Conditional generation
in the LHC context

Tobias Golling,

University of Geneva

ν^2-Flows: Fast and improved neutrino reconstruction in multi-neutrino final states with conditional normalizing flows

John Andrew Raine,[1,*] Matthew Leigh,[1,†] Knut Zoch,[1,2] and Tobias Golling[1]

[1]*Département de physique nucléaire et corpusculaire, University of Geneva, Switzerland*

[2]*Laboratory for Particle Physics and Cosmology, Harvard University, Cambridge, 02138 MA, USA*

4. **Ambiguities**
   Often interpreting the underlying features of an event in terms of the observed particles can be ambiguous. For example in events with two top quarks, each decaying via the sequence t->bW, W->mu+neutrino, there are 6 unknowns (the components at each neutrino's 3-momentum) and 6 constraints; because the some of the constraints are quadratic, we can get 4 different solutions. ML procedures can distinguish among them, but how are they doing this. Are they using extra information, not used by the analytic solutions, or is it via the training samples we are using (see next point)?

Quick intro for wider audience

Generic tool of conditional generation

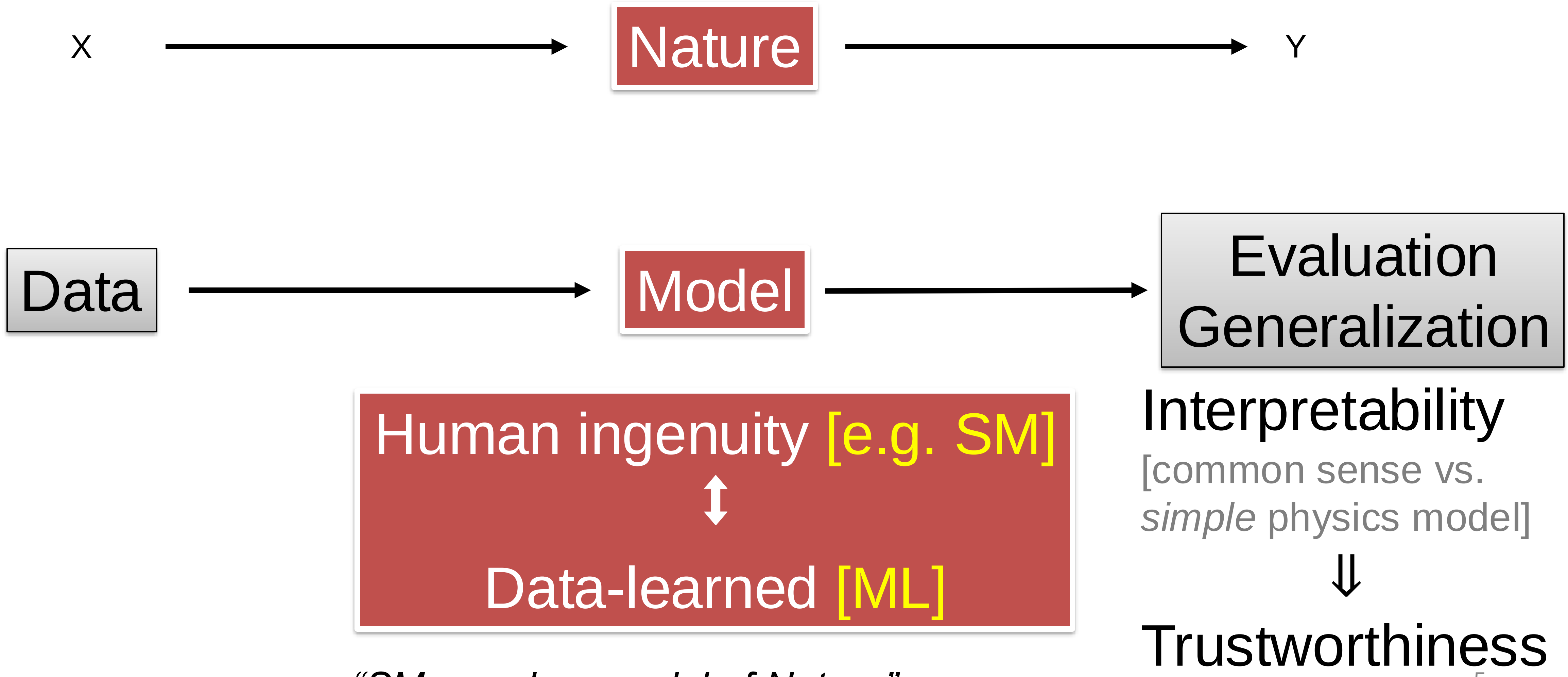
Concrete application to infer neutrino kinematics

Discussion

Task: learn a model from data

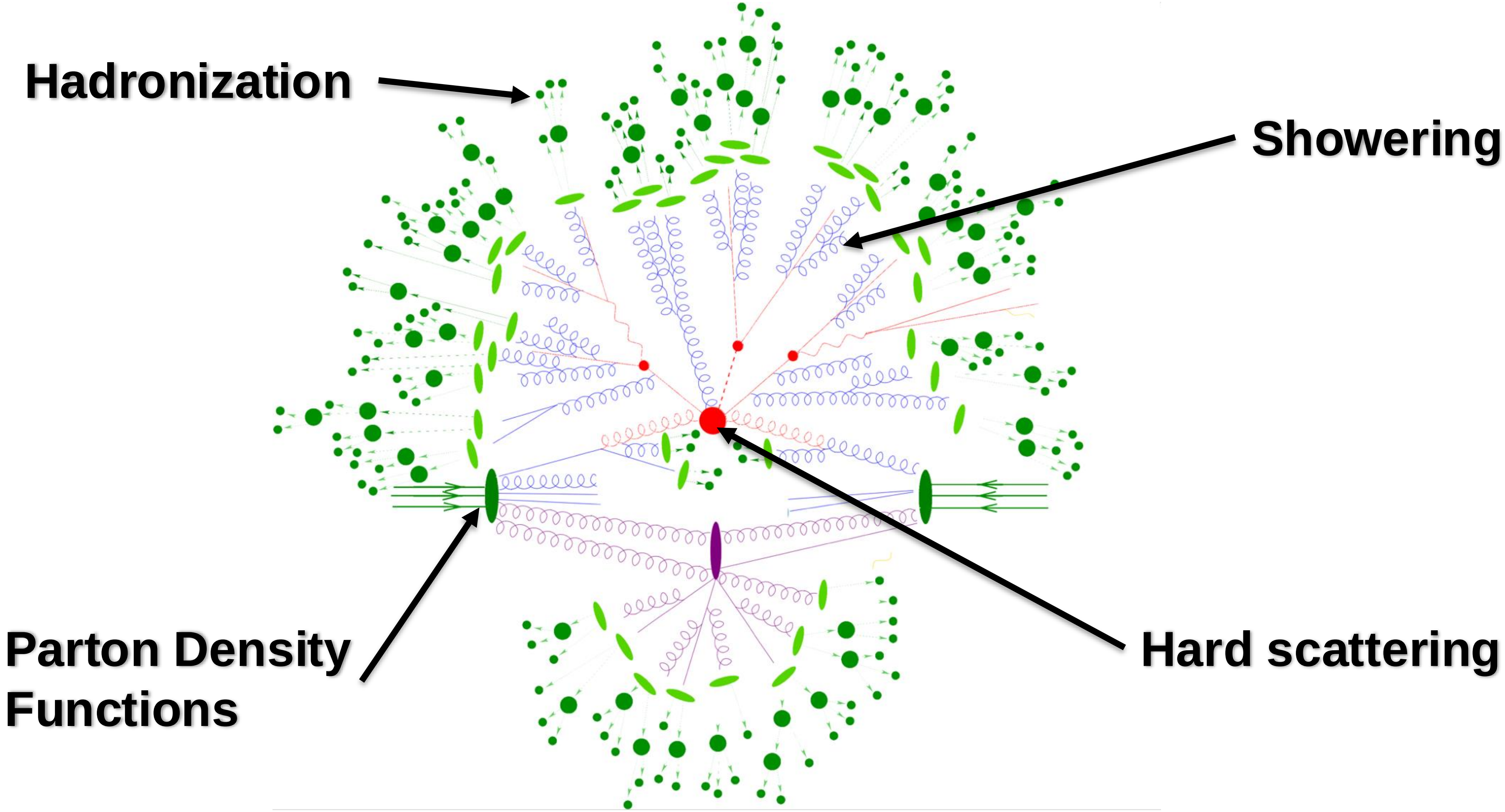
Scientists **model** the world

[Leo Breiman 2001 on statistical modeling: the two cultures]

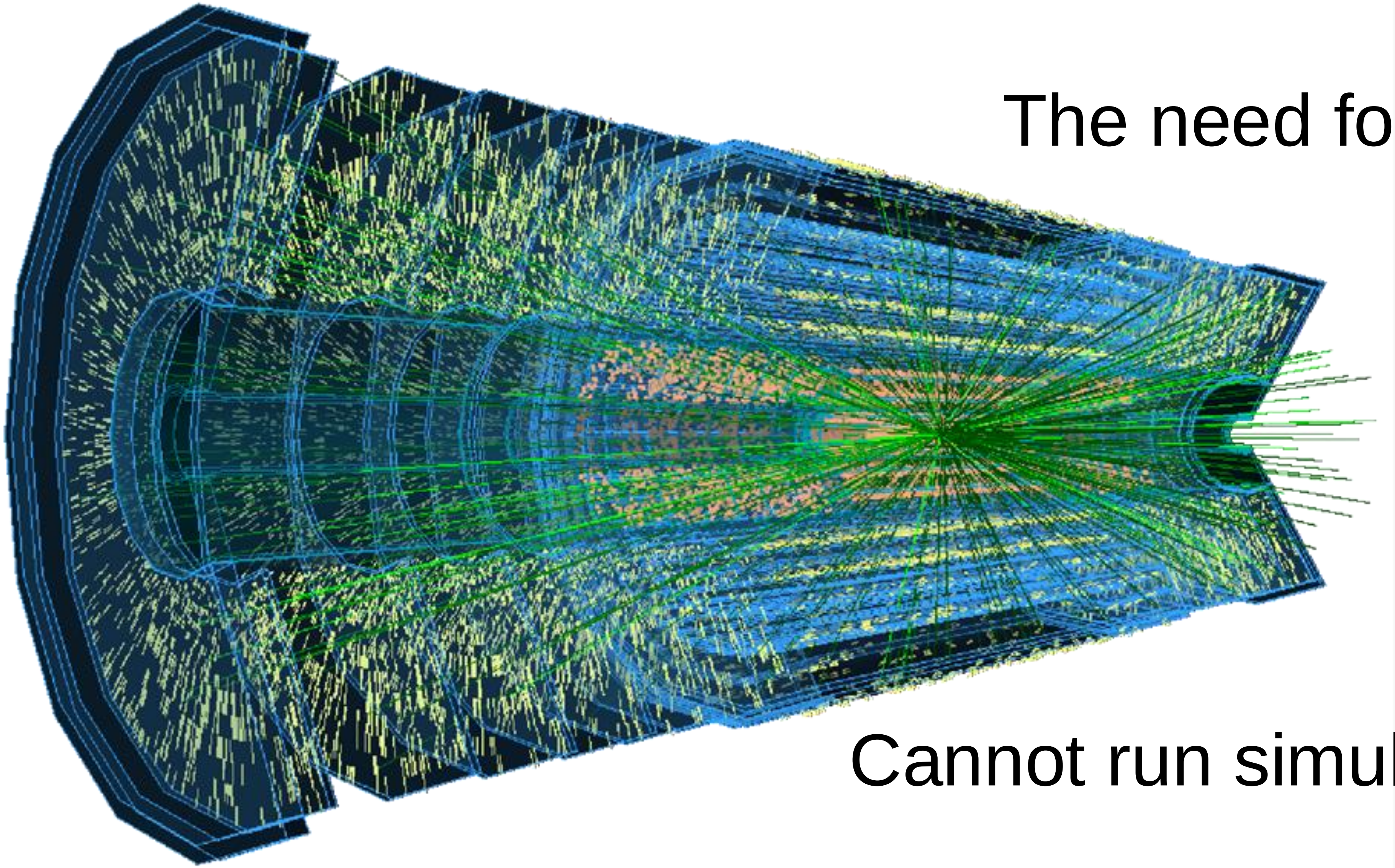


“SM merely a model of Nature”

Cannot calculate P(**data**|**theory**)



Can forward **simulate** P(**data**|**theory**)

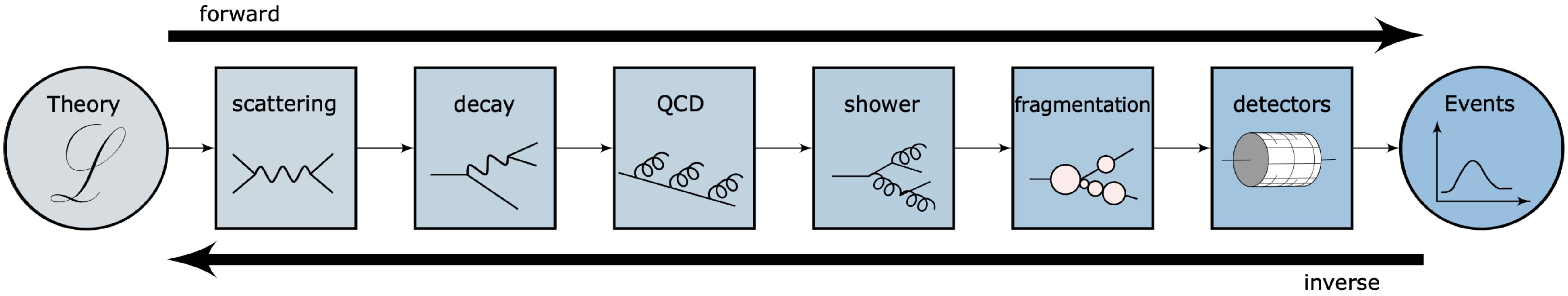


The need for synthetic data:
MC simulation

Cannot run simulator backwards

The inverse problem: inference

P(**data**|**theory**)



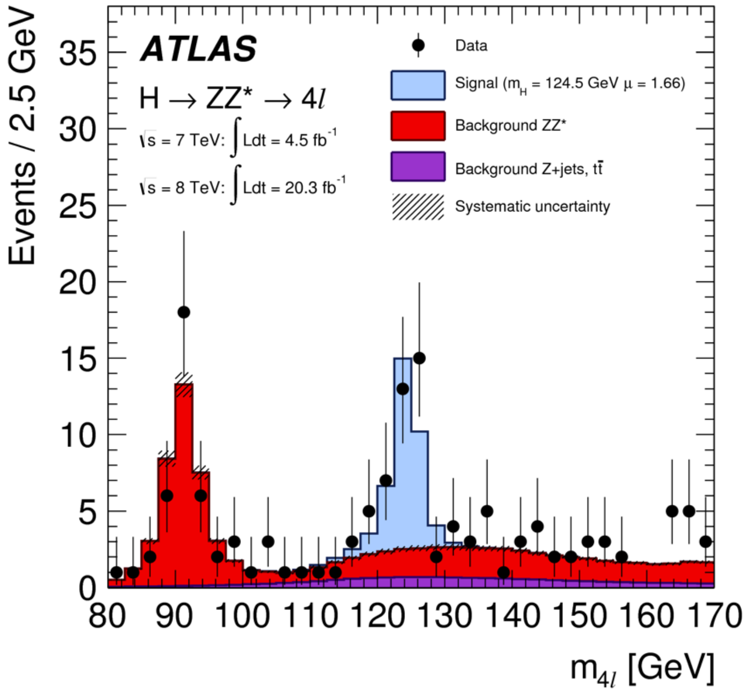
P(**theory**|**data**)

[image from 2203.07460]

Sufficient test statistics?

Project to O(1) dimension

Meaningful learned representation



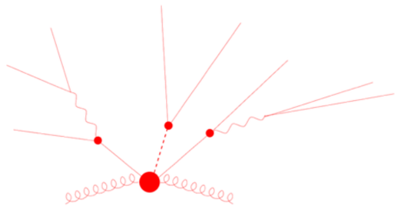
$$p(x)$$

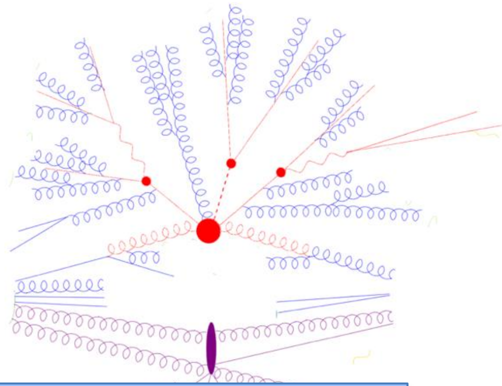
1D < x < 100M-D

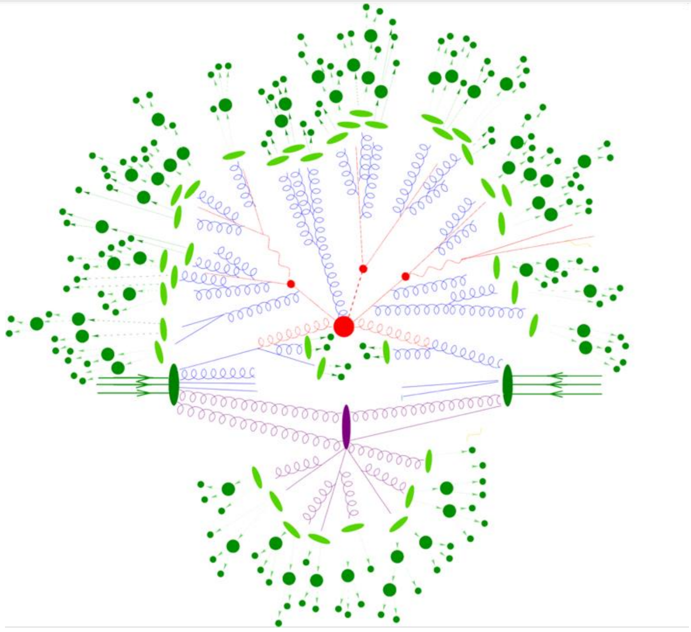
No guarantee of optimality !

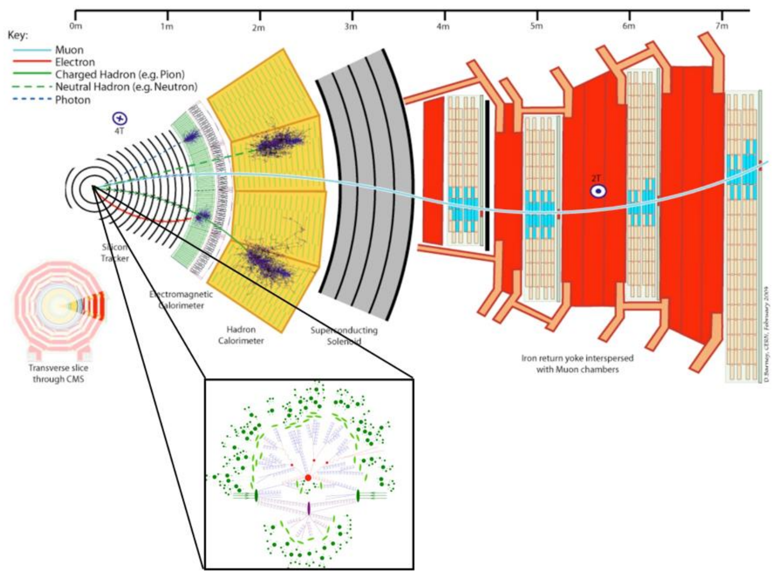
Unfolding: reco → *truth* space

Folding









Unfolding allows to
- Compare at theory level
- Compare between experiments
- Data preservation: more *useful* data

Hard scatter

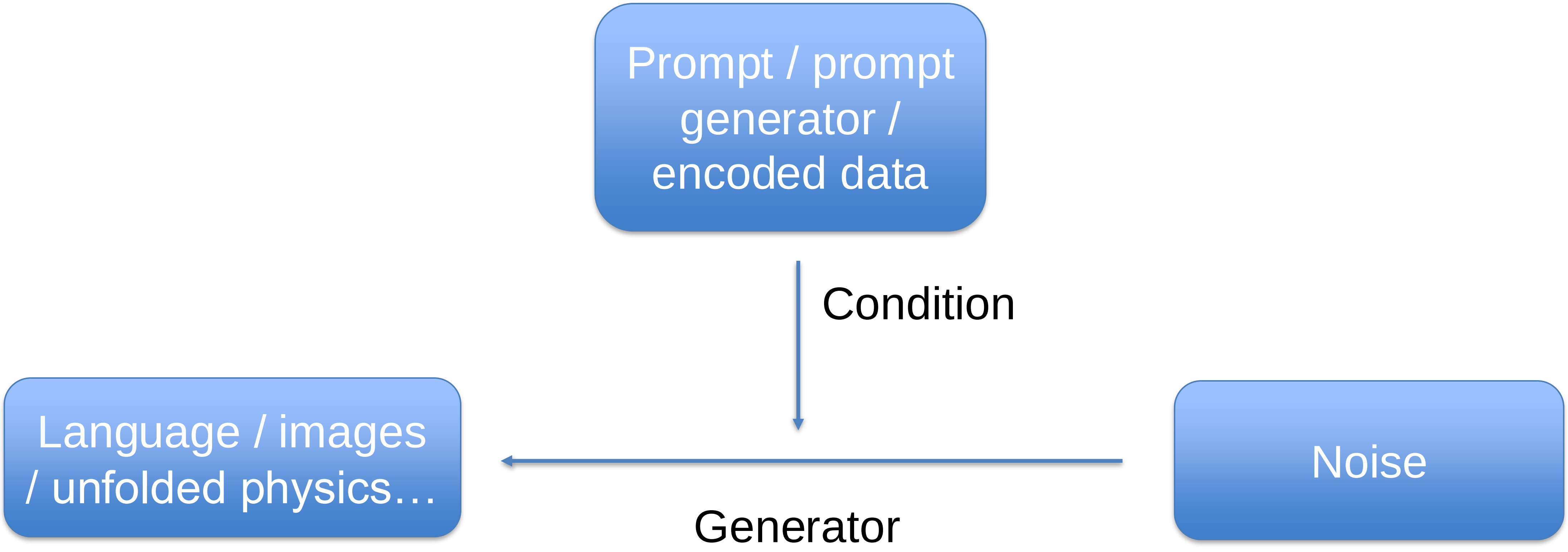
Radiation

Hadronization

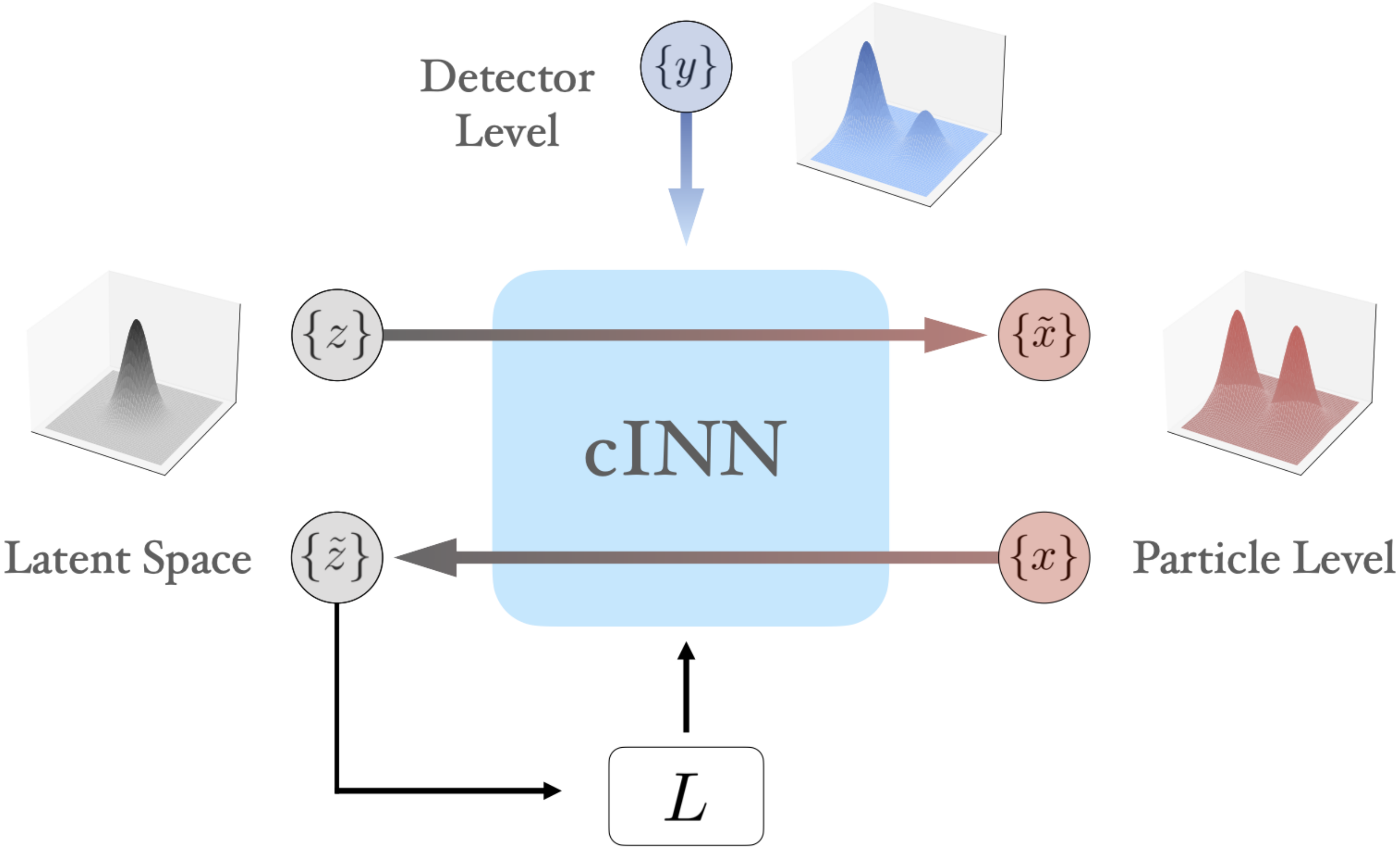
Detector

Unfolding

The tool: conditional generation



Conditional generative unfolding



[2212.08674, see also talk by Tilman Plehn yesterday]

Conditional generation

Can use this technique to generate ANY distribution in our simulation chain

We can simulate the target ⇒ we can train a surrogate

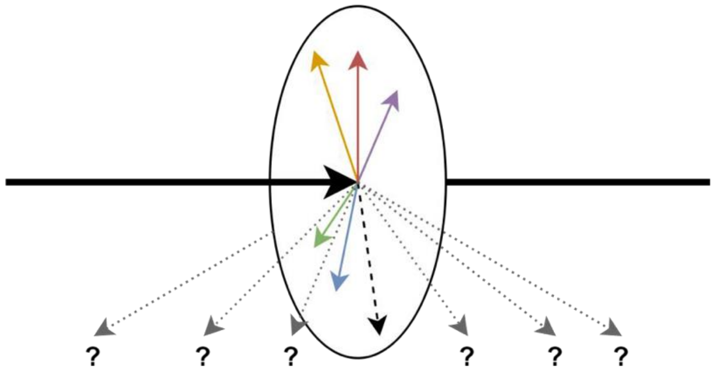
Including weakly interacting particles like neutrinos

Neutrinos are special

Neutrinos don’t interact with the detector

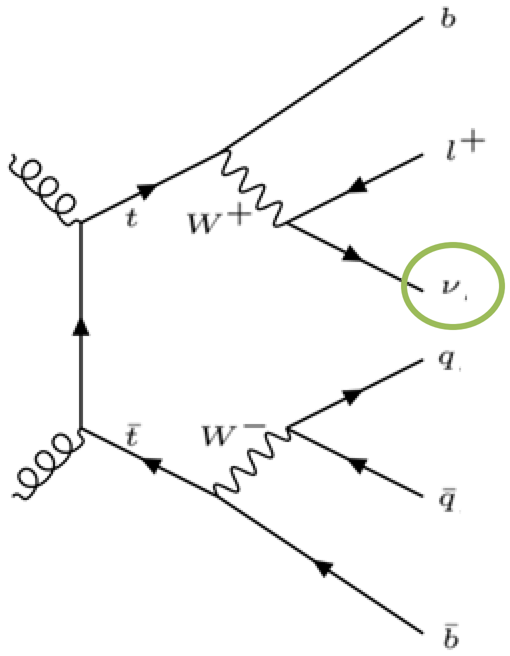
Infer their presence from conservation of momentum in transverse plane

Longitudinal component *unconstraint*



Learn ν **3**-momentum

Why interesting?



Top quark reconstruction & measurements

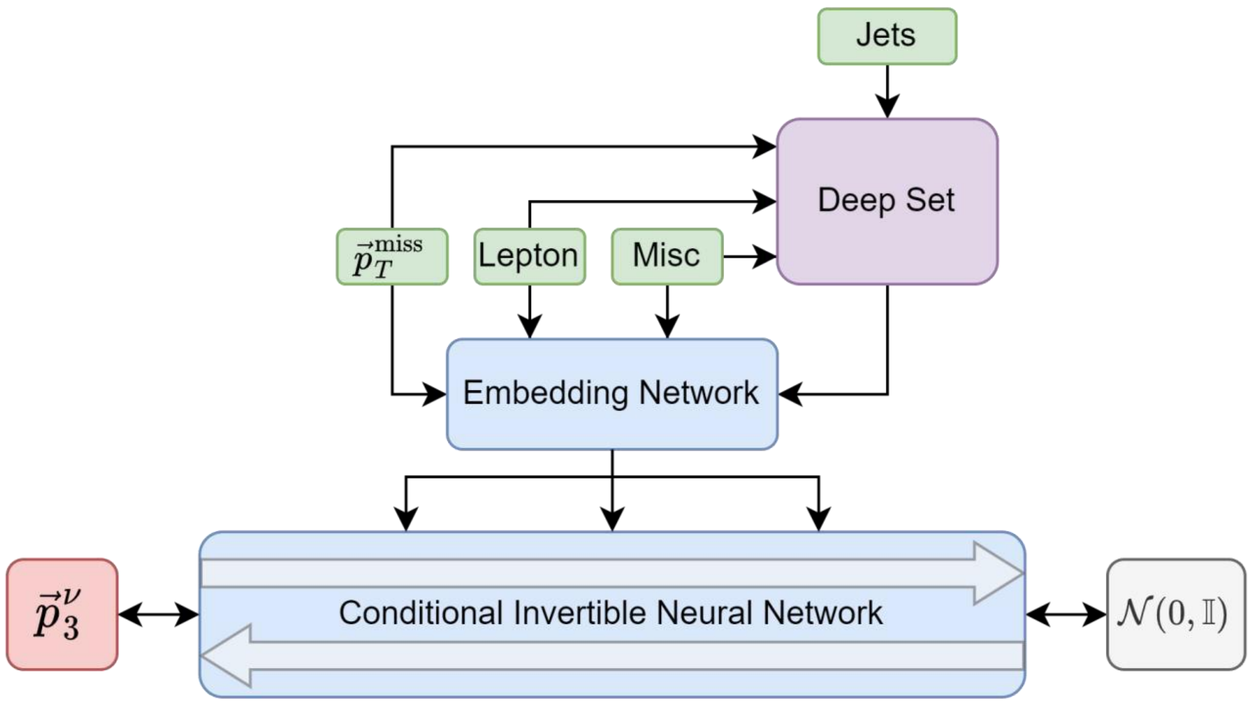
Combinatorics: assign jets to partons

Interpretability:

human-understandable distributions

[2207.00664]

Conditional generation: ν-flows



Two components:

CINN to generate neutrino 3-momentum

Embedding network to encode event information

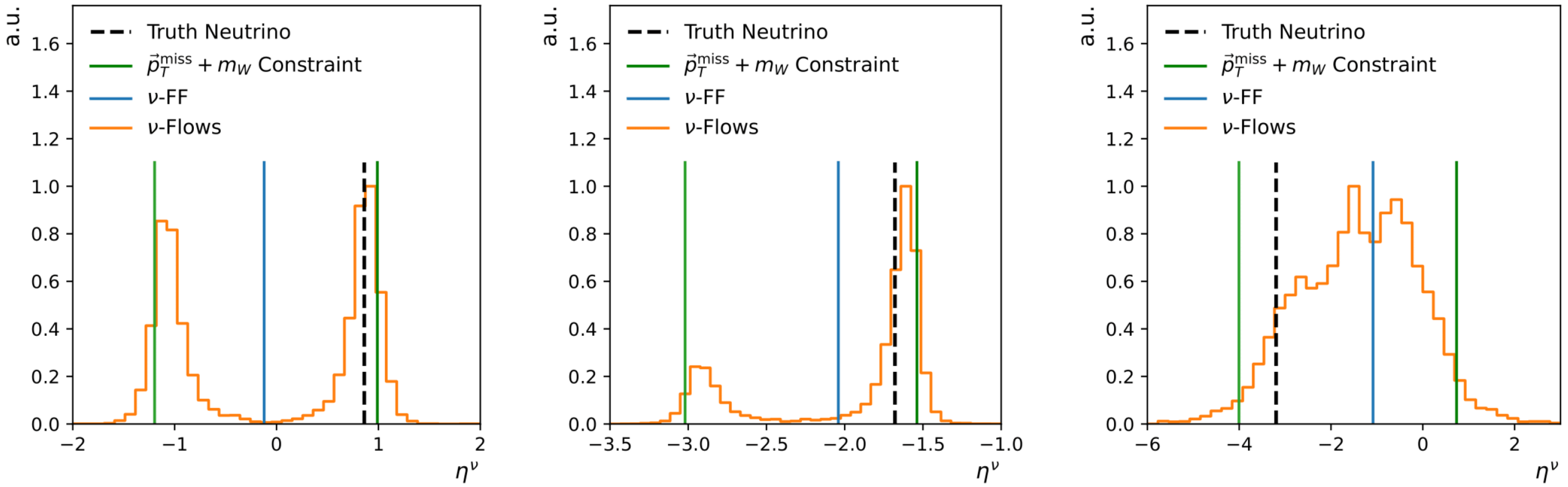
[2207.00664]

Learn conditional probability $p(\vec{p}_3^{\,\nu}|\text{Event})$

Conditional probability over neutrino momenta **assuming an underlying process**

Can sample from this posterior for a given observed event

[2207.00664]



ν-flows summary

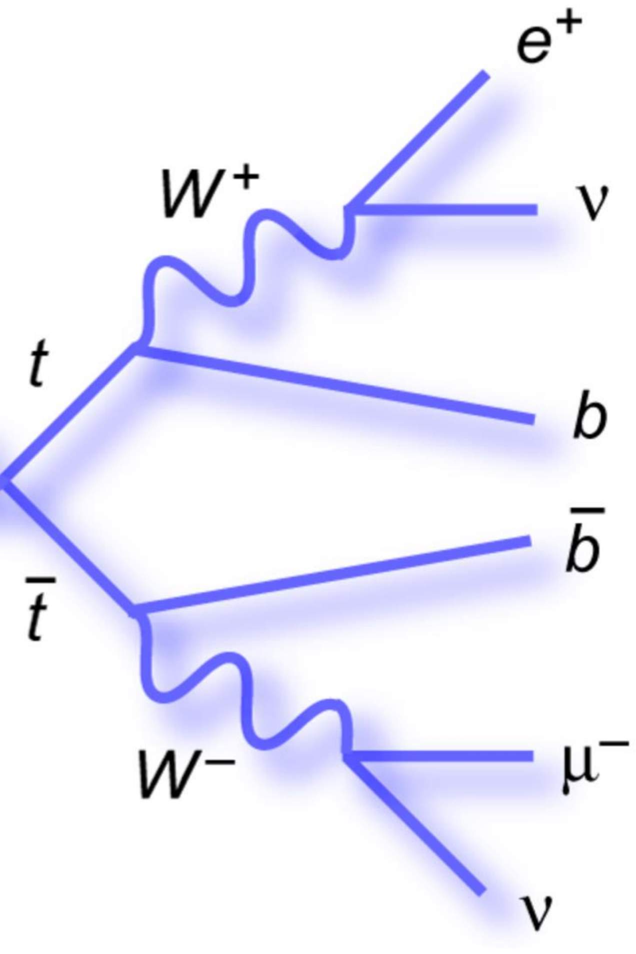
Meaningful probabilistic treatment

Learning conditional density of particle-level quantities conditioned on reconstructed inputs

Improve over traditional method

[2207.00664]

Adopt to events with 2 neutrinos: ν^2-flows



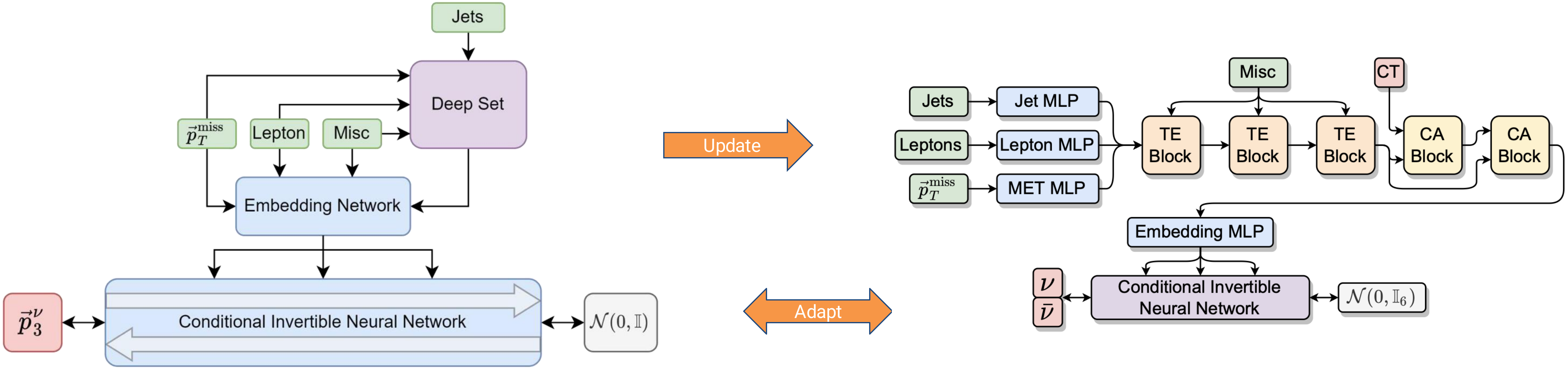
Conceptionally identical

Output: 6D vector

Embedding network updated to TE

[2307.02405]

ν-flows $\Rightarrow$ ν^2-flows



[Transformer Encoders & Cross Attention]

[2207.00664]

[2307.02405]

Reference methods

Compare to two standard approaches (relying on hard assumptions on mass)

Neutrino Weighting

Ellipse method

$$(\ell_{1,2} + \nu_{1,2})^2 = m_w^2 = (80.38\,\mathrm{GeV})^2,$$

$$(\ell_{1,2} + \nu_{1,2} + b_{1,2})^2 = m_t^2 = (172.5\,\mathrm{GeV})^2,$$

Scan eta values for both neutrinos
Choose solution which maximises a weight

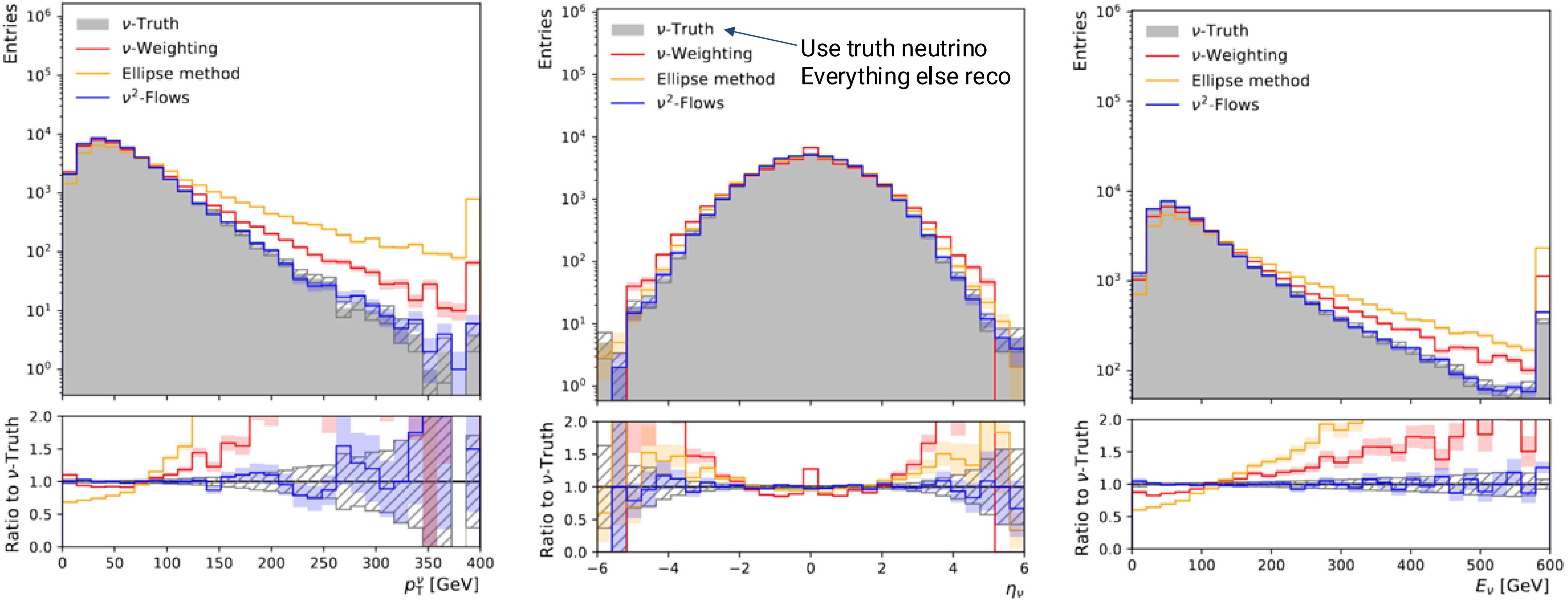
$$w = \exp\left(-\frac{||\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} - \vec{p}_{\mathrm{T}}^{\,\nu\bar{\nu}}||_2^2}{2\sigma^2}\right)$$

Additionally scan top quark mass values to improve acceptance

Use observed missing momentum to constrain solution further

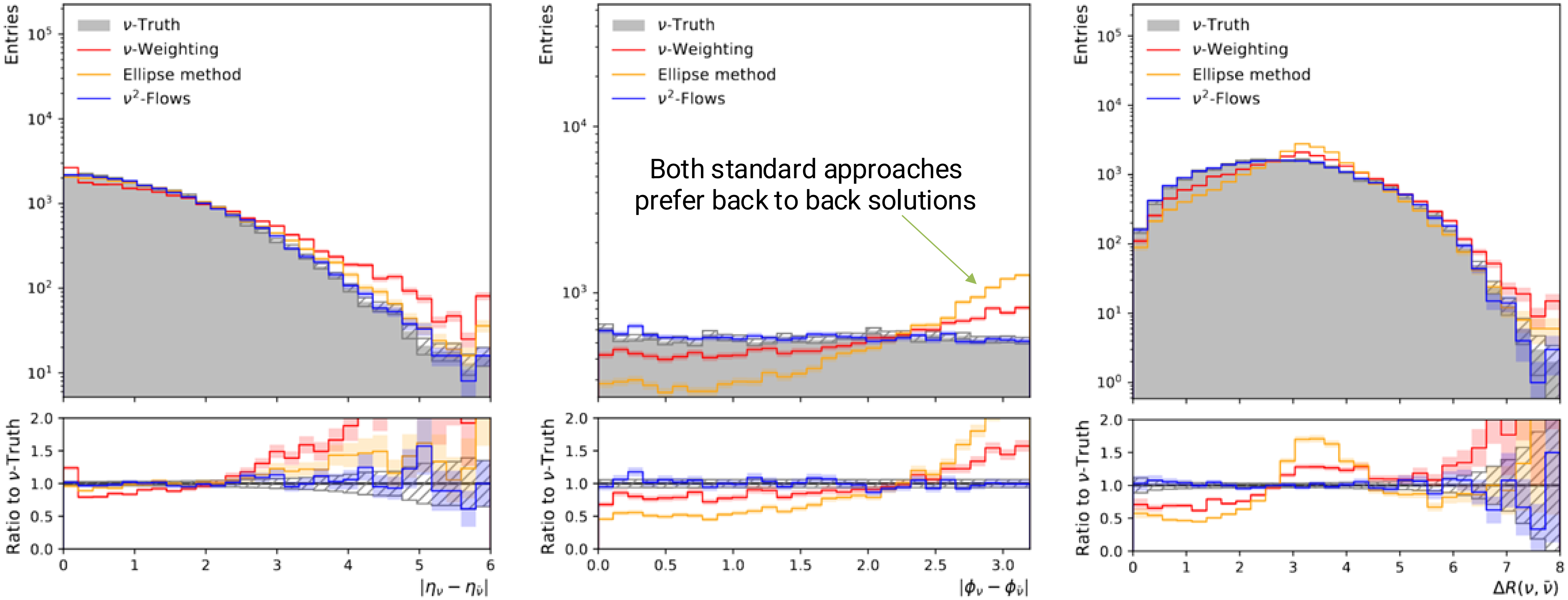
Less flexible to resolution effects but computationally more efficient

Kinematics



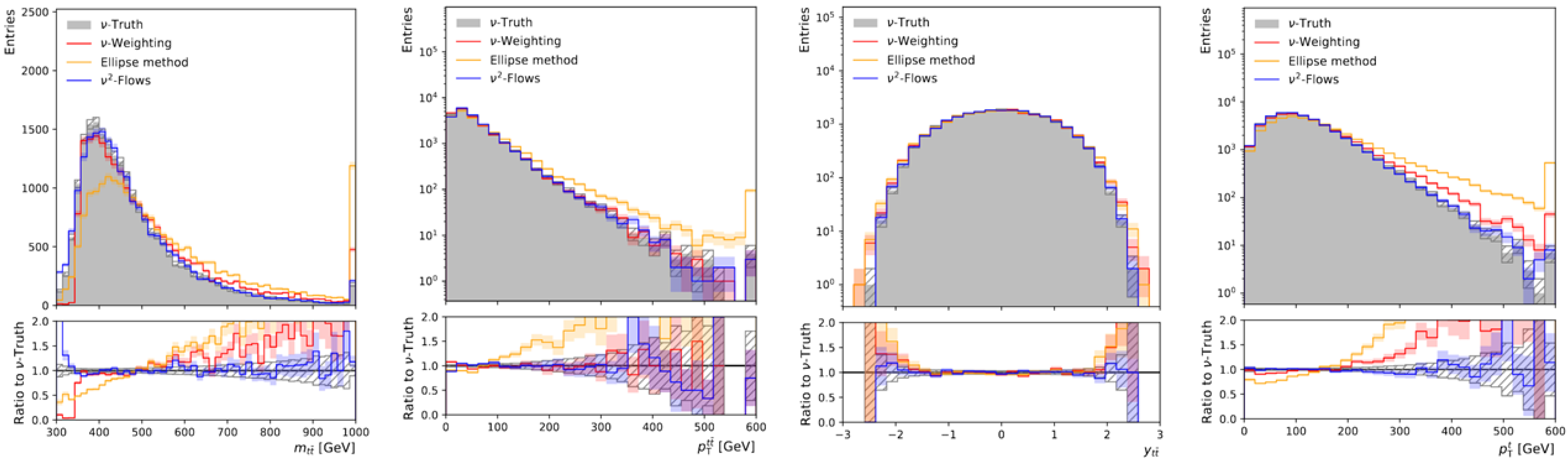
[2307.02405]

Neutrino correlations



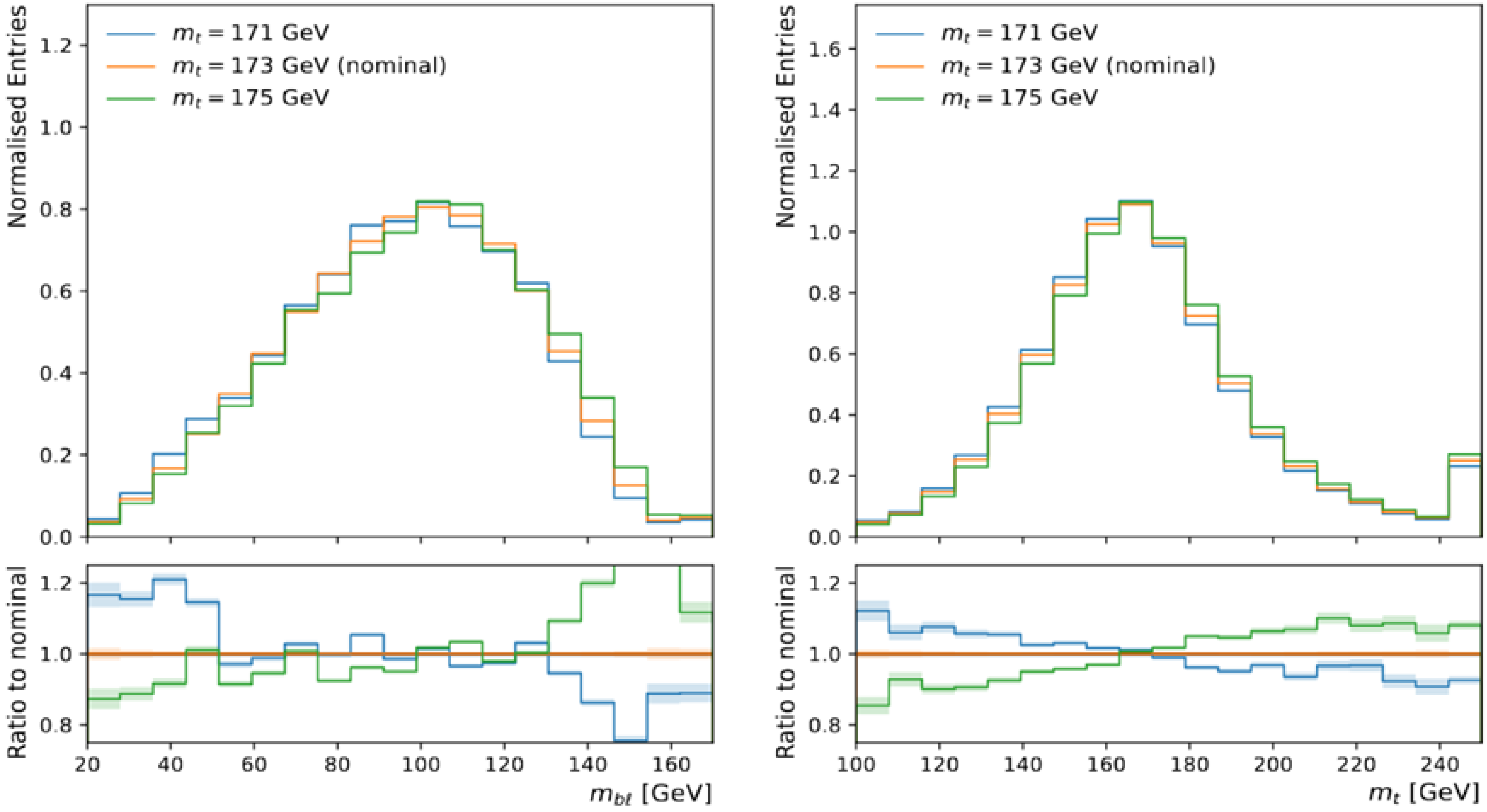
[2307.02405]

Top quark kinematics



[2307.02405]

Retain sensitivity to top mass



[2307.02405]

v^2-flows summary

- Drop-in ML solution to replace conventional approach
- Transformer + conditional normalizing flows
- Outperforms standard approaches
- Fast inference
- Mass sensitivity
- Extendable to any neutrino multiplicity & final state
- Available as off-the-shelf tool in ATLAS [TopCPToolkit]

Relevance to discussion topics

Thanks, Louis, for giving us the *opportunity* to play the **devil's advocate** !

Why do we need this intermediate result?

- **Argue one side**: useful in traditional analysis approaches; offers additional interpretation and feeds into other intermediate results
  - Gain trust, human-understandable, improve downstream tasks

- **Argue the other side**: in a perfect ML world everything learned from low-level data
  - Intermediate features superfluous

- The real world = compromise

5. Relevance of training samples

- Important to stay **in-domain**
  - e.g. train on tt and evaluate on tt

- Simulator conveniently provides all configurations in the **correct proportions**

6. Mis-modelling of training samples

Mismodeling impacts all MC-based training

(Standard) uncertainty estimate is necessary

Interpretability

4. **Ambiguities**
   Often interpreting the underlying features of an event in terms of the observed particles can be ambiguous. For example in events with two top quarks, each decaying via the sequence t->bW, W->mu+neutrino, there are 6 unknowns (the components at each neutrino's 3-momentum) and 6 constraints; because the some of the constraints are quadratic, we can get 4 different solutions. ML procedures can distinguish among them, but how are they doing this. Are they using extra information, not used by the analytic solutions, or is it via the training samples we are using (see next point)?

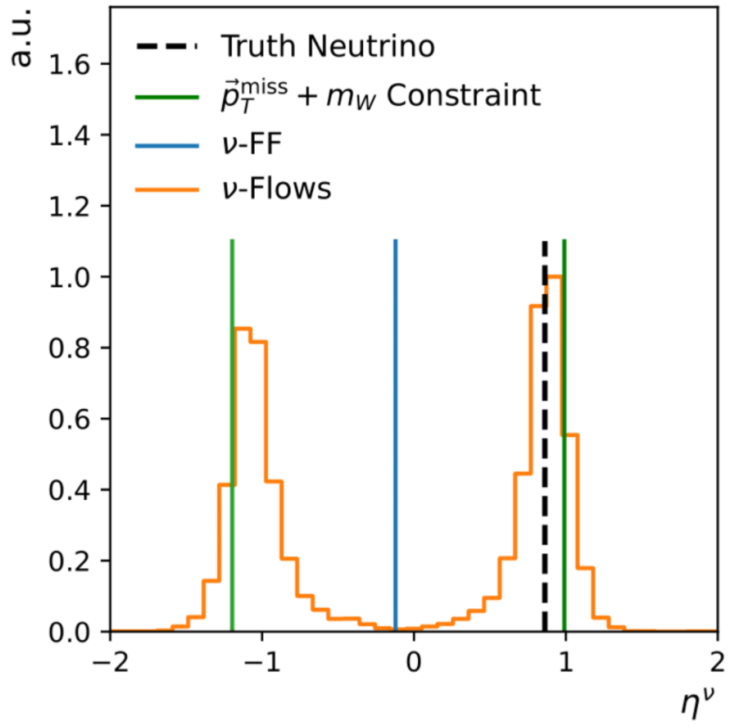
All inverse problems ill-defined in the sense that it is a many-to-one mapping which we try to invert

Can still do it – with some regularization…

Neutrino reconstruction is a version of this concept

Why can we learn *something* about the ν's?

- If nu 3-vectors were completely random
  - Data would NOT allow us to measure ν's
- There is **some** correlation between ν 3-vectors configurations and "the rest of the event we measure"
  - **Some** information about ν's can be gained



The *unknowable* theoretical accuracy limit

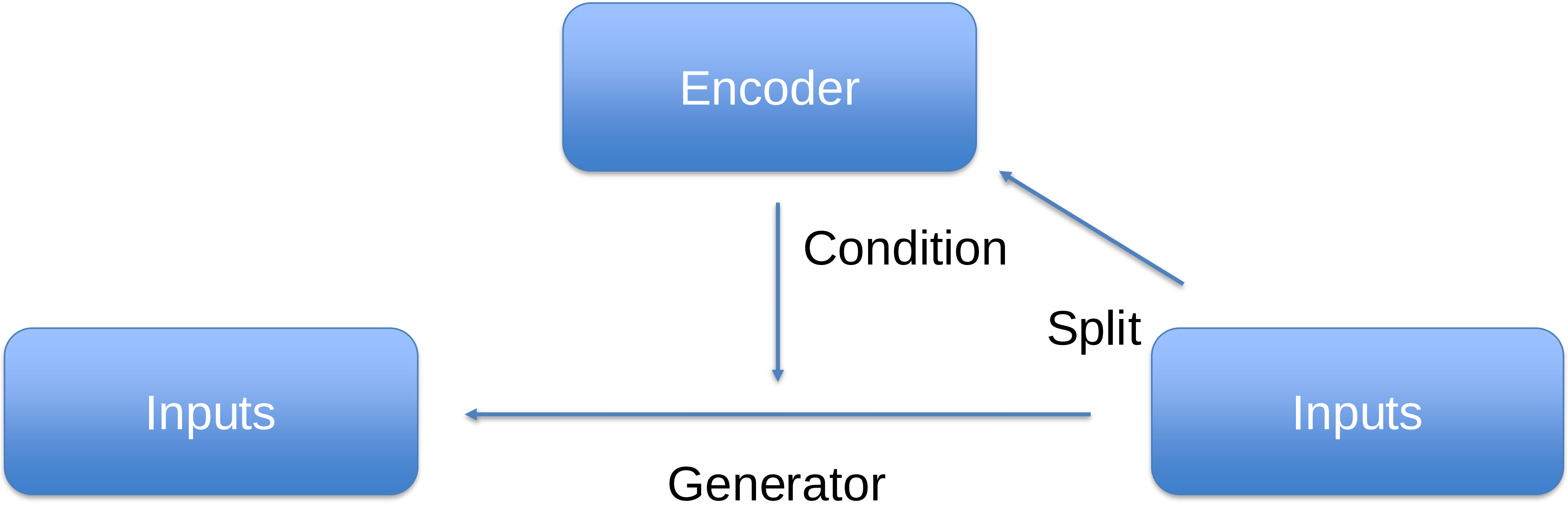
- Or: what we **can** and what we **cannot** learn from a given data set
  - A priori unknowable
  - Since we **cannot** retrace and calculate every bit

- Way out: forward simulate
  - Accuracy becomes empirical

More discussion points

- Quality of learned posterior depends on:
  - How good is the generative model
  - How much information is in our data [unknowable]
  - How well is it encoded for the conditioning
- Which one is the bottleneck?
- Where does the heavy lifting happen?
- Train end-to-end or train data-encoder separately?

Conditional generation in SSL context



Example: variation of masked particle modeling [2401.13537]