:

goal: extract signal fraction θ,

without knowing *a priori* the signal location, θ_ν.
(θ_ν is a global NP.)

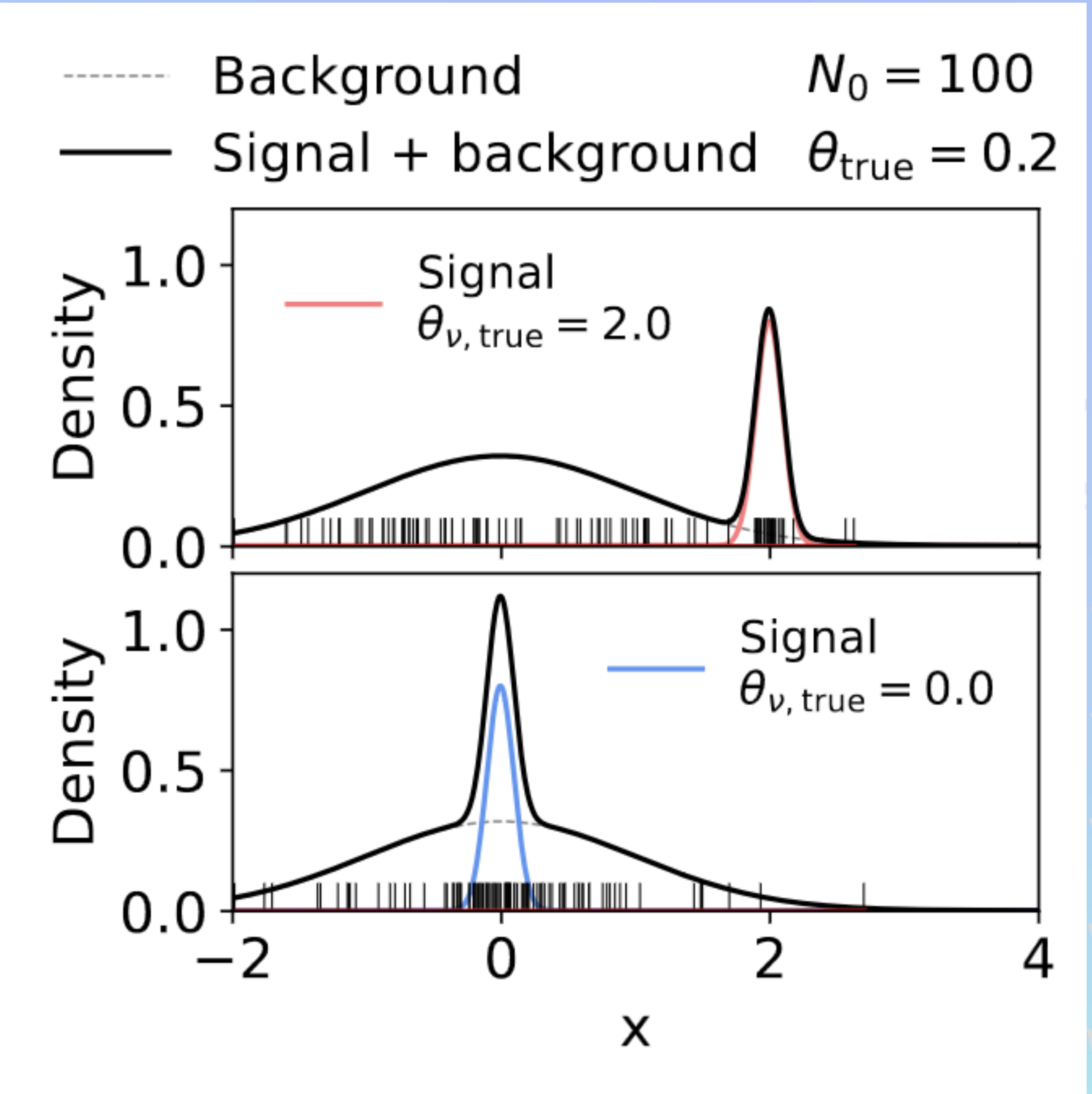
plus a wide background contribution.

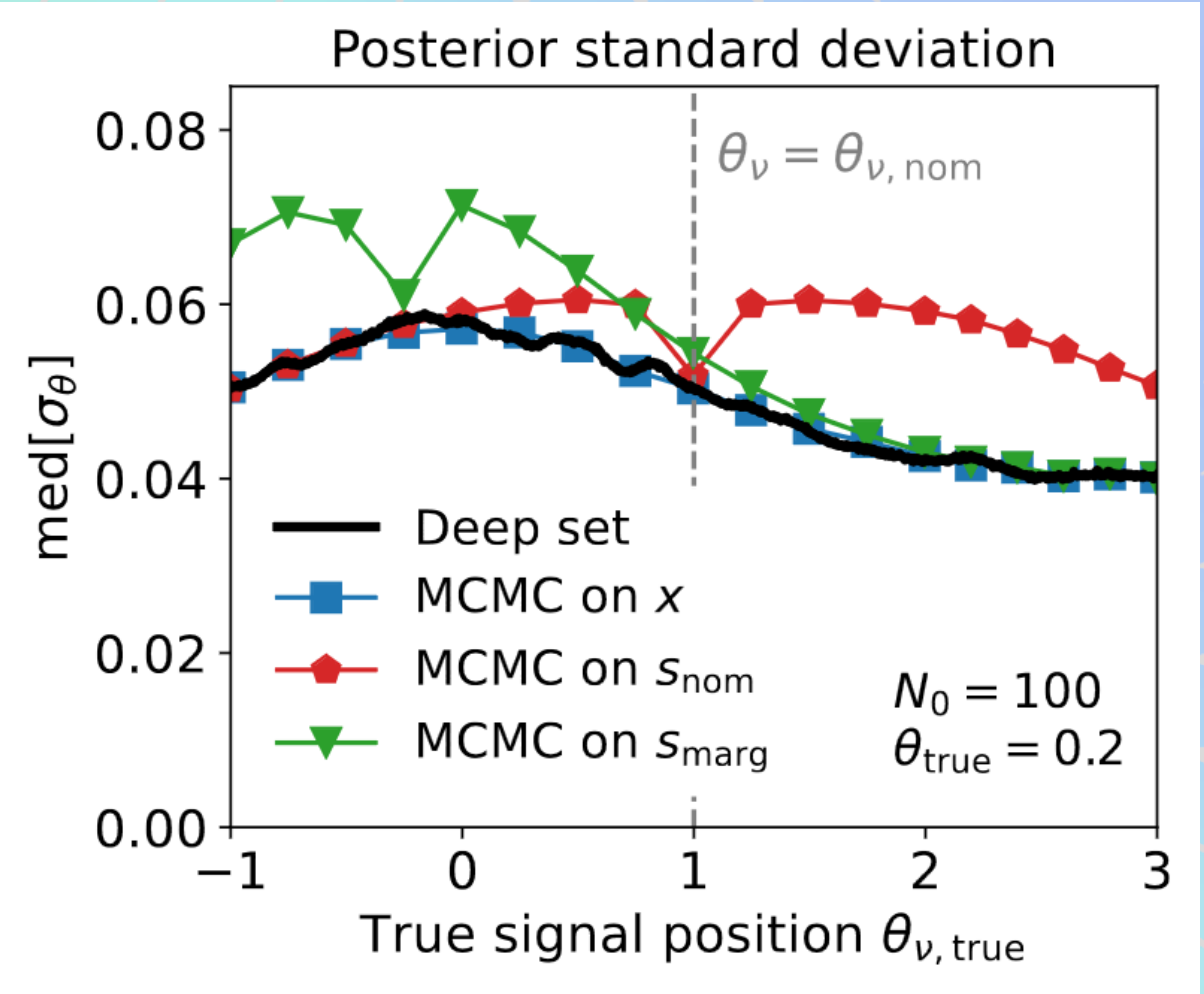
particle-physics “inspired” example:

goal: extract signal fraction θ,

without knowing *a priori* the signal location, θ_ν.
(θ_ν is a global NP.)

plus a wide background contribution.





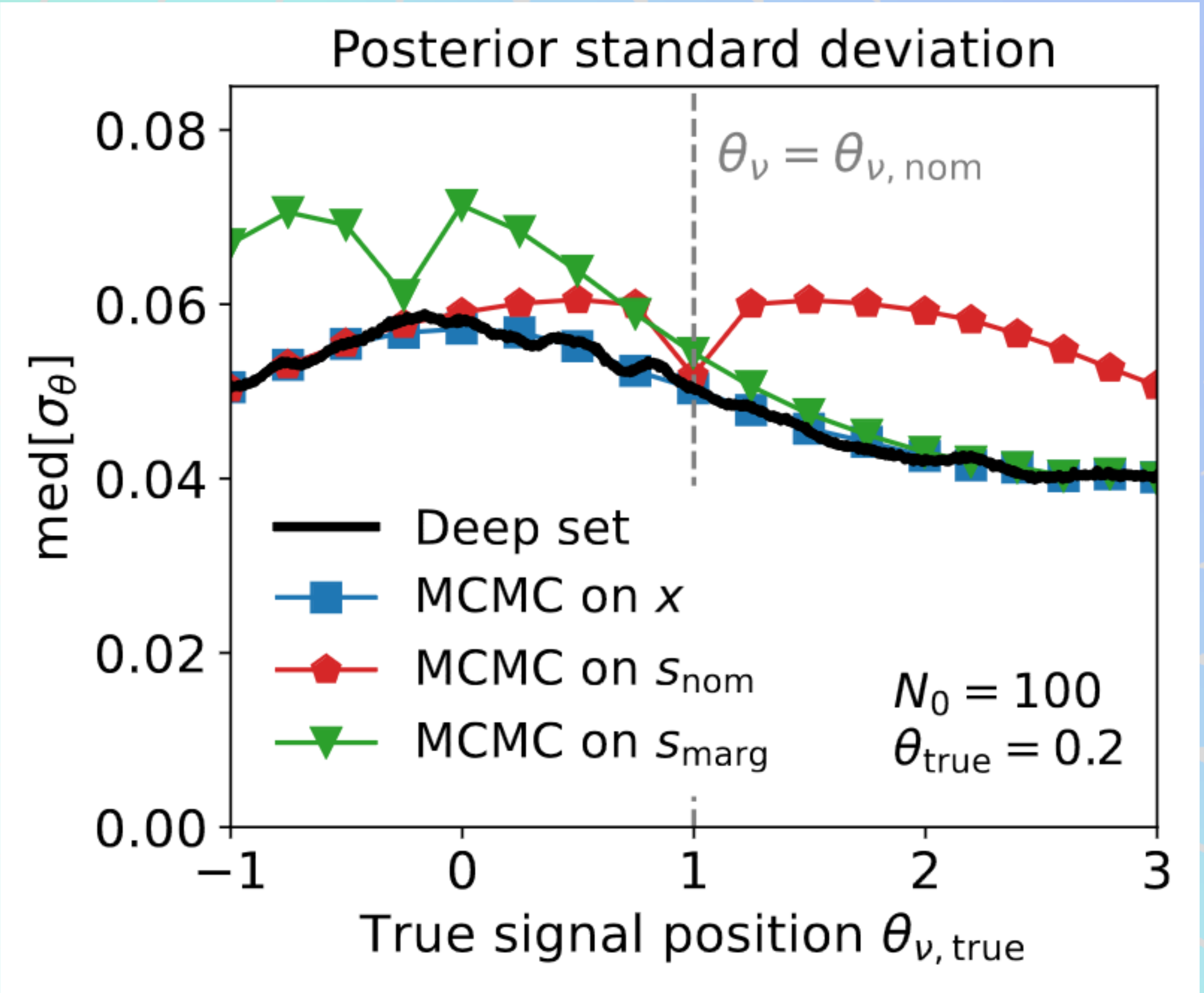
as we sweep θ_ν,

the **deep-set posterior**

agrees with

brute-force (MCMC)

for posterior estimation on $\{x\}$.

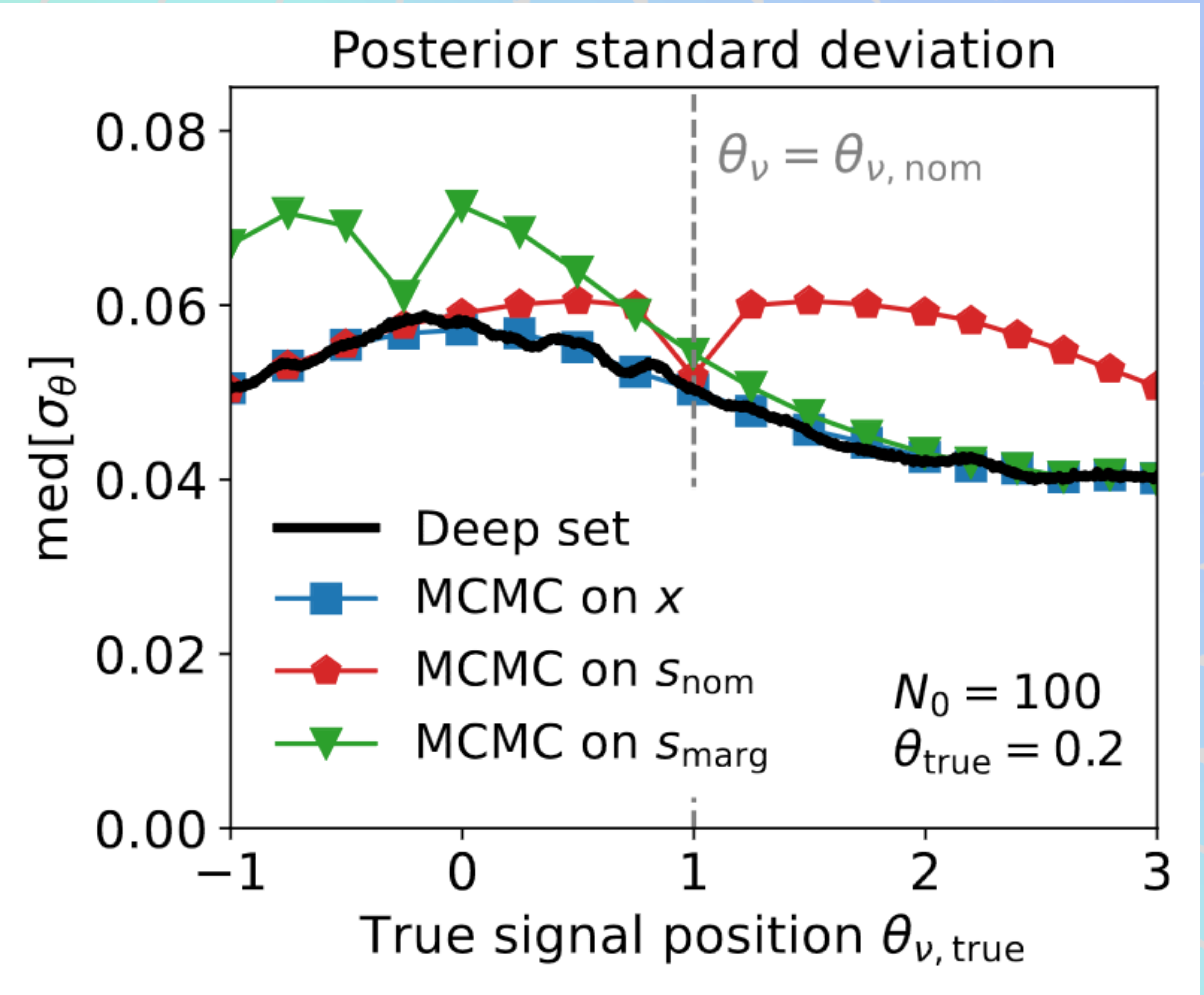


discriminators learned per-event on x

at the nominal signal position

and marginalized over θ_ν

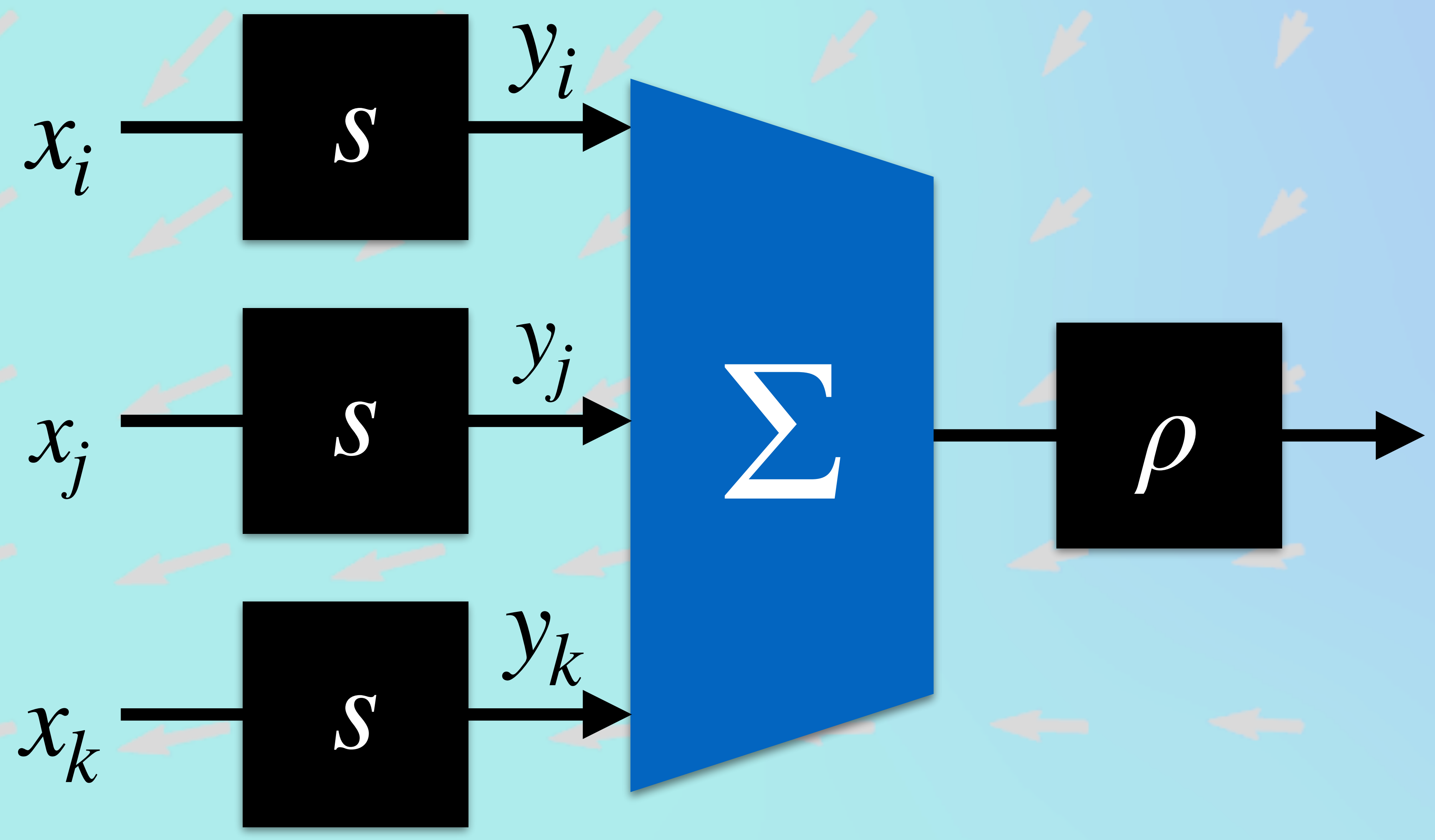
do not preserve enough information to build the correct posterior $\forall\, \theta_\nu$.



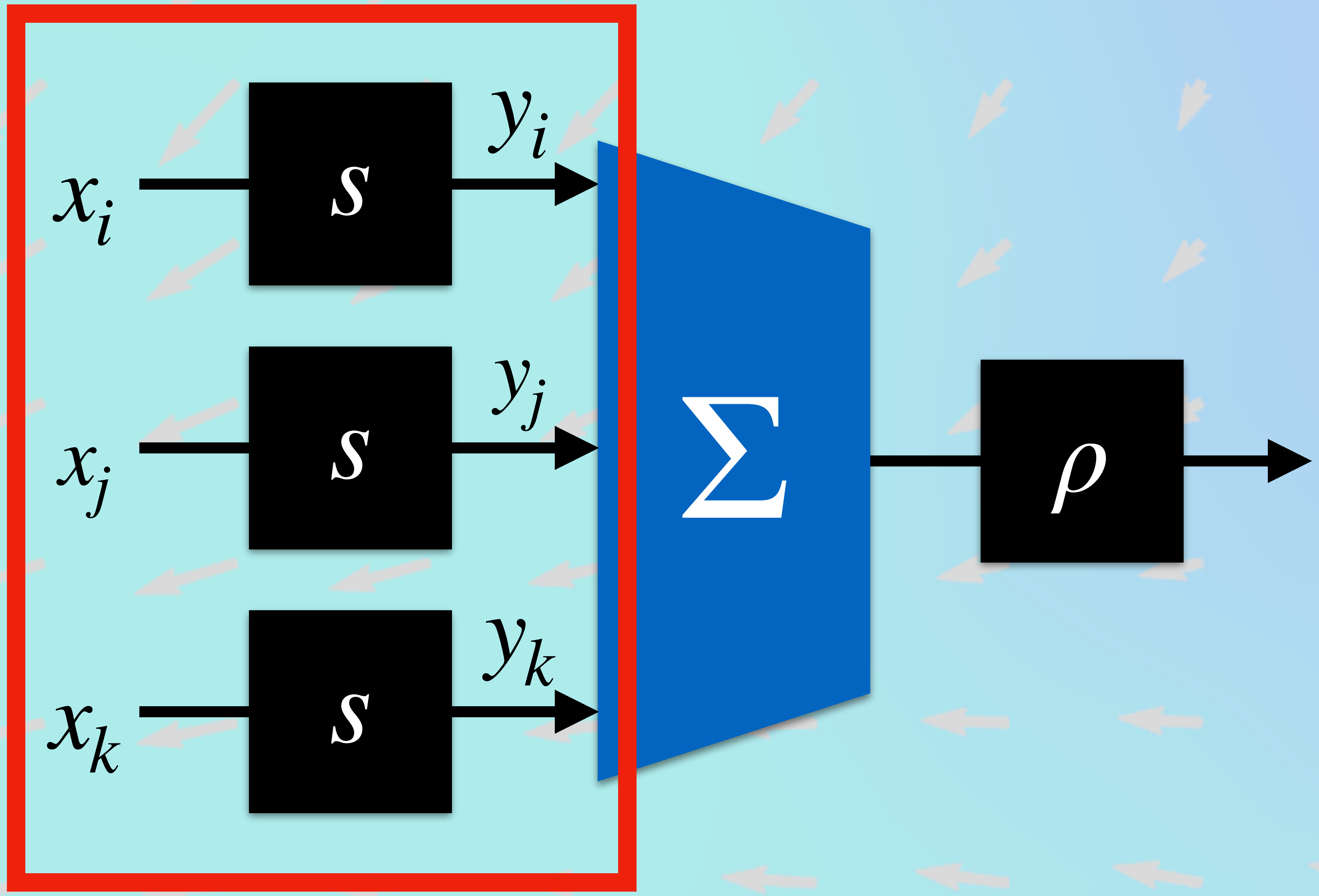
the deep set is
many orders of magnitude faster to run

than MCMC

or a “conventional” profiled likelihood fit.



coming back to
our network architecture

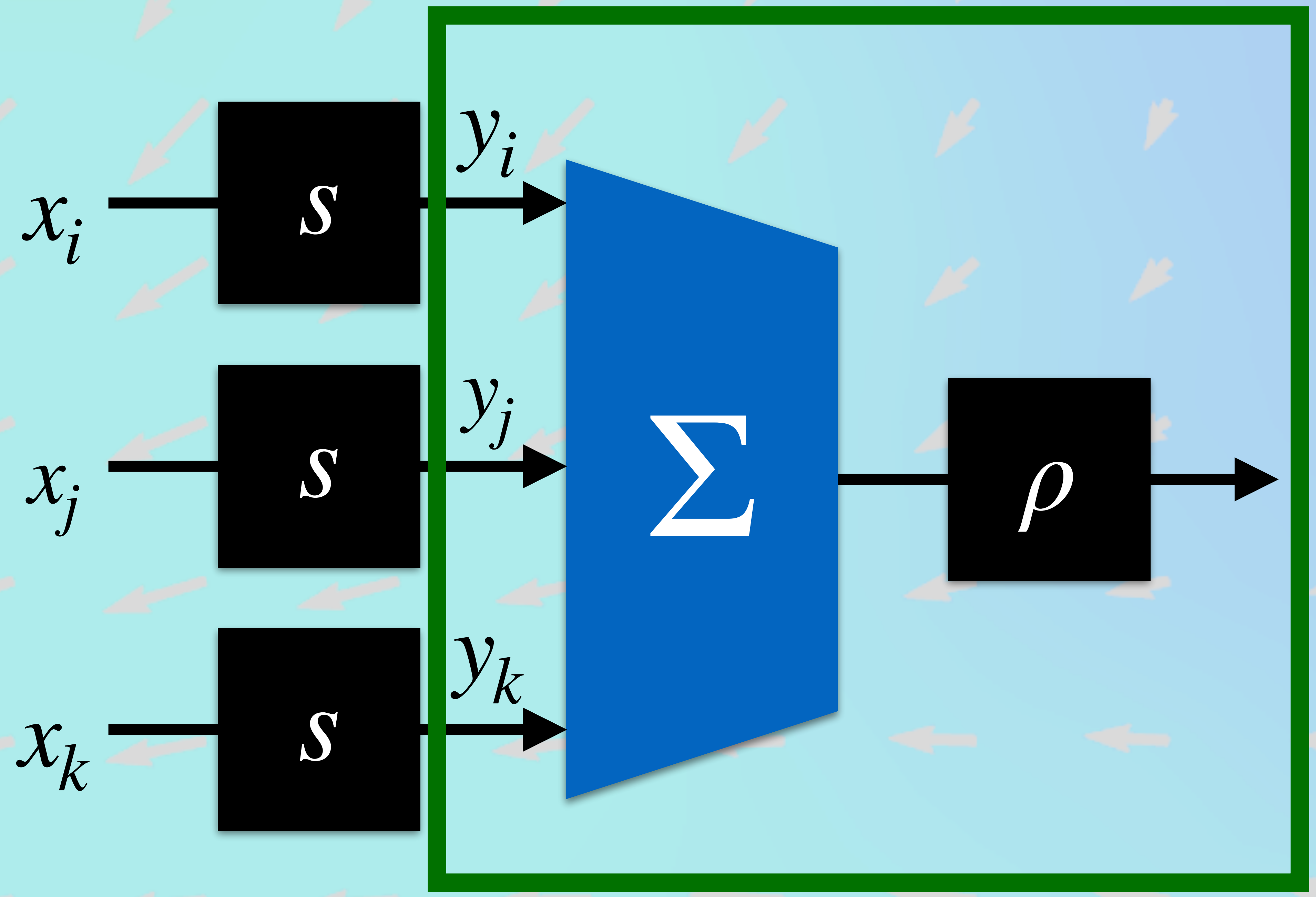


the *per-event embedding*

$$y \equiv s(x)$$

is information-preserving w.r.t. the posterior or likelihood over θ,

even in the presence of global NPs.



updating the posterior with new data is simple:

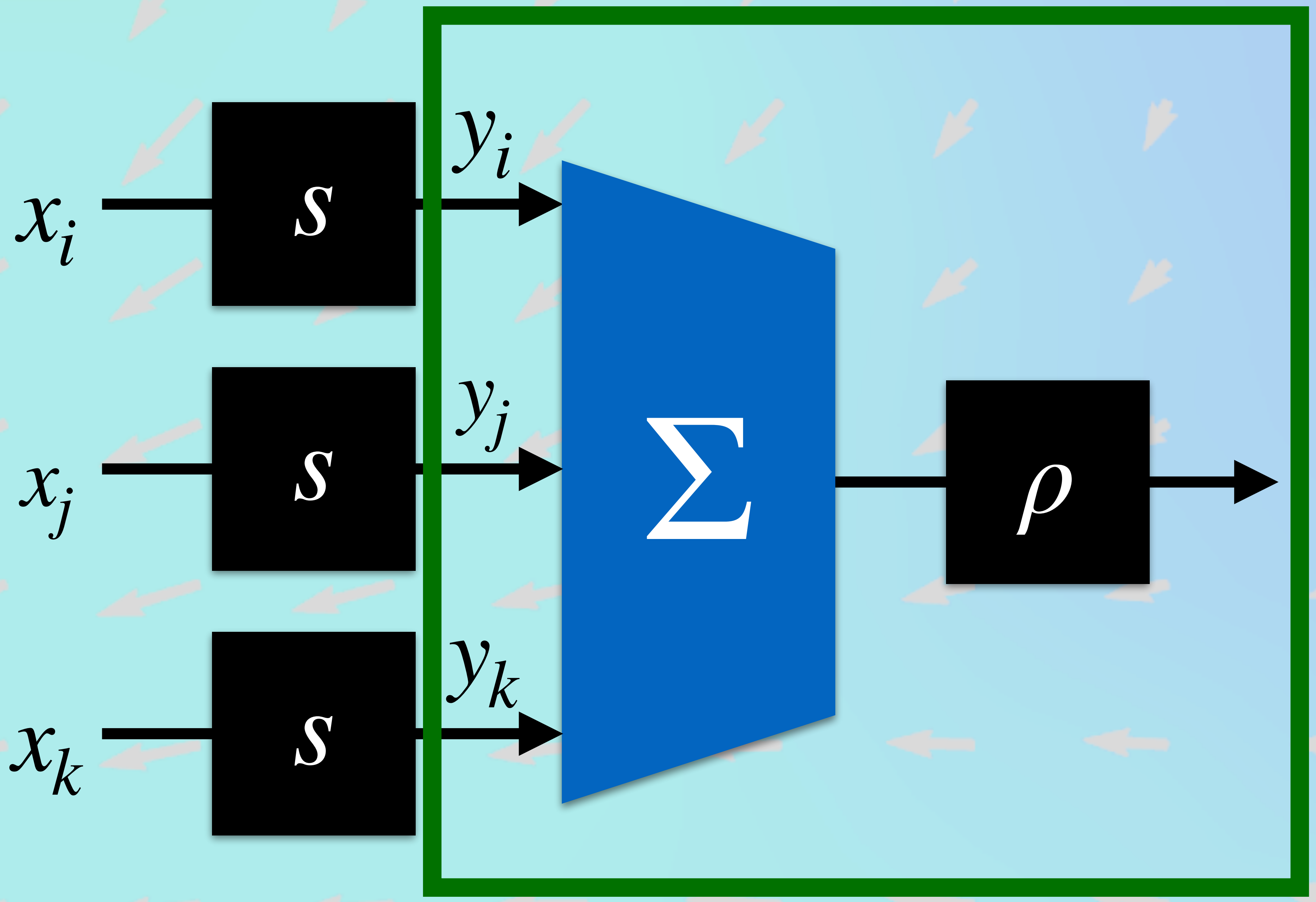
$$y_{\{x\}+x_0} = y_{\{x\}} + y_{x_0}$$

then feed this to ρ.

$\rightarrow$ this is extremely fast: can run full inference ~in real time.

“nSBI over ensembles”

https://arxiv.org/abs/2306.12584



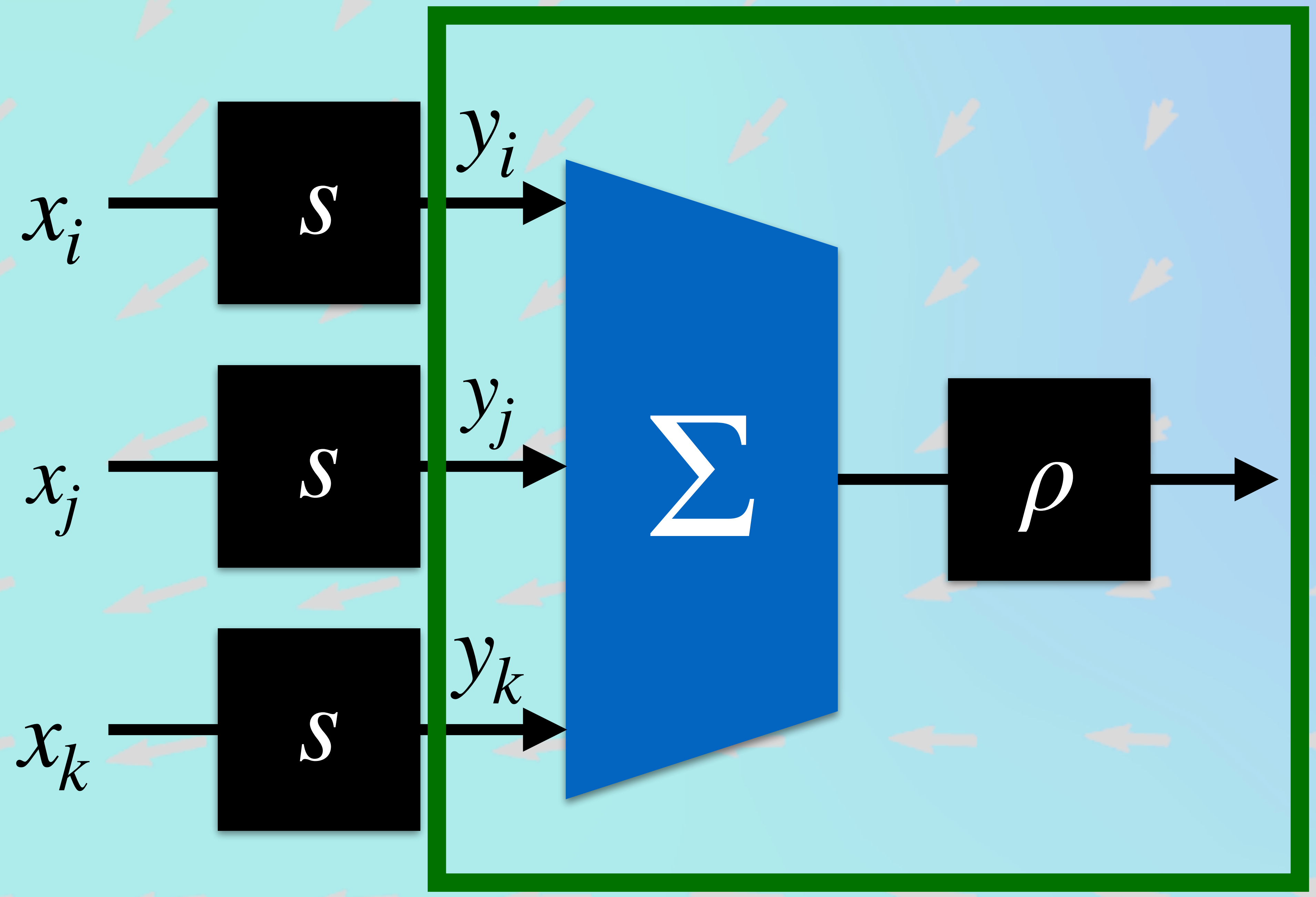
updating the posterior

data is simple:

$$y_{\{x\}+x_0} = y_{\{x\}} + y_{x_0}$$

then feed this to ρ.

→ this is extremely fast: can run full inference ~in real time.



“nSBI over ensembles”
https://arxiv.org/abs/2306.12584

updating the posterior w... data is simple:

$$y_{\{x\}+x_0} = y_{\{x\}} + y_{x_0}$$

then feed this to ρ.

$\rightarrow$ this is extremely fast: can run full inference ~in real time.

ongoing work: scaling up to datasets with millions of events +

https://animegenius.live3d.io/features/dragon-ai-art-generator

model mis-specification

model mis-specification

summary:

if we believe NNs work the way they should…

then given a simulator $p(x\,|\,\phi)$ **arbitrarily complex!**

we can learn an *arbitrarily precise approximation* of

the entire analysis is basically "done."

$$\log \frac{p(x\,|\,\phi)}{p(x\,|\,\hat{\phi})} \quad \text{or} \quad p(\phi\,|\,x).$$

leads to *best* possible constraints

we claimed for

"arbitrarily complex x and ϕ"

given a simulator $p(x\,|\,\phi)$

we can perform "correct" inference.

model mis-specification

summary:

if we believe NNs work the way they should…

then given a simulator $p(x\,|\,\phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

the entire analysis is basically "done."

$$\log \frac{p(x\,|\,\phi)}{p(x\,|\,\hat{\phi})} \quad \text{or} \quad p(\phi\,|\,x).$$

leads to *best* possible constraints

is the simulation *arbitrarily trustworthy*?

model mis-specification

summary:

if we believe NNs work the way they should...

then given a simulator $p(x\,|\,\phi)$ **arbitrarily complex!**

we can learn an *arbitrarily precise approximation* of

the entire analysis is basically "done."

$$\log \frac{p(x\,|\,\phi)}{p(x\,|\,\hat{\phi})} \quad \text{or} \quad p(\phi\,|\,x).$$

leads to *best* possible constraints

is the simulation
arbitrarily trustworthy?

of course the answer is "no"!

how do we validate/calibrate
very complex observables x
that enable strong,
correct constraints?

model mis-specification

summary:

if we believe NNs work the way they should…

then given a simulator $p(x \mid \phi)$

arbitrarily complex!

we can learn an *arbitrarily precise approximation* of

the entire analysis is

$\log \frac{p(x \mid \phi)}{p(x \mid \hat{\phi})}$ or $p(\phi \mid x)$.

leads to *best* possible

in general a very difficult problem, but we are working on it…

is the simulation *arbitrarily trustworthy*?

of course the answer is "no"!

how do we validate/calibrate very complex observables x that enable strong, ***correct*** constraints?

we have a recipe for reweighting a density $p(x)$ (sim) to $q(x)$ (data):

- train a discriminator $D_p^q(x)$ and weight each sample x by $\exp D_p^q(x)$.
- allows weight-based calibration for complex observables.

we have a recipe for reweighting a density $p(x)$ (sim) to $q(x)$ (data):

- train a discriminator $D_p^q(x)$ and weight each sample x by $\exp D_p^q(x)$.
- allows weight-based calibration for complex observables.

in many instances it is preferable to *move* events to perform a calibration:

$$T : \vec{x} \to \vec{x} \implies T_{\#}p \approx q$$

- for any p and q, at least one T exists, and it is unitary.
- we usually want to change the simulation *as little as possible*:
  - ✦ we need to find the unique $\hat{T}$ that minimally (or "optimally") morphs p into q.

we have been implicitly using optimal transport (OT) for decades in HEP:

◉ correct simulated scale of a gaussian density ↔ move the mean of simulated distribution.

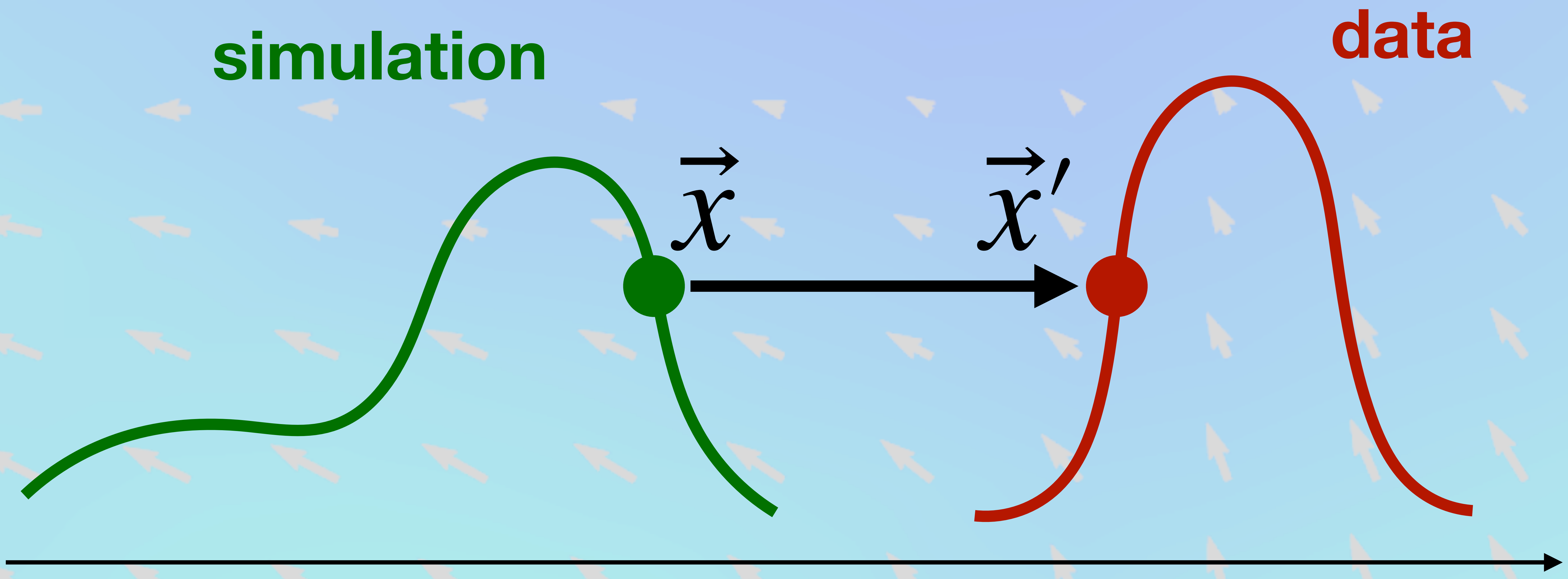
neural OT generalizes this process.

◉ for euclidean spaces, the OT map is the gradient of some convex potential:

$$\hat{T}\vec{x} \equiv \vec{\nabla}\phi(\vec{x}).$$

◉ we have recently managed to learn ϕ_z and therefore $\hat{T}_z$ in concrete use-cases:

✦ the OT map is *conditional* on z.



we have been implicitly using optimal transport (OT) fo … EP:

◉ correct simulated scale of a gaussian density ↔ move the mean … tribution.

optimal transport for calibrations
https://arxiv.org/abs/2107.08648

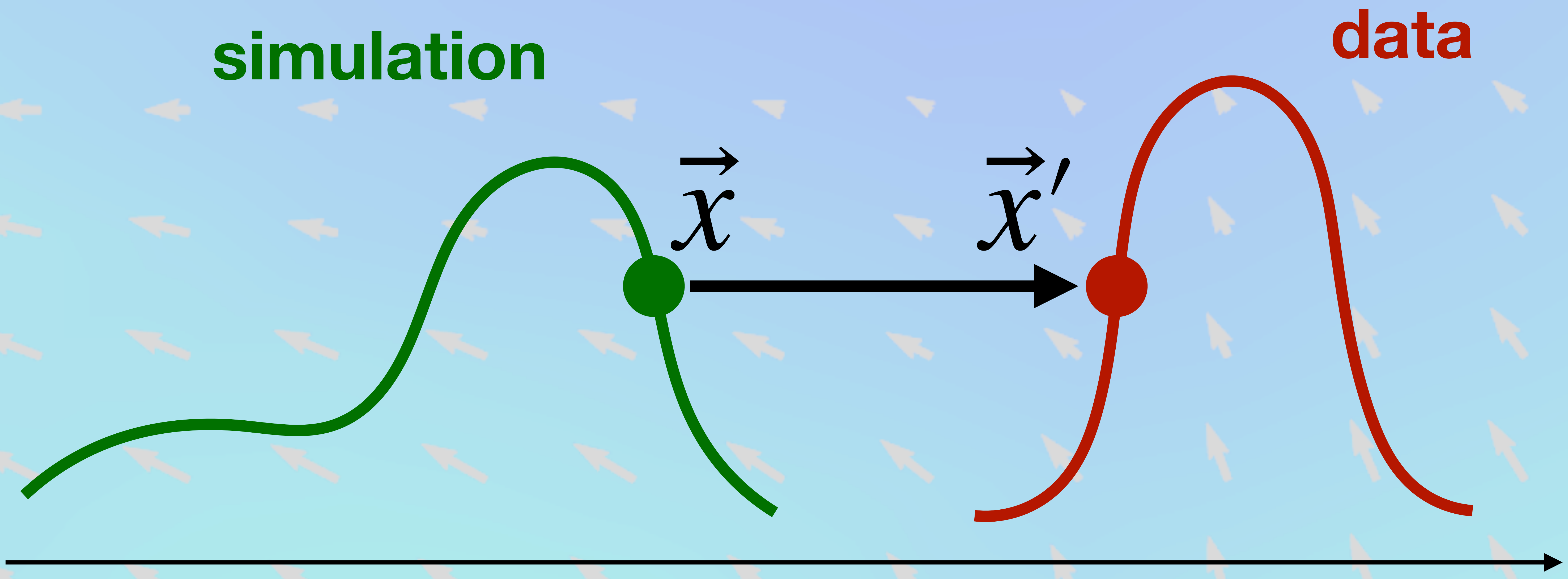
neural OT generalizes this process.

◉ for euclidean spaces, the OT map is the gradient of some convex potential:

$$\hat{T}\,\vec{x} \equiv \vec{\nabla}\,\phi(\vec{x}).$$

◉ we have recently managed to learn ϕ_z and therefore $\hat{T}_z$ in concrete use-cases:

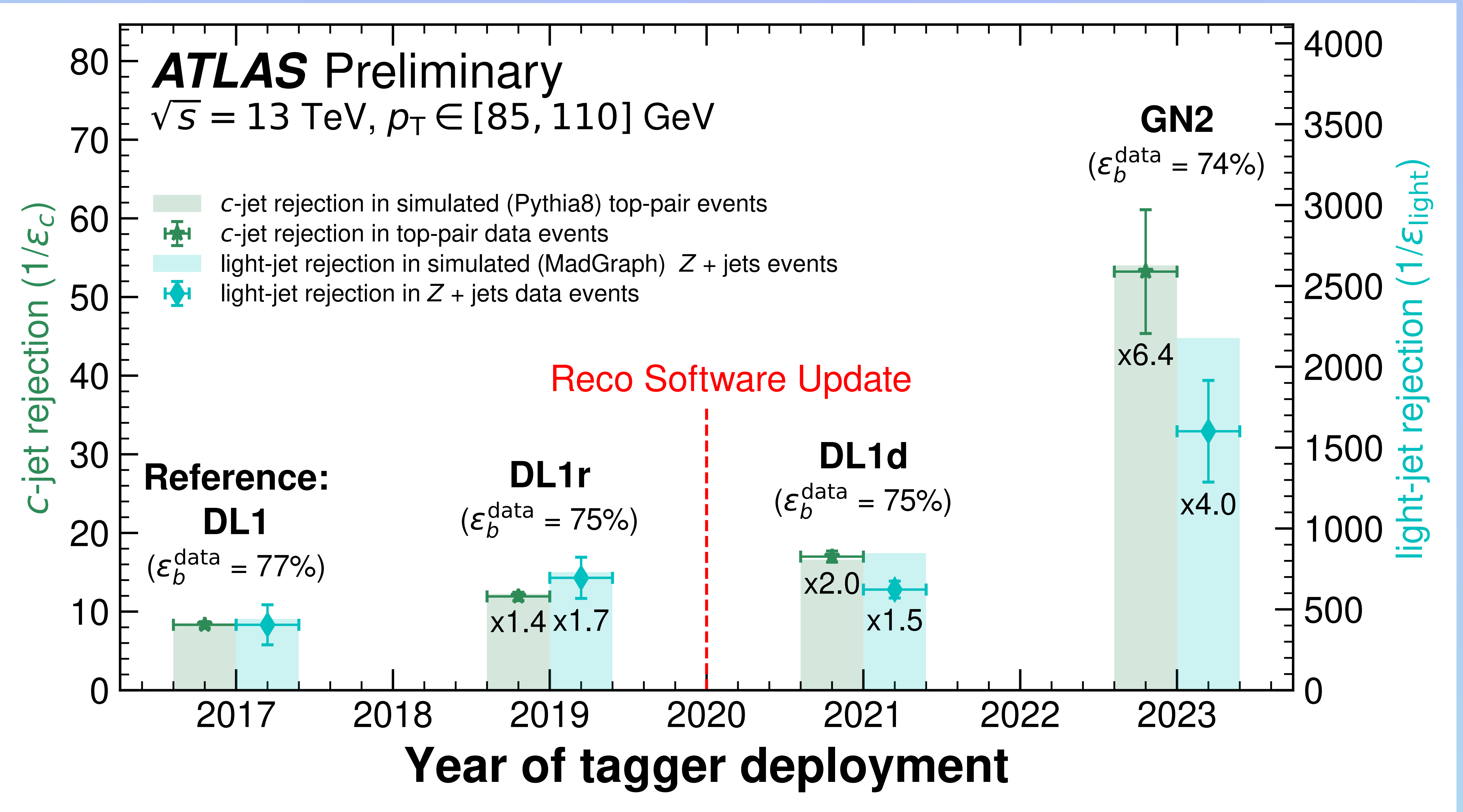
✦ the OT map is *conditional* on z.



a concrete example: jet flavor-tagging in ATLAS

jet flavor-tagging is a classification problem:

- ATLAS's classifiers emit the probability of a jet to contain a b-hadron, c-hadron, or neither (p_b, p_c, p_u).
- modern algorithms are very complex:
  - ✦ charged-particle tracks as inputs.
  - ✦ incredible separating power, but clear mismodeling
- until now, we had no direct calibration for these probabilities.
  - ✦ to do so, we set $q_i \equiv \operatorname{logit} p_i$, treat $\vec{q}$ as euclidean, and calibrate via OT.



notation:

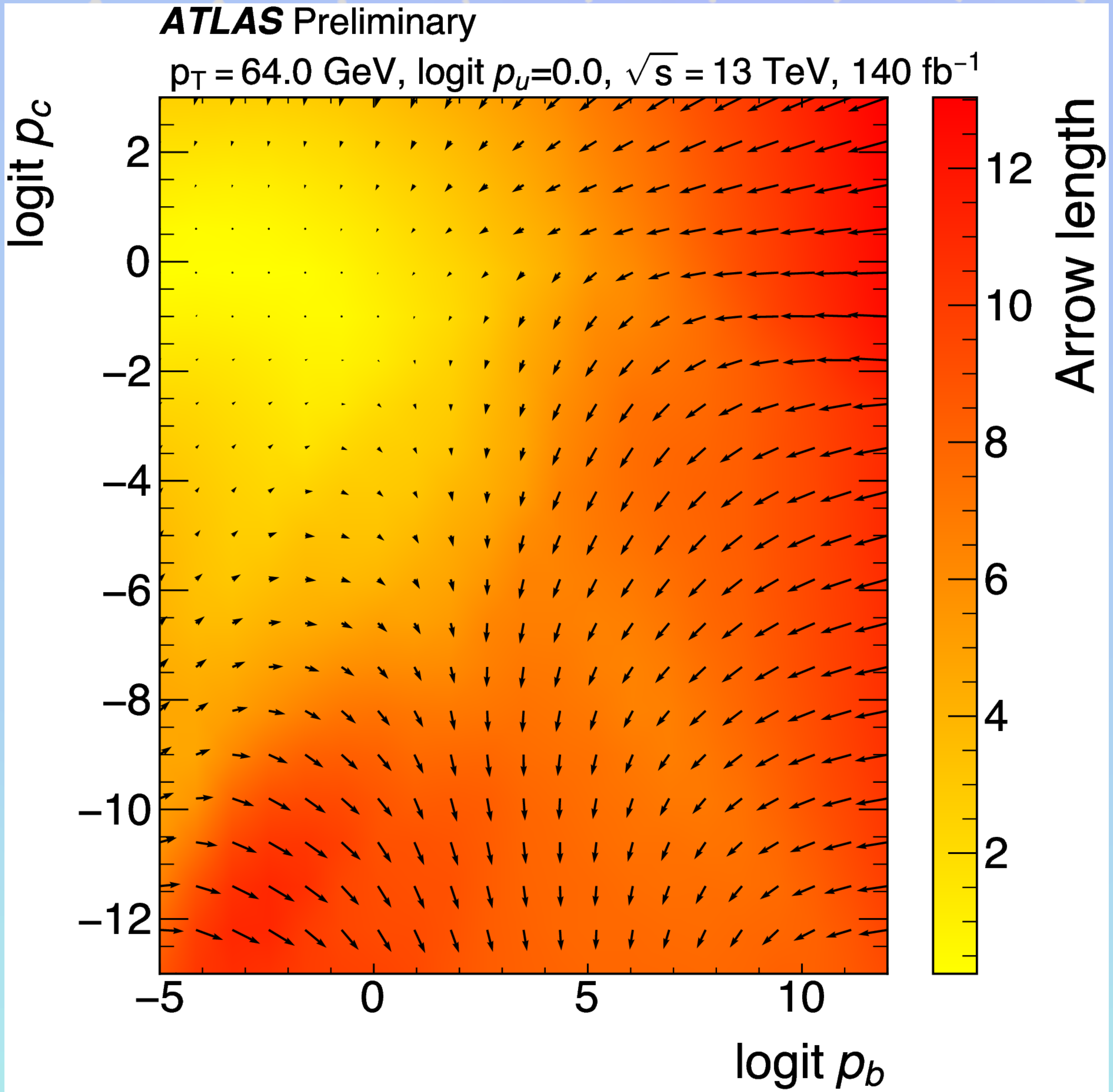
- $q_i \equiv \text{logit}\, p_i$: flav. class. scores
- $p_{\text{sim}}(\vec{q}\,|\,p_T) \equiv p_{\text{sim}}(\vec{q}\,|\,p_T)\, p_{\text{data}}(p_T)$
- $\hat{T}_\# \equiv p_T$-dependent OT map

result:

◉ we obtain the full 3D OT maps in $\vec{q}$ space s.t. $\hat{T}_\# p_{\text{sim}} \approx p_{\text{data}}$,

✦ derived as a function of jet p_T.

◉ here we show a 2D slice for $q_b \times q_c$ at fixed $q_u \times p_T$.



notation:

- $q_i \equiv \operatorname{logit} p_i$: flav. class. scores
- $p_{\rm sim}(\vec{q}\,|\,p_T) \equiv p_{\rm sim}(\vec{q}\,|\,p_T)\,p_{\rm data}(p_T)$
- $\hat{T}_\# \equiv p_T$-dependent OT map

the technology works!

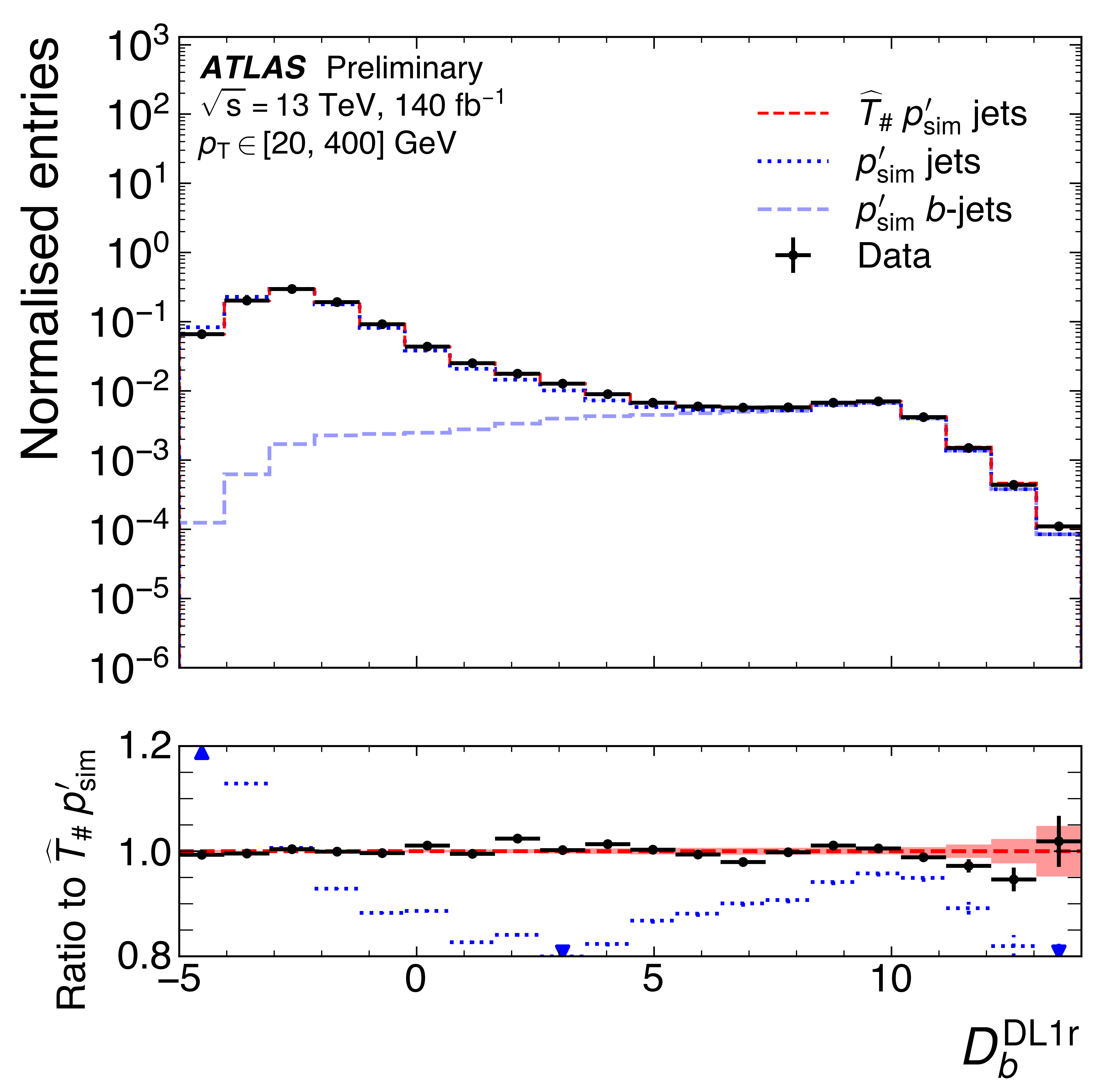
◉ before calibration, poor modeling of many quantities…

✦ including the b-tagging discriminator

$$D_b \equiv \log \frac{p_b}{f_c p_c + (1-f_c) p_u}$$

◉ after calibration, very good agreement even for this non-trivial quantity:

✦ higher-order correlations between p_i are properly corrected.



notation:

- $q_i \equiv \operatorname{logit} p_i$: flav. class. scores
- $p_{\rm sim}(\vec{q}\,|\,p_T) \equiv p_{\rm sim}(\vec{q}\,|\,p_T)\,p_{\rm data}(p_T)$
- $\hat{T}_\# \equiv p_T$-dependent OT map

the technology works!

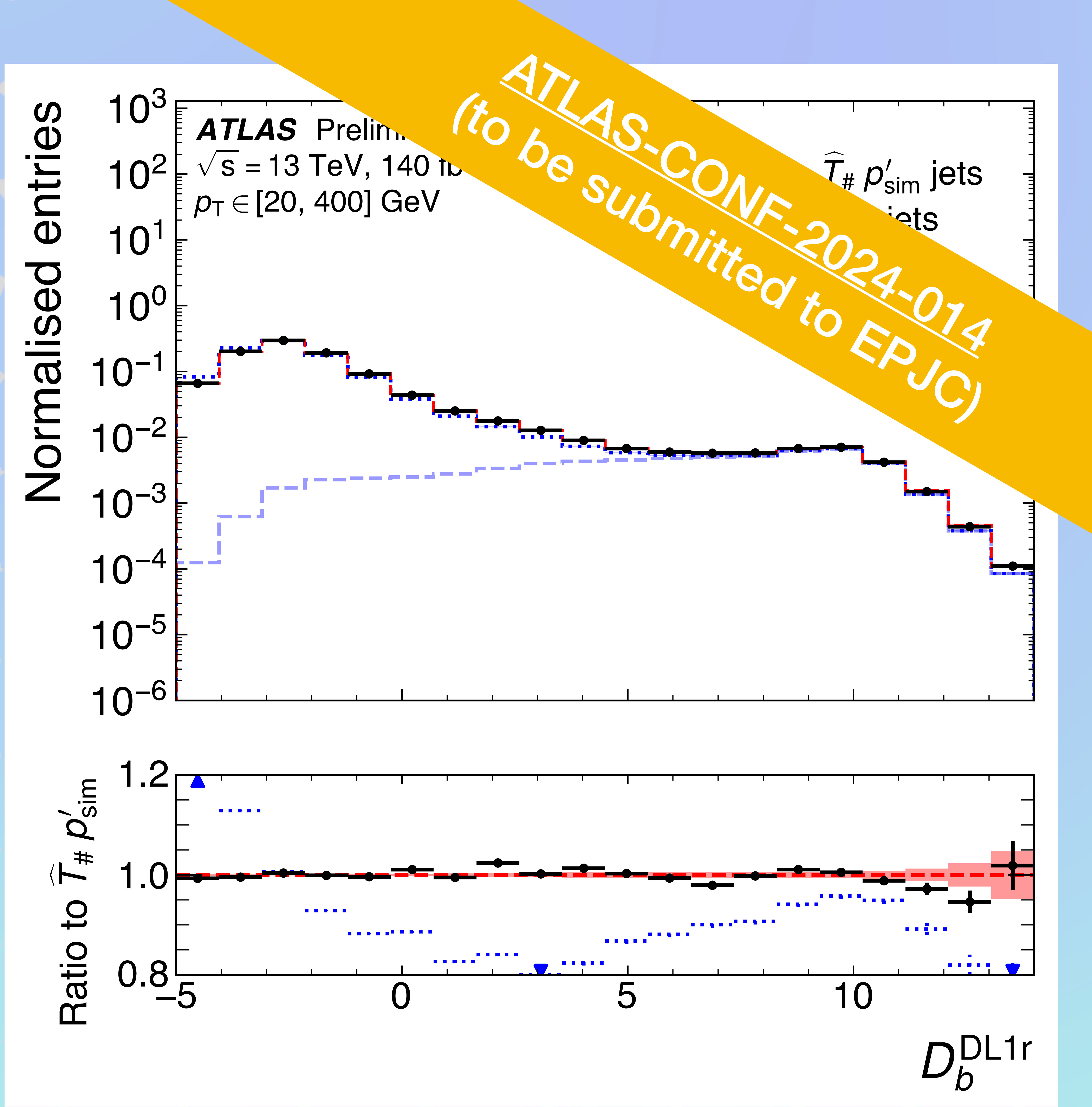
◉ before calibration, poor modeling of many quantities…

✦ including the b-tagging discriminator

$$D_b \equiv \log \frac{p_b}{f_c p_c + (1 - f_c) p_u}$$

◉ after calibration, very good agreement even for this non-trivial quantity:

✦ higher-order correlations between p_i are properly corrected.



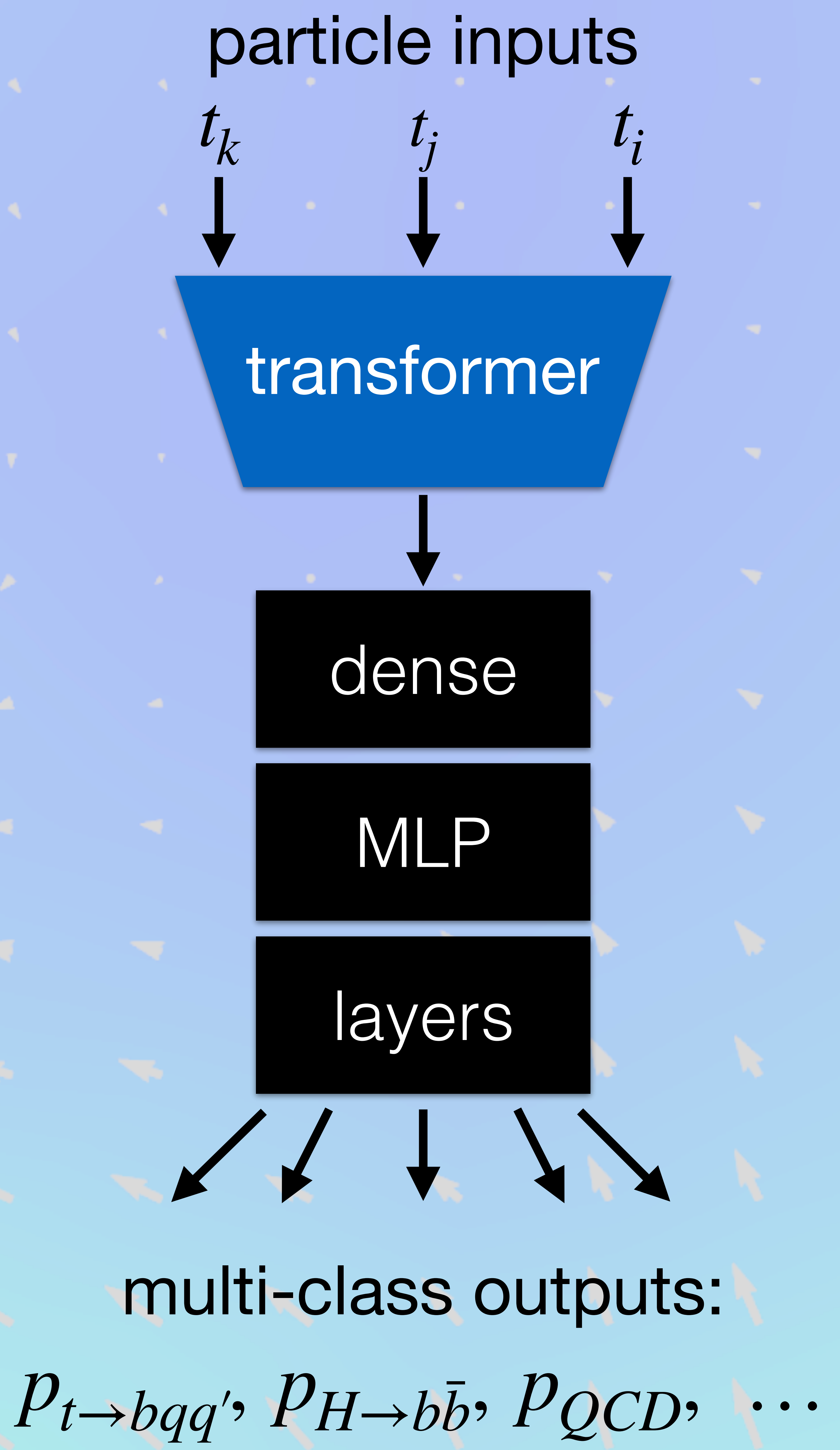
the future (?): scaling this up

ongoing work: scaling it up

- preliminary results show that this scales nicely to *very* high dimensions:

- start with a JetClass-like classifier that discriminates between 10 distinct types of large-radius jets:

    - ✦ $H \to bb, H \to cc, t \to bqq'$, inclusive ("QCD") jets, etc

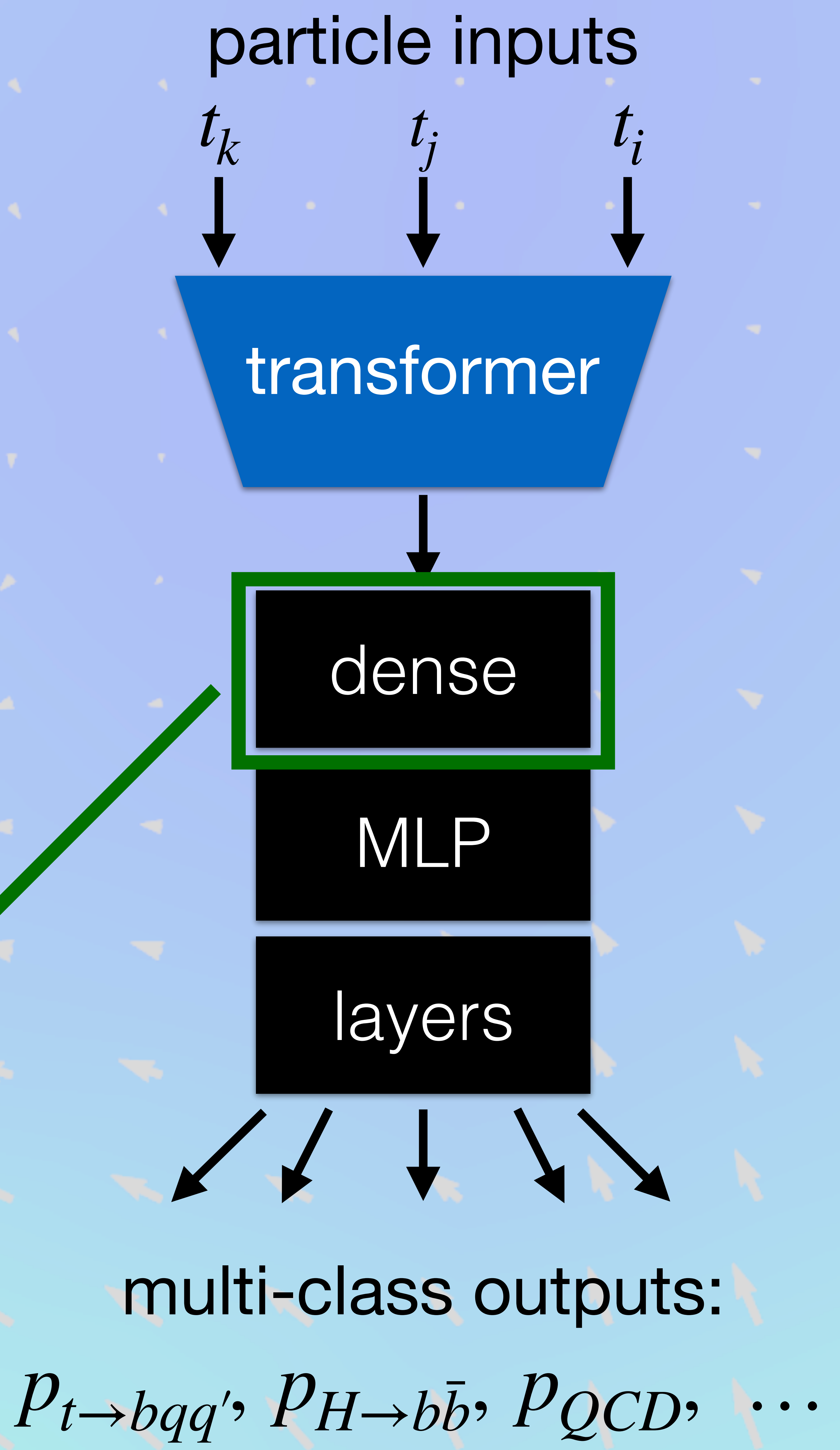


ongoing work: scaling it up

- preliminary results show that this scales nicely to *very* high dimensions:

- start with a JetClass-like classifier that discriminates between 10 distinct types of large-radius jets:
  - $H \to bb, H \to cc, t \to bqq',$ inclusive (“QCD”) jets, etc

- *calibrate the internal 128-dim representation of the jet information*

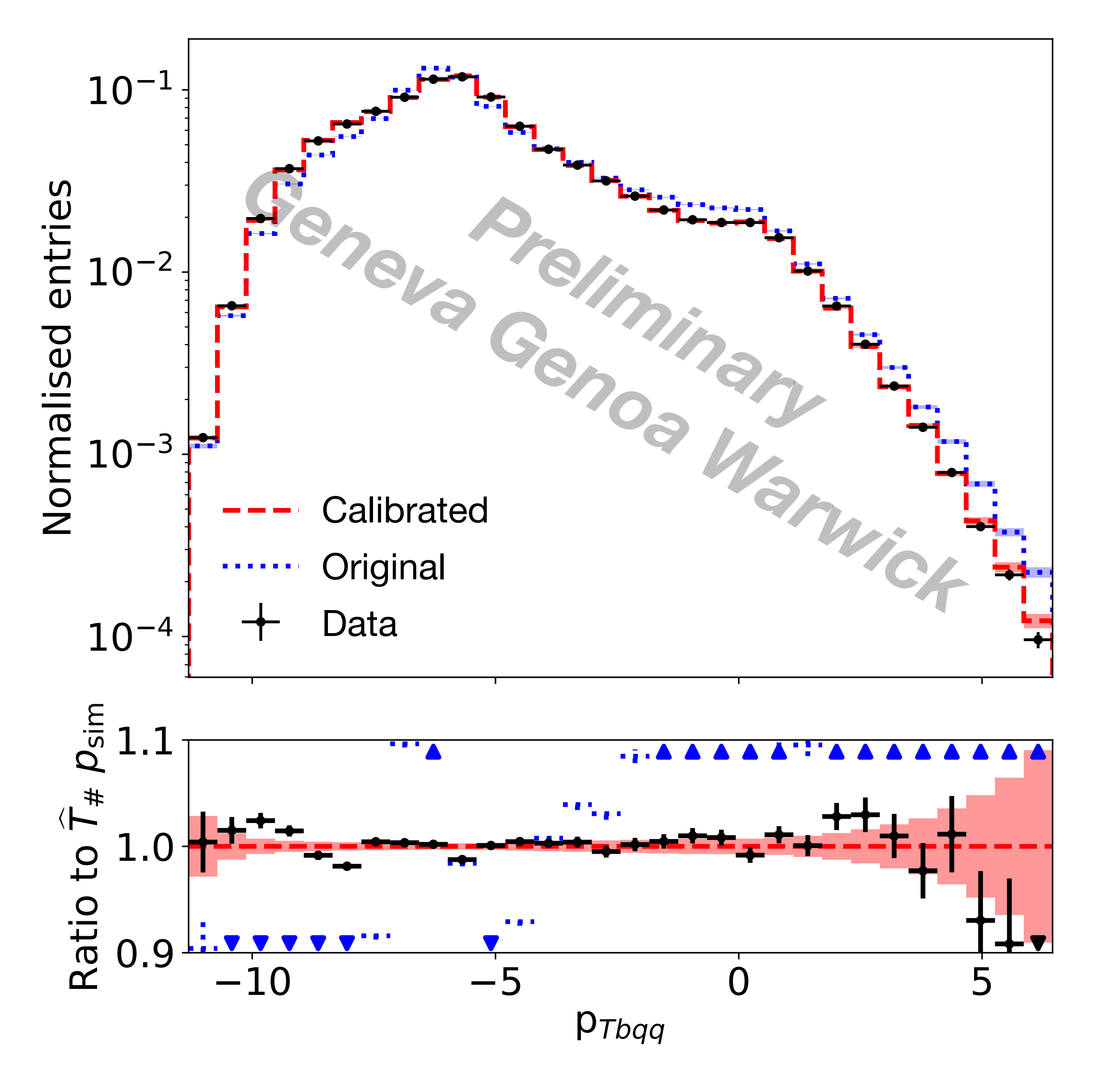


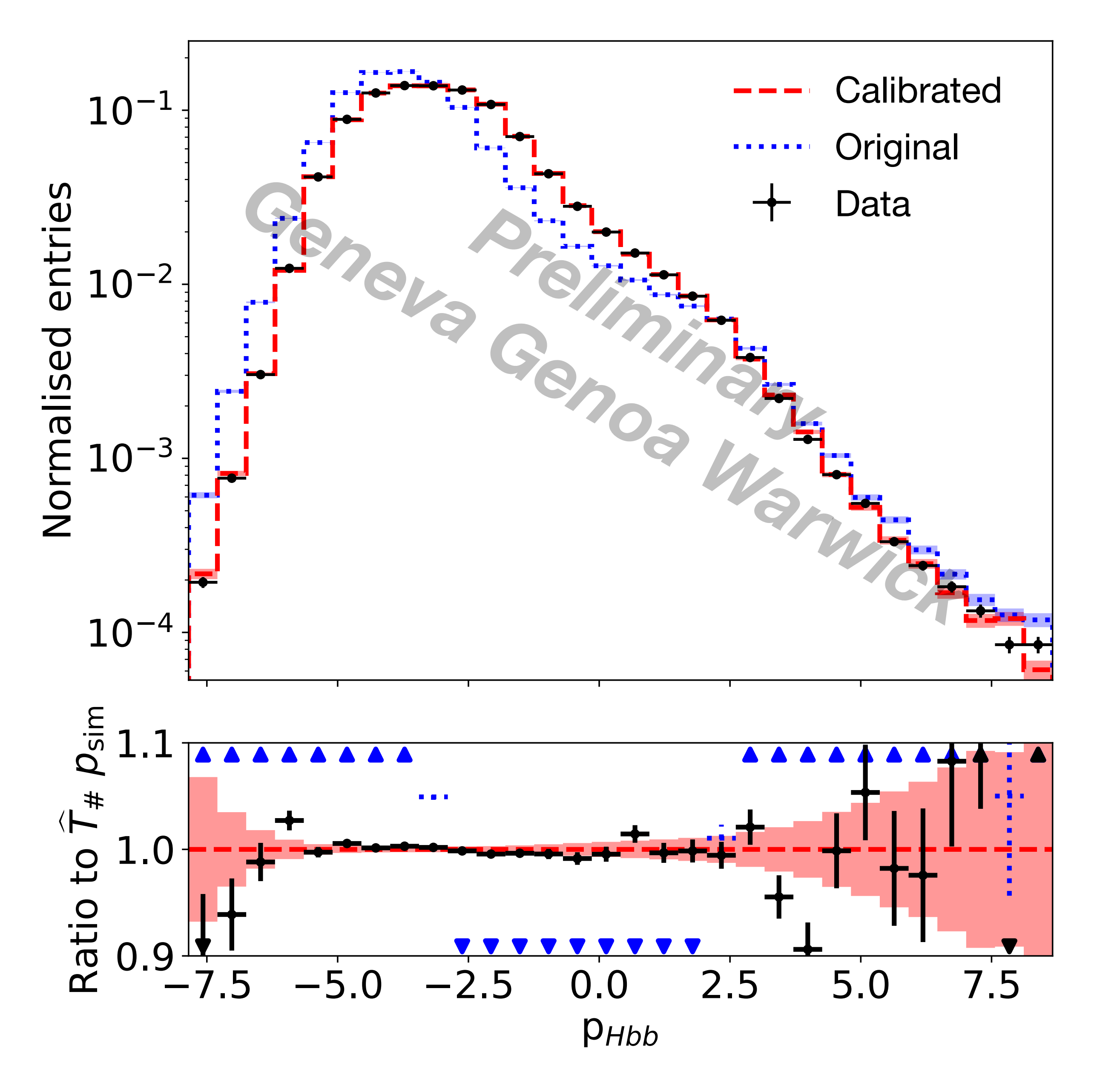
ongoing work: scaling this up

- with the 128-dim “latent representation” of the jet calibrated,
    - we observe that the original 10 classification scores close very well.

ongoing work: scaling this up

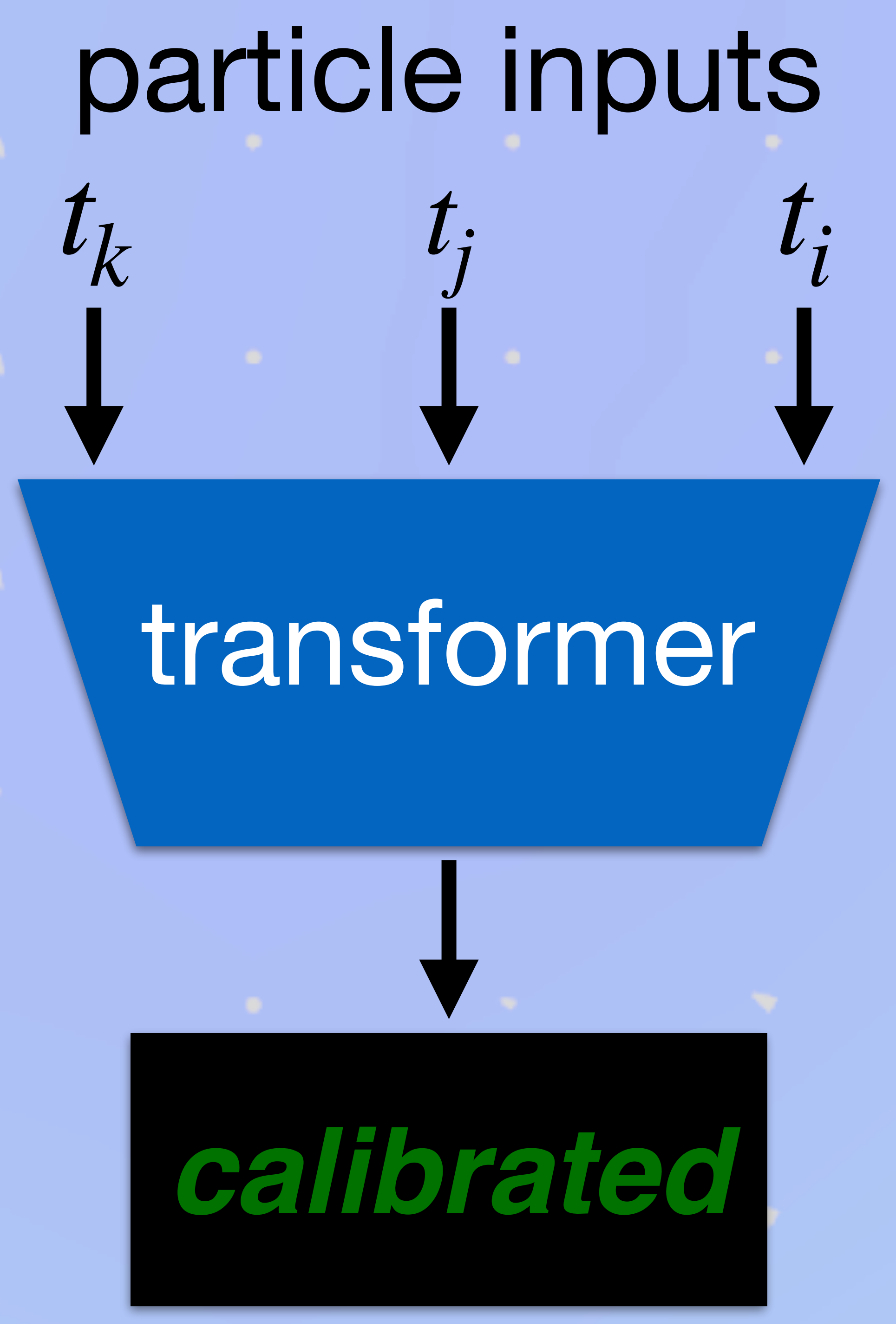
- with the 128-dim “latent representation” of the jet calibrated,
  - ✦ we observe that the original 10 classification scores close very well.





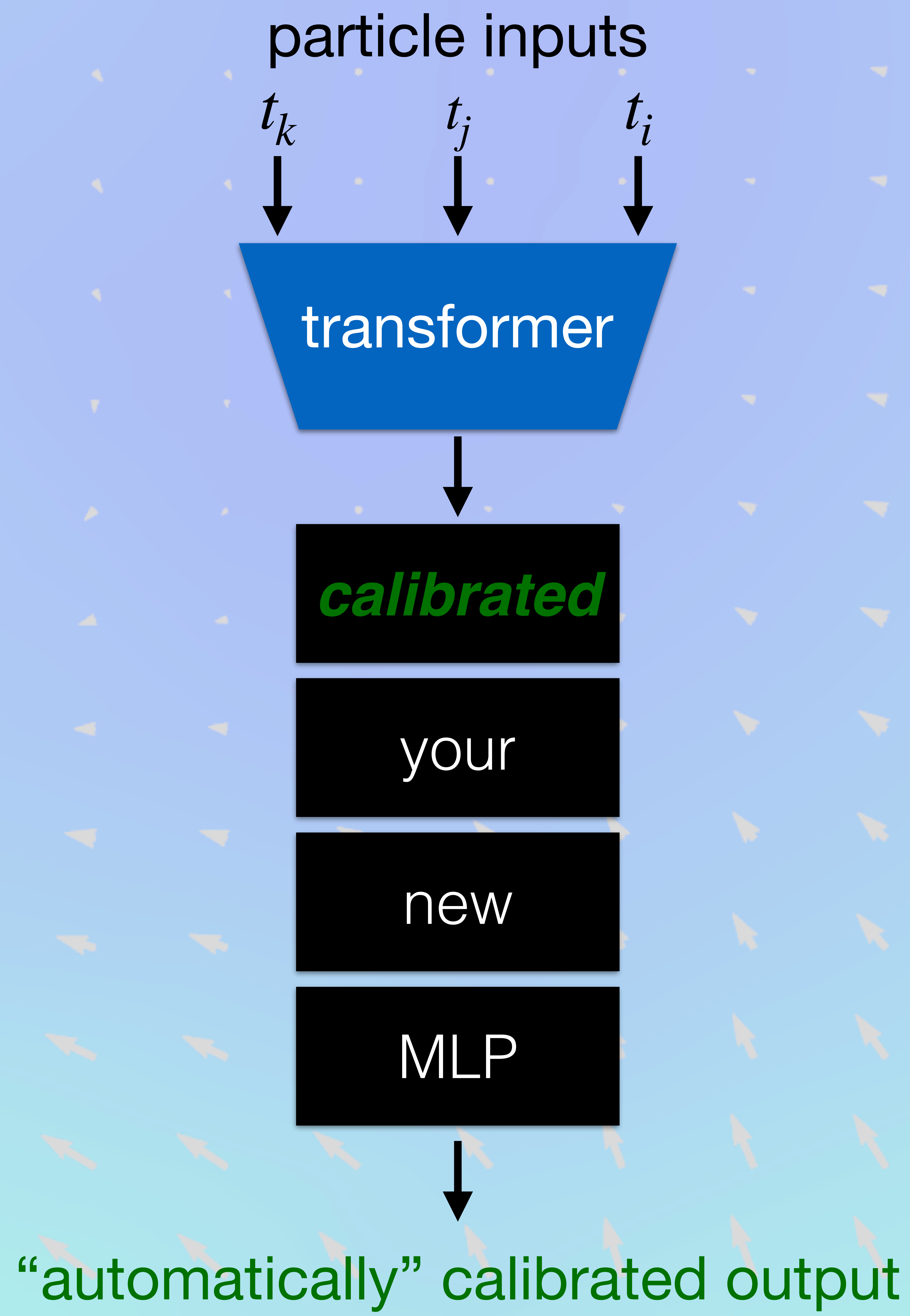
ongoing work: scaling this up

- with the 128-dim "latent representation" of the jet calibrated,
  - ✦ we observe that the original 10 classification scores close very well.
- but this enables "arbitrary" use of the information contained in that representation:



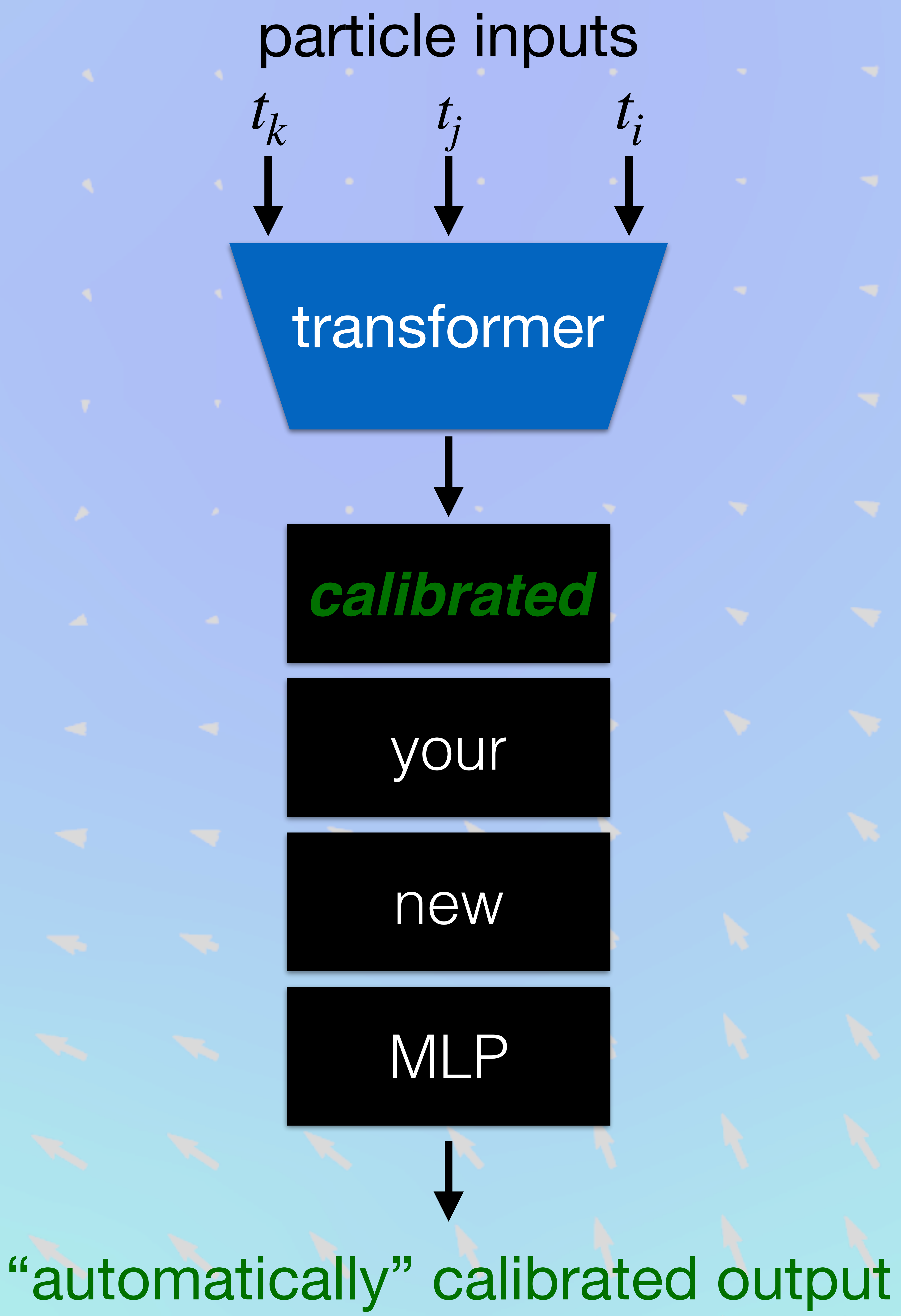
ongoing work: scaling this up

- with the 128-dim "latent representation" of the jet calibrated,
  - ✦ we observe that the original 10 classification scores close very well.
- but this enables "arbitrary" use of the information contained in that representation:



ongoing work: scaling this up

- with the 128-dim "latent representation" of the jet calibrated,
  - ✦ we observe that the original 10 classification scores close very well.
- but this enables "arbitrary" use of the information contained in that representation:
  - ✦ allows calibration of "foundation" models for broad use.
  - ✦ useful well beyond particle physics!



nSBI promises to yield the best possible constraints on a parameter ϕ given an observable x.

- that's a ***big*** claim but also worth pursuing…

- *all* the information of x relevant to ϕ in the simulator $p(x \mid \phi)$ can be used for inference:

  - allows more complex x and ϕ than standard approaches.

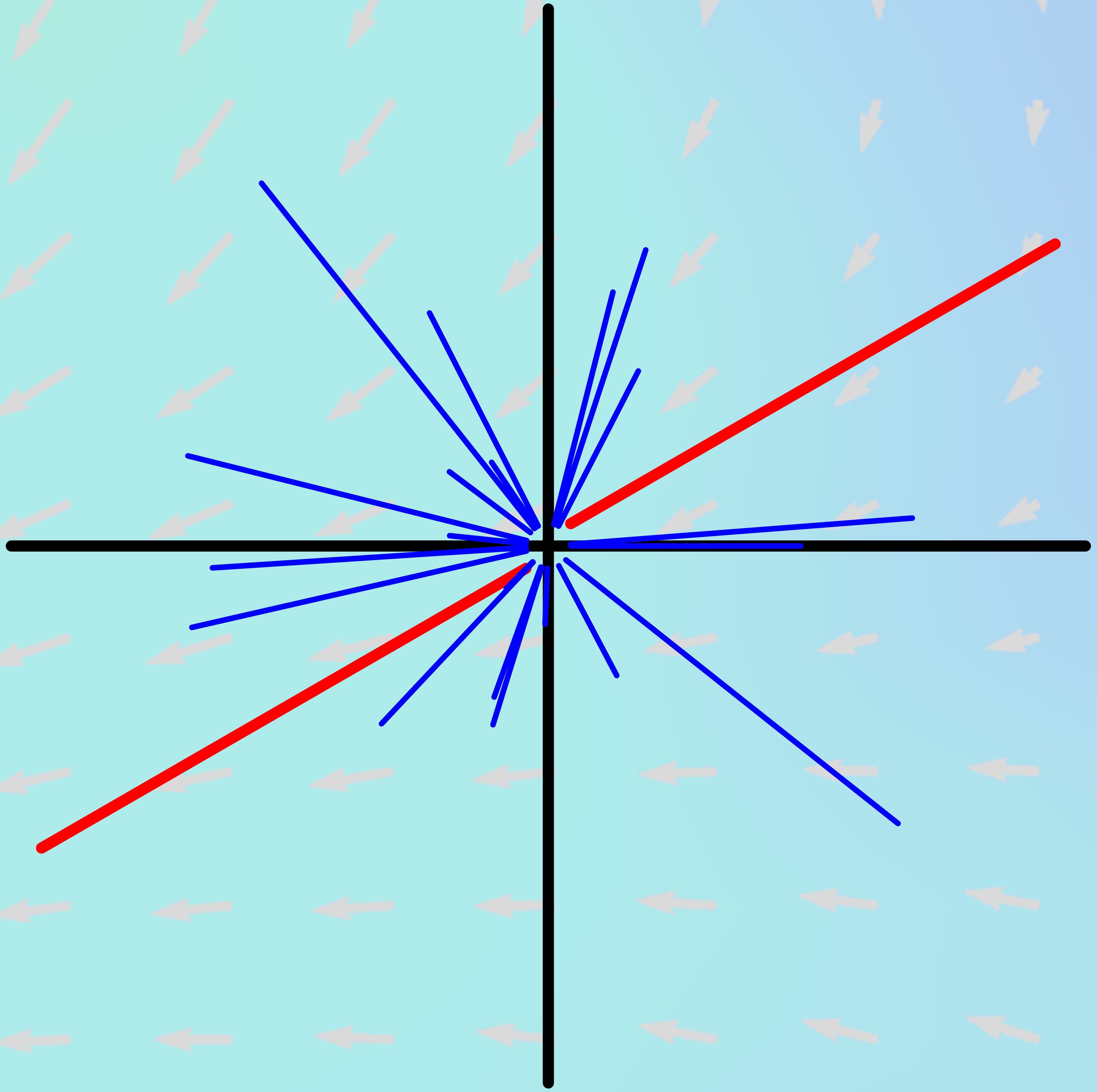
  - potential huge gains in cases where NPs are very correlated with PoIs.

  - **requires very good (correct!) simulation.**

nSBI promises to yield the best possible constraints on a parameter ϕ given an observable x.

- ◉ lots of ongoing work to make this a reality:
  - ✦ to contend with "technical" problems like convergence and global NPs,
  - ✦ to calibrate complicated observables x to enable ***correct*** constraints.
- ◉ large potential speed-up in inference time.
- ◉ I'm hopeful that in $\mathcal{O}(5)$ years there will be "over the counter" solutions available — but it will take effort to get there.
  - ✦ It's a very interesting space with lots of work to do!


thank you!



why do we need summaries / embeddings / "physics objects"?

😱 **tractability**: ~millions of detector channels to read out per LHC bunch crossing.

😱 **correctness**: difficult to construct a simulator that *adequately describes all details* of the data!

indeed, this is perhaps the main "point" of constructing jets:

- we cannot correctly predict the details of QCD with arbitrary accuracy;

- we *can* predict the "large-scale structure" of the fragmentation of partons.

"large-scale structure": calculable features

calibration region

analysis region

even so, having access *more relevant details* can enable better constraints.

- we can often *measure* the density over some features better than we can predict it.
- a *calibration region* may be used to measure said density and to *correct* the simulation.

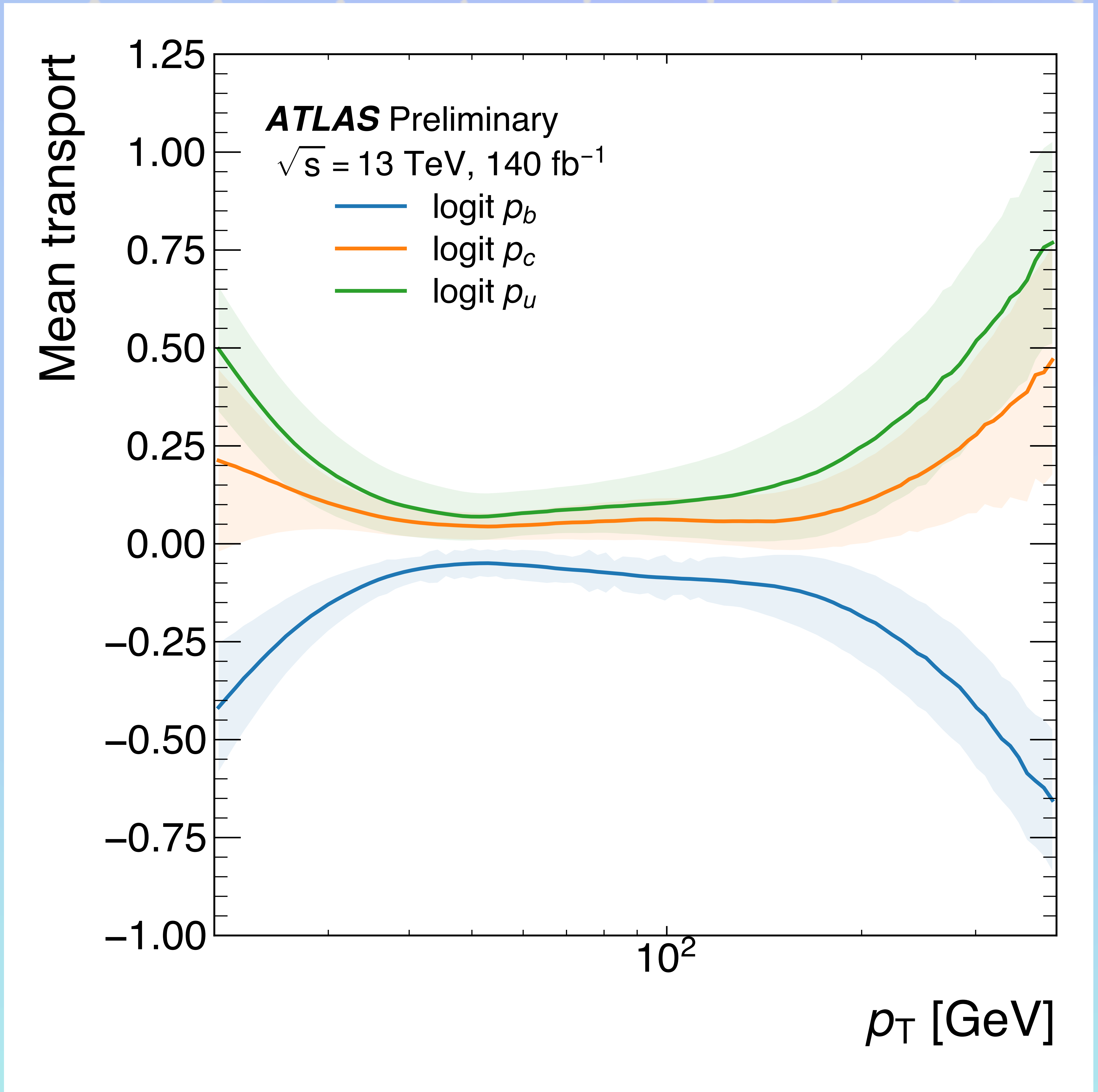
notation:

- $q_i \equiv \text{logit}\, p_i$: flav. class. scores
- $p_{\text{sim}}(\vec{q}\,|\,p_T) \equiv p_{\text{sim}}(\vec{q}\,|\,p_T)\, p_{\text{data}}(p_T)$
- $\hat{T}_{\#} \equiv p_T$-dependent OT map

◉ for b-jets: $\hat{T}\, p_b < p_b$, while the reverse is true for p_c and p_u.

✦ the simulation overstates its classification power.

◉ we have much more information about what aspects of the simulation are incorrect than before.



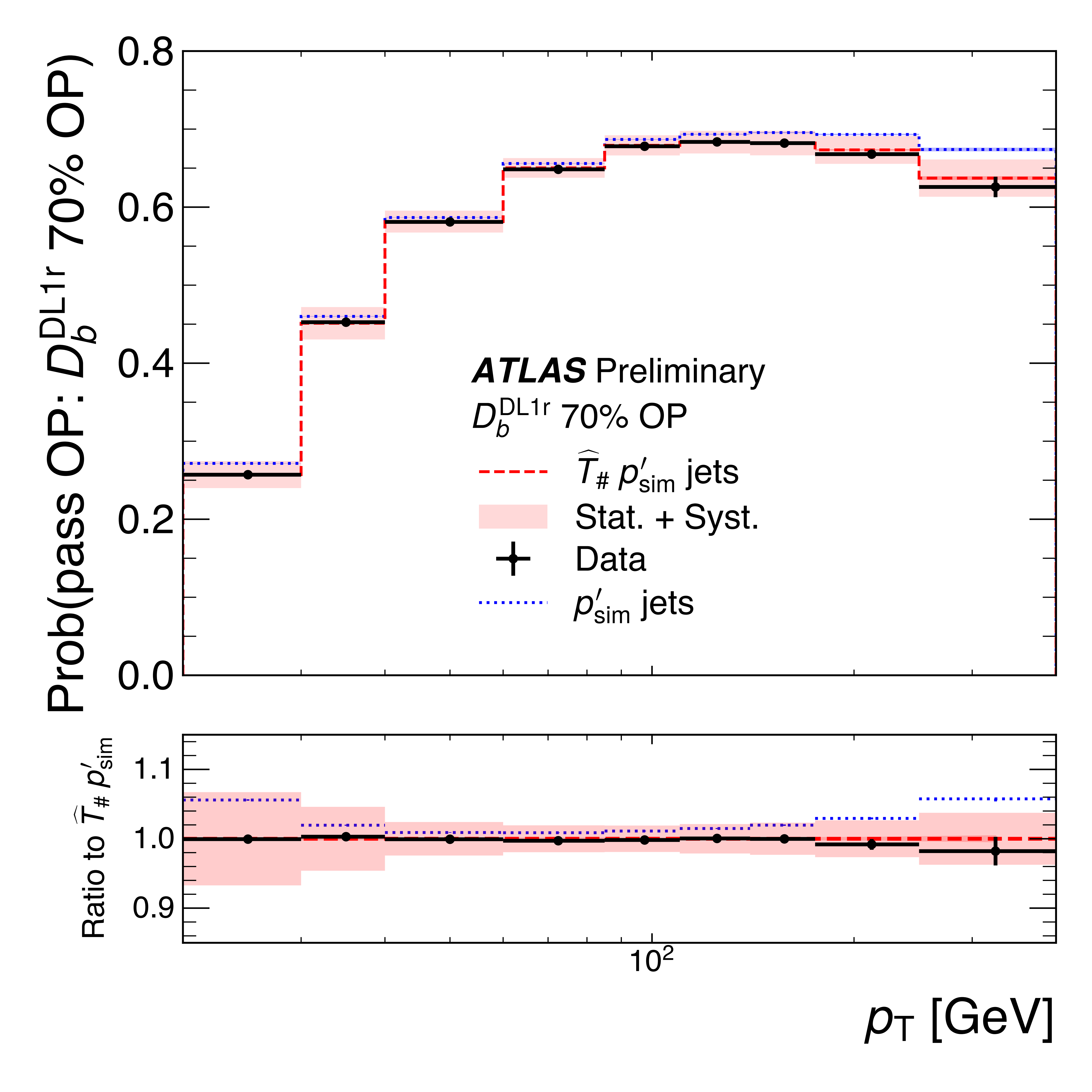
notation:

- $q_i \equiv \operatorname{logit} p_i$: flav. class. scores
- $p_{\rm sim}(\vec{q}\,|\,p_T) \equiv p_{\rm sim}(\vec{q}\,|\,p_T)\,p_{\rm data}(p_T)$
- $\hat{T}_\# \equiv p_T$-dependent OT map

conventional operating points are "automatically" corrected.

◉ excellent closure observed for the "standard" b-tagging discriminant points

✦ used to observe Higgs couplings to W, b, t.



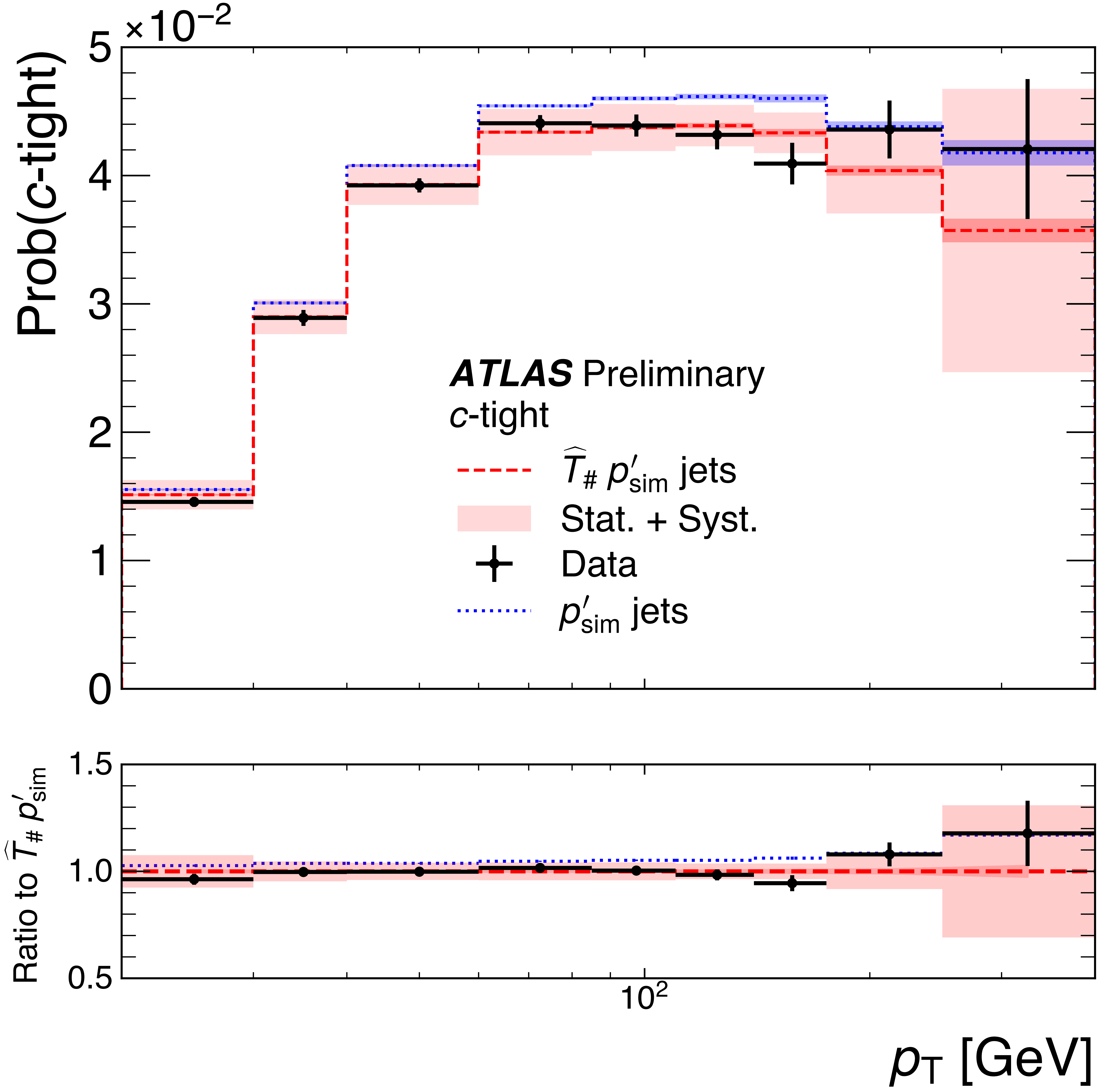
notation:

- $q_i \equiv \operatorname{logit} p_i$: flav. class. scores
- $p_{\text{sim}}(\vec{q}\,|\,p_T) \equiv p_{\text{sim}}(\vec{q}\,|\,p_T)\, p_{\text{data}}(p_T)$
- $\hat{T}_{\#} \equiv p_T$-dependent OT map

more general uses of flavor-tagging become possible.

◉ charm-tagging discriminator used to constrain the $H \leftrightarrow c$ couplings shows good agreement.

◉ closure depends on the full 3D density $\hat{T}_{\#}\, p_{\text{sim}}(\vec{q}\,|\,p_T)$ agreeing with data.



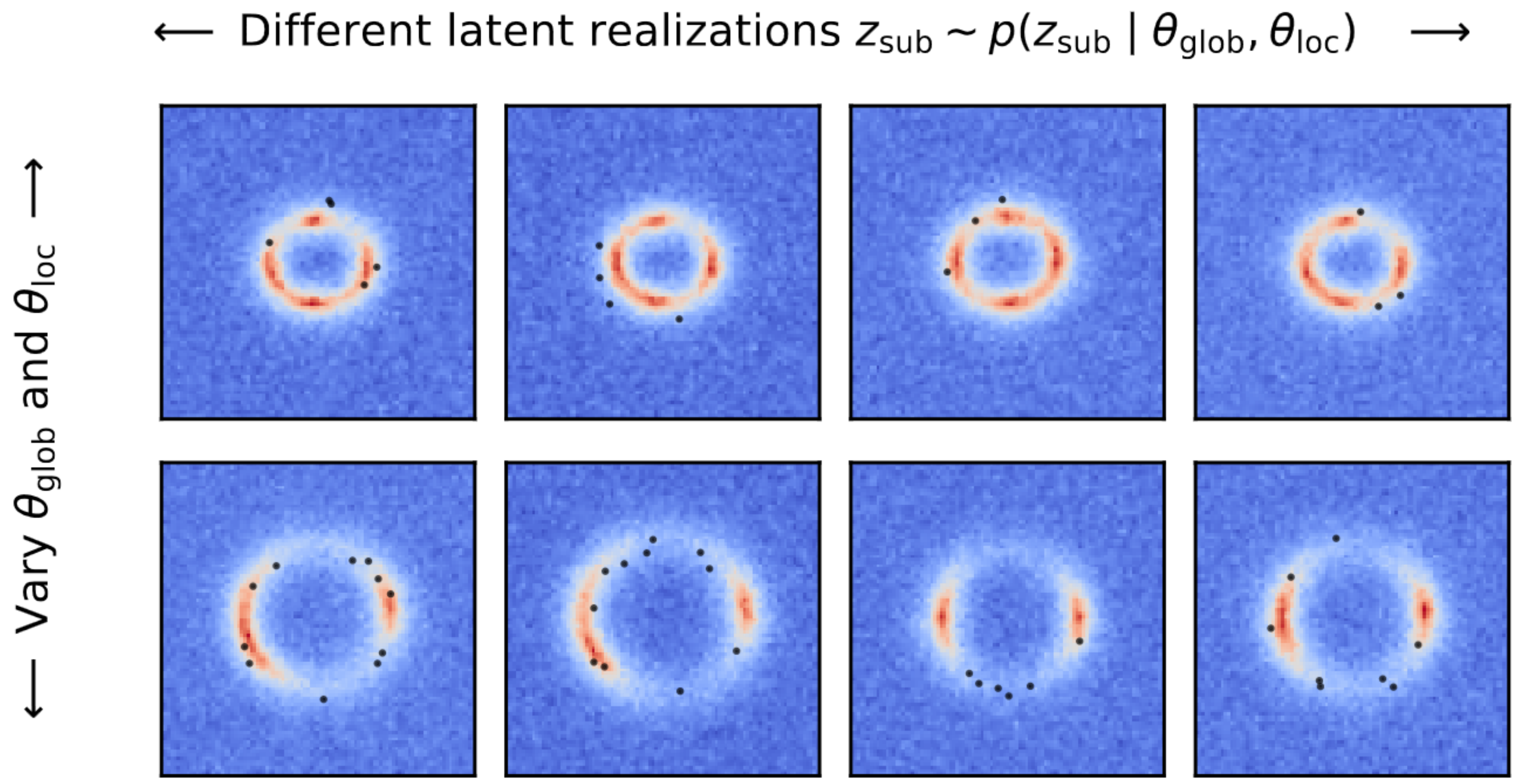
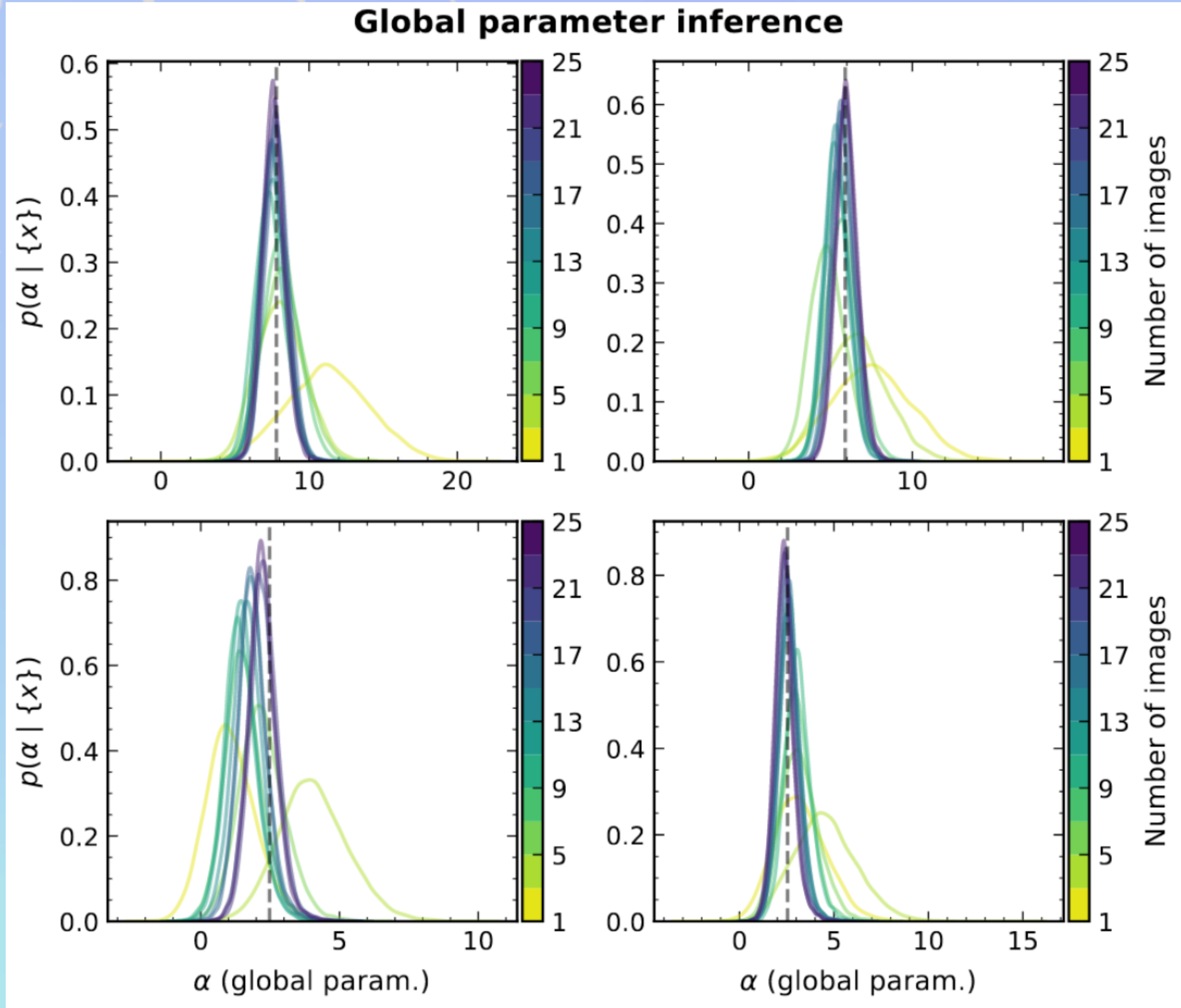
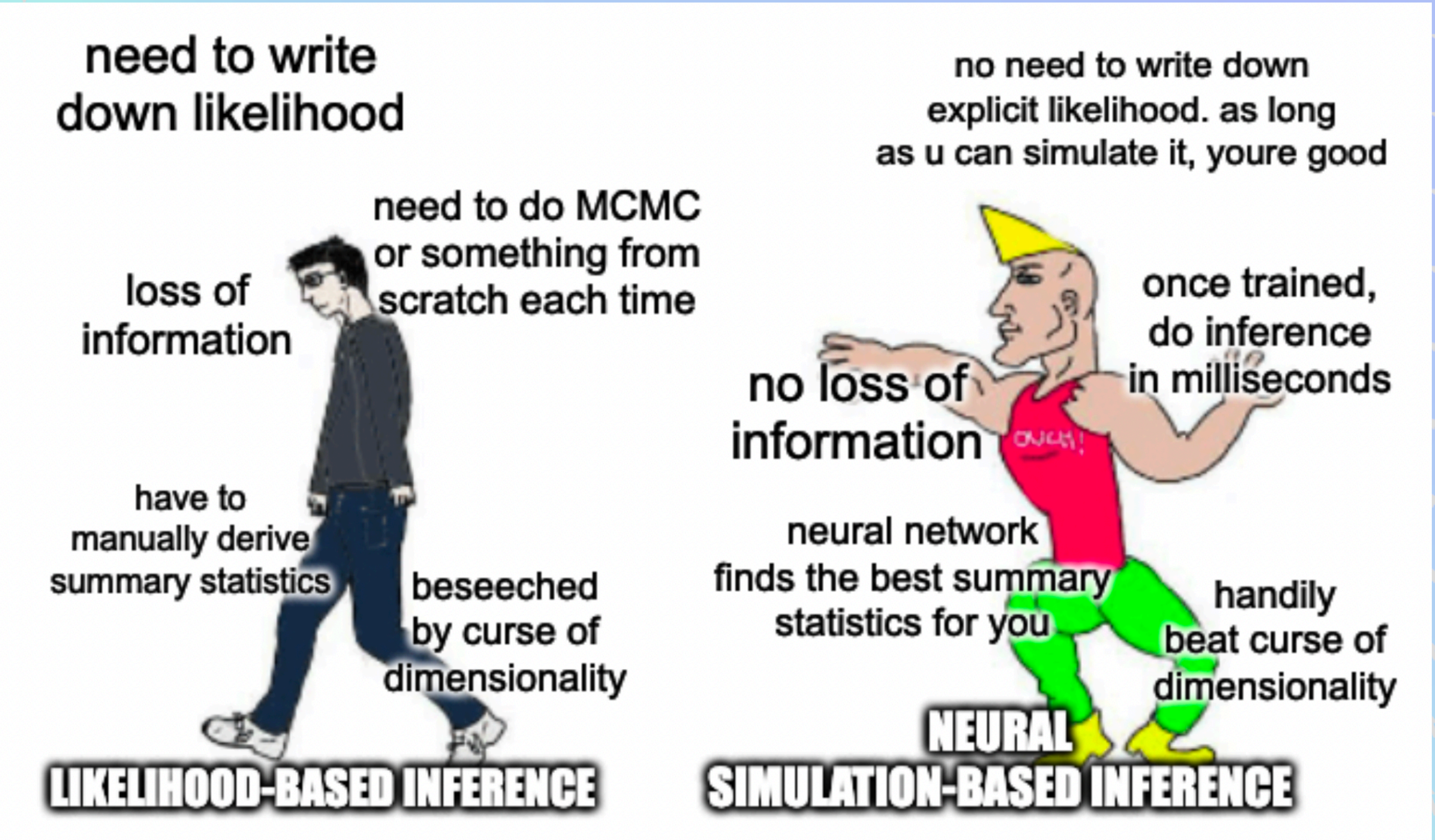


Figure 5: Illustrative samples from the lensing model. The rows show two different choices of local (per-event) parameters, while the different columns show variations on the global (set-wide) parameters for the fixed choice of local parameters. Sample-to-sample variation induced by the global parameters, which control the abundance of a subhalo population in the lens, can be seen. The scatter points shows the location of individual subhalos in each image.





credit: Siddarth Mishra-Sharma

before we can talk about neural inference,

we have to believe machine-learning is doing what it claims!

why does it work?
(from the perspective of a physicist…)

$$|\,\text{continuous}\,(\mathbb{R}^m \to \mathbb{R}^n)\,| = |\,\mathbb{N} \to \mathbb{R}\,|$$

continuous real functions $\leftrightarrow$ functions from $\mathbb{N}$ to $\mathbb{R}$.

$$|\,\mathrm{continuous}\,(\mathbb{R}^m \to \mathbb{R}^n)\,| = |\,\mathbb{N} \to \mathbb{R}\,|$$

continuous real functions $\leftrightarrow$ functions from $\mathbb{N}$ to $\mathbb{R}$.

i.e. any continuous function can be *parameterized* by a countable number of real numbers.

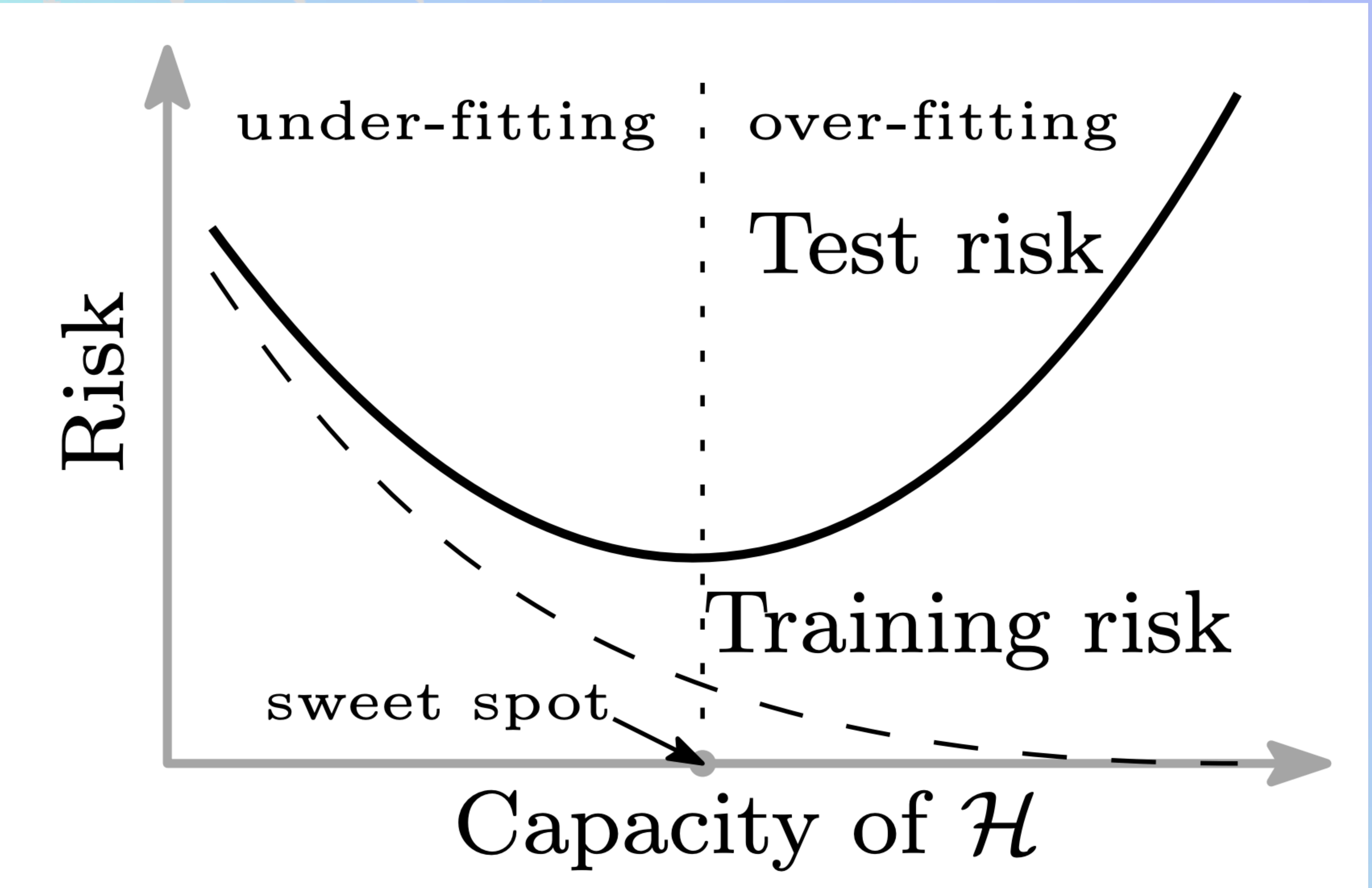
$$|\,\text{continuous}\,(\mathbb{R}^m \to \mathbb{R}^n)\,| = |\,\mathbb{N} \to \mathbb{R}\,|$$

continuous real functions $\leftrightarrow$ functions from $\mathbb{N}$ to $\mathbb{R}$.

i.e. any continuous function can be *parameterized*
by a countable number of real numbers.

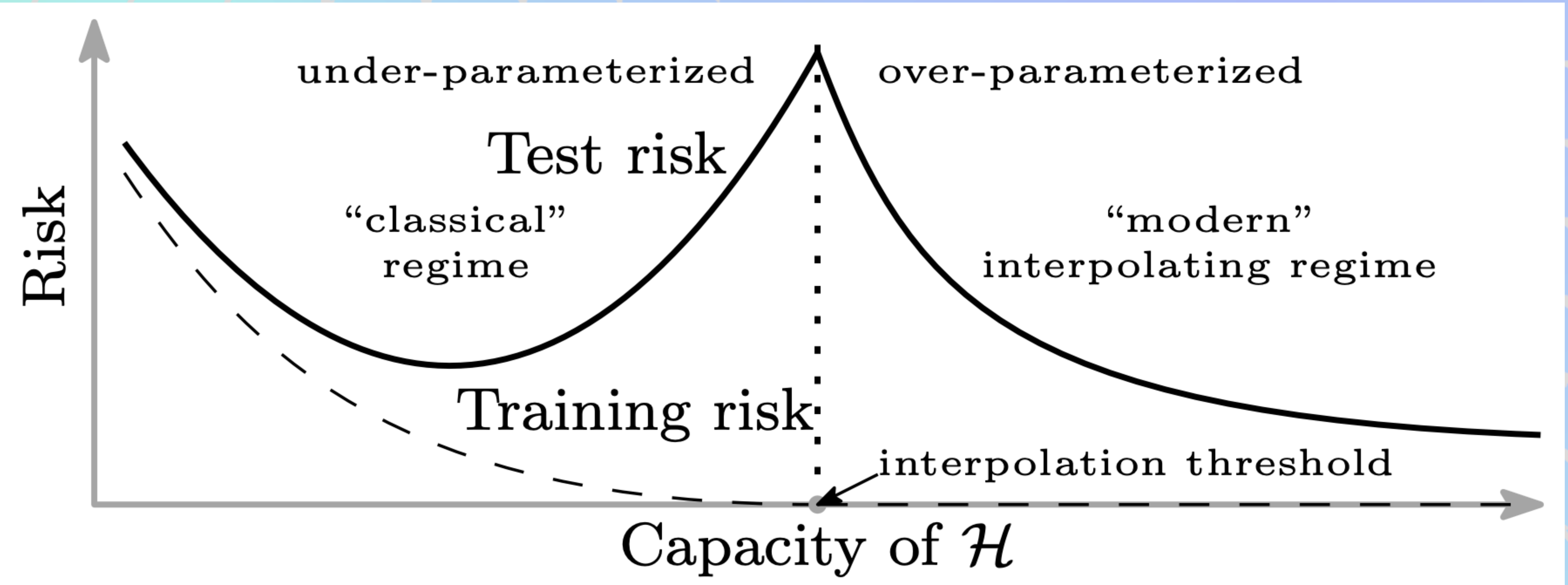
(underpins e.g. Fourier series, eigenvector decomposition, etc)

Risk ~ error of fit on unseen data

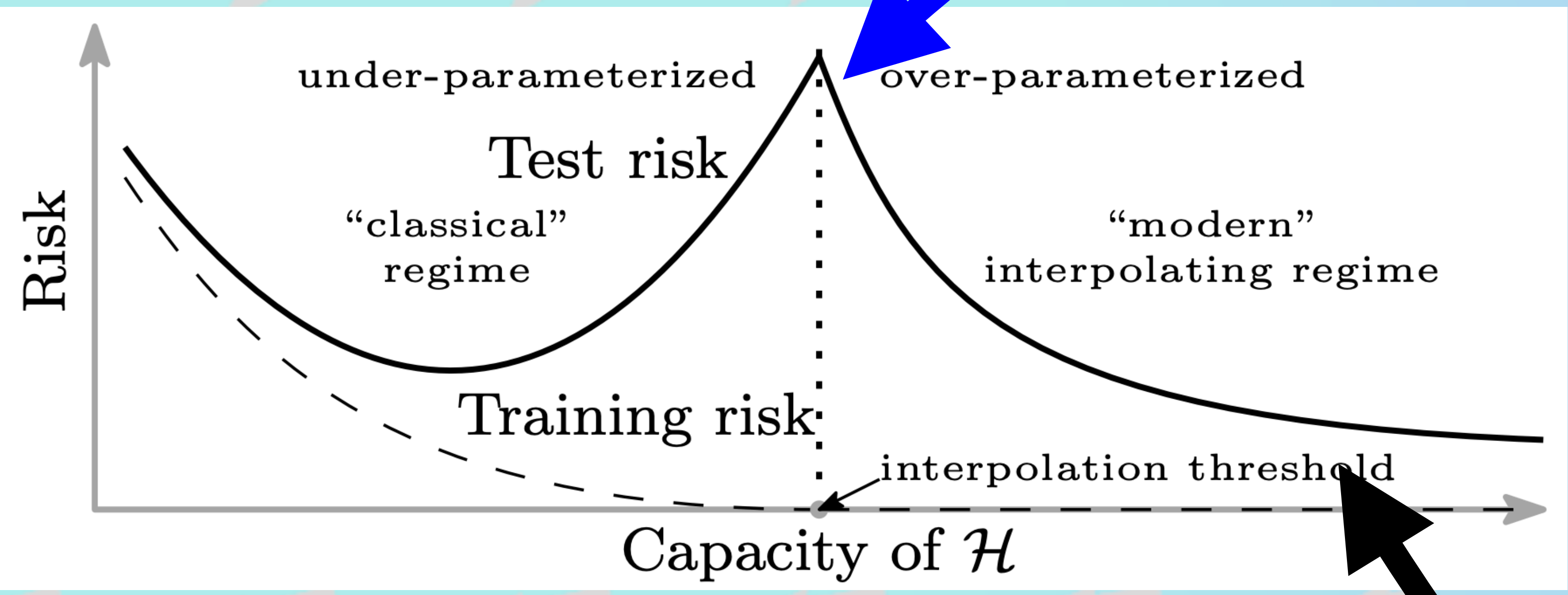


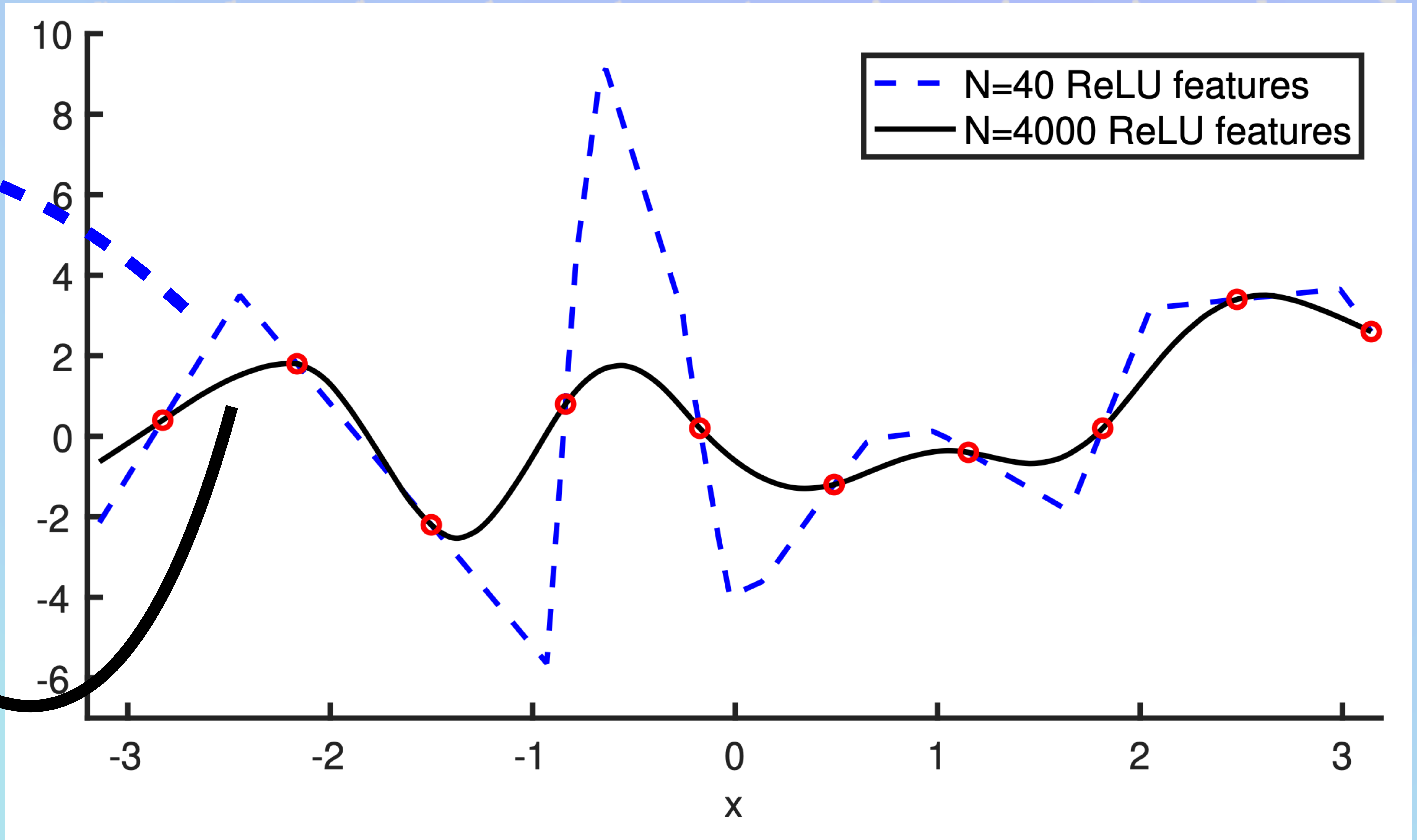
“classical” view of fitting data

Risk ~ error of fit on unseen data

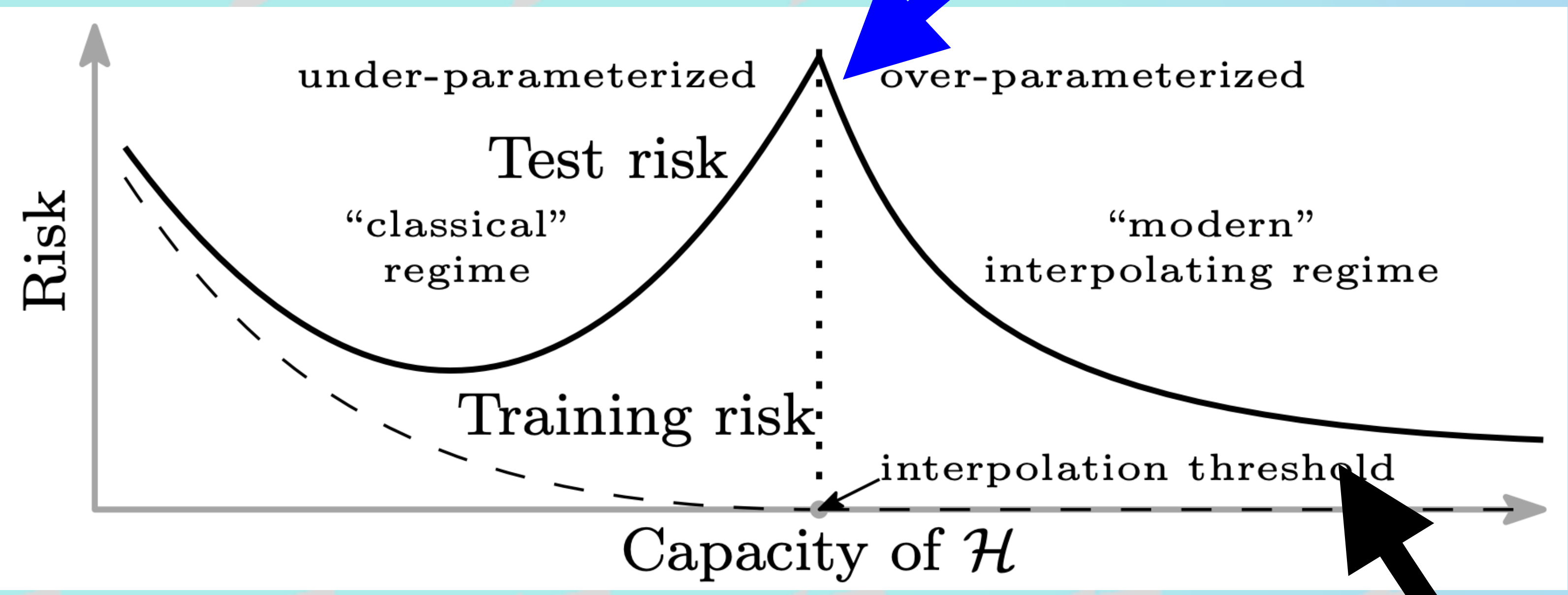


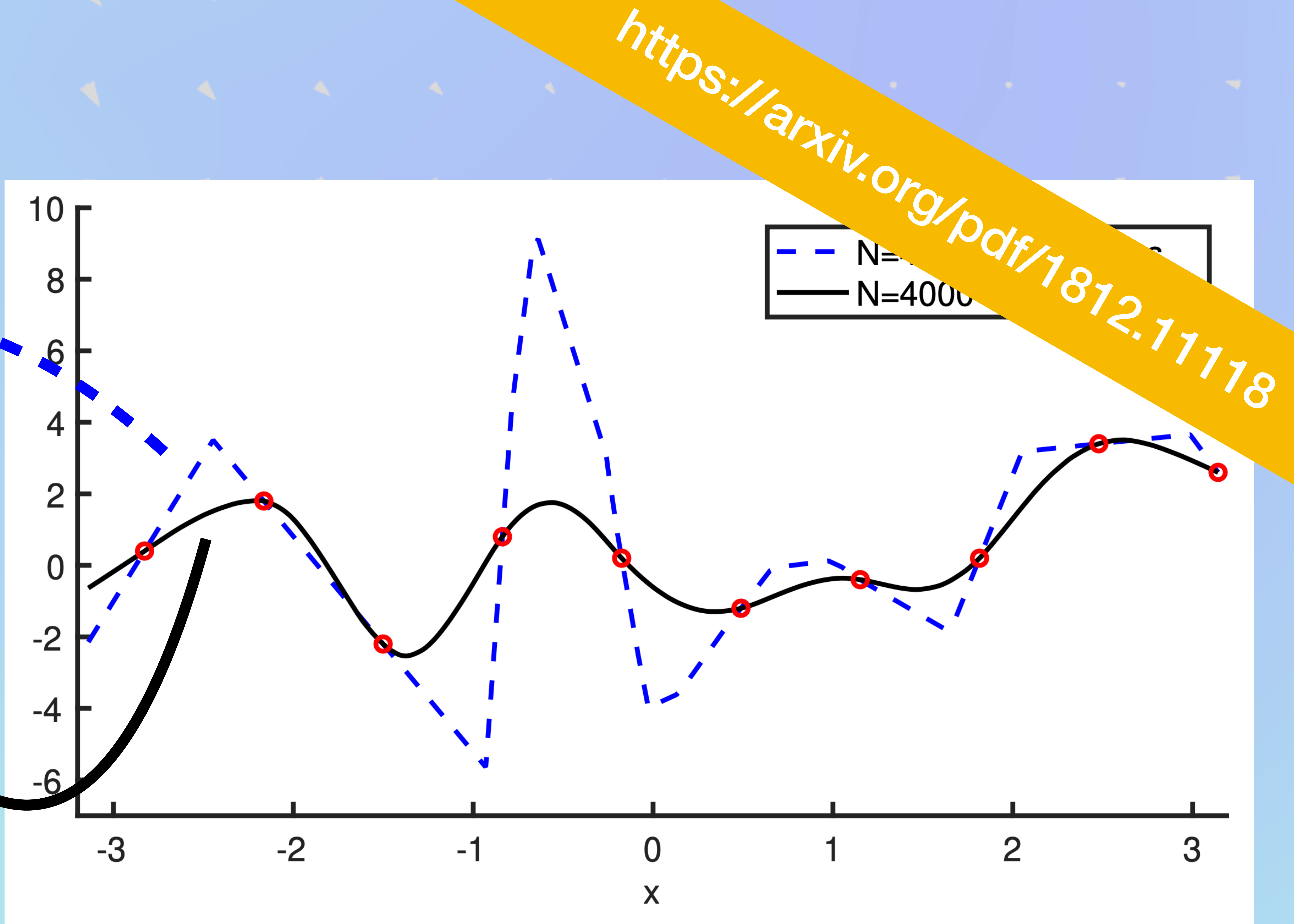
growing (empirical) evidence:
"classical" under- and over-fitting ideals do not apply.





growing (empirical) evidence:
"classical" under- and over-fitting ideals do not apply.





growing (empirical) evidence:
"classical" under- and over-fitting ideals do not apply.

to summarize:

1) NNs approximate any real function to arbitrary precision with a finite number of parameters (and enough training data).

2) over-parameterization yields high-quality (very predictive) fits.

3) over-parameterization *increases* the odds the fit converges to a "good" local minimum.

mounting evidence that NNs really are learning what they claim.

to summarize:

1) NNs approximate any real function to arbitrary precision with a finite number of parameters (and enough training data).

2) over-parameterization yields high-quality (very predictive) fits.

→ growing confidence that NNs are learning what they claim to.