Computational challenges for unbinned ML-based analyses

Rafael Coelho Lopes de Sa (UMass),
Aishik Ghosh (UCI), Ben Nachman (LBNL)

Computational challenges for unbinned ML-based analyses (and how HPCs can be used)

Rafael Coelho Lopes de Sa (UMass),
Aishik Ghosh (UCI), Ben Nachman (LBNL)

Unbinned ML-based analyses

Roughly speaking, there are three classes of analyses that are making use of ML for unbinned, high-dimensional analysis:

Neural Simulation-based Inference (NSBI)

- Parameter Estimation - inference in the data space; result is a number(s) and a confidence interval(s) (full likelihood is a NN). **Trained on simulation.**
- Unfolding - aka deconvolution; inference is in particle-level/latent space; result is a differential cross section in many dimensions (represented as a NN). **Trained on simulation + data.**

Anomaly Detection* - inference in data space; result is cross section limit (for a given model) or p-value(s) for model-independent result. **Trained on only data.**

**Solving some of the computational challenges here are part of BN's DOE ECRP; happy to partner! Won't say more about this here, but can follow up offline.*

Unbinned ML-based analyses - *examples w/ ATLAS data*

Roughly speaking, there are three classes of analyses that are making use of ML for unbinned, high-dimensional analysis:

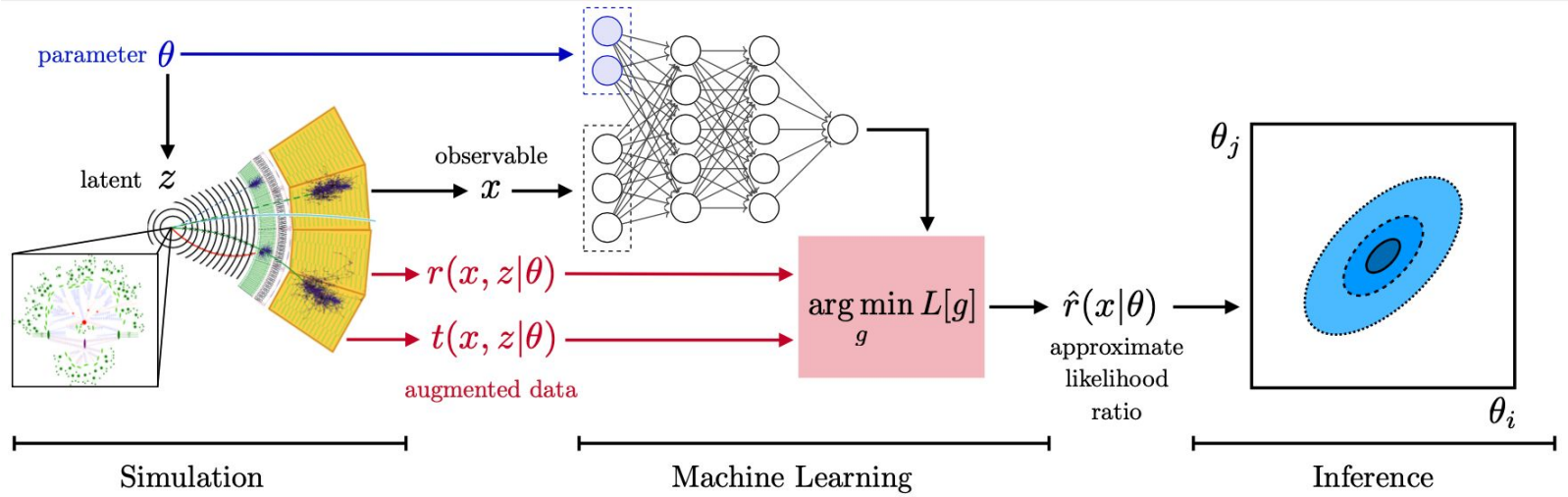
Neural Simulation-based Inference (NSBI)

- Parameter Estimation - Example: 2403.15873, trained **O(10k) NNs** for stability, precision, UQ
- Unfolding - Example: 2405.20041, trained **O(10k) NNs** for stability, precision, UQ

Anomaly Detection - Example: 2005.02983, trained **O(10k) NNs** for robustness, sensitivity, and probing search volume

UQ : uncertainty quantification

ML-based unbinned parameter inference



Very nice, but ...

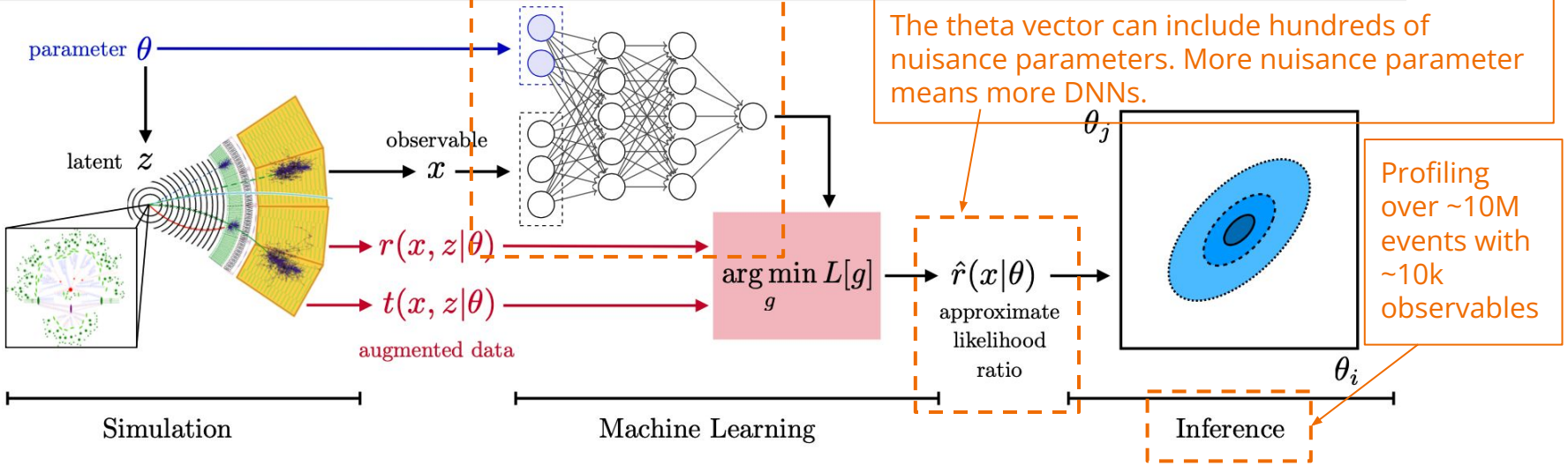
Image credit: 1805.00013

ML-based unbinned parameter inference

For realistic modeling, this NN can be thousands of DNNs with a total of 1-10B trainable parameters

The theta vector can include hundreds of nuisance parameters. More nuisance parameter means more DNNs.

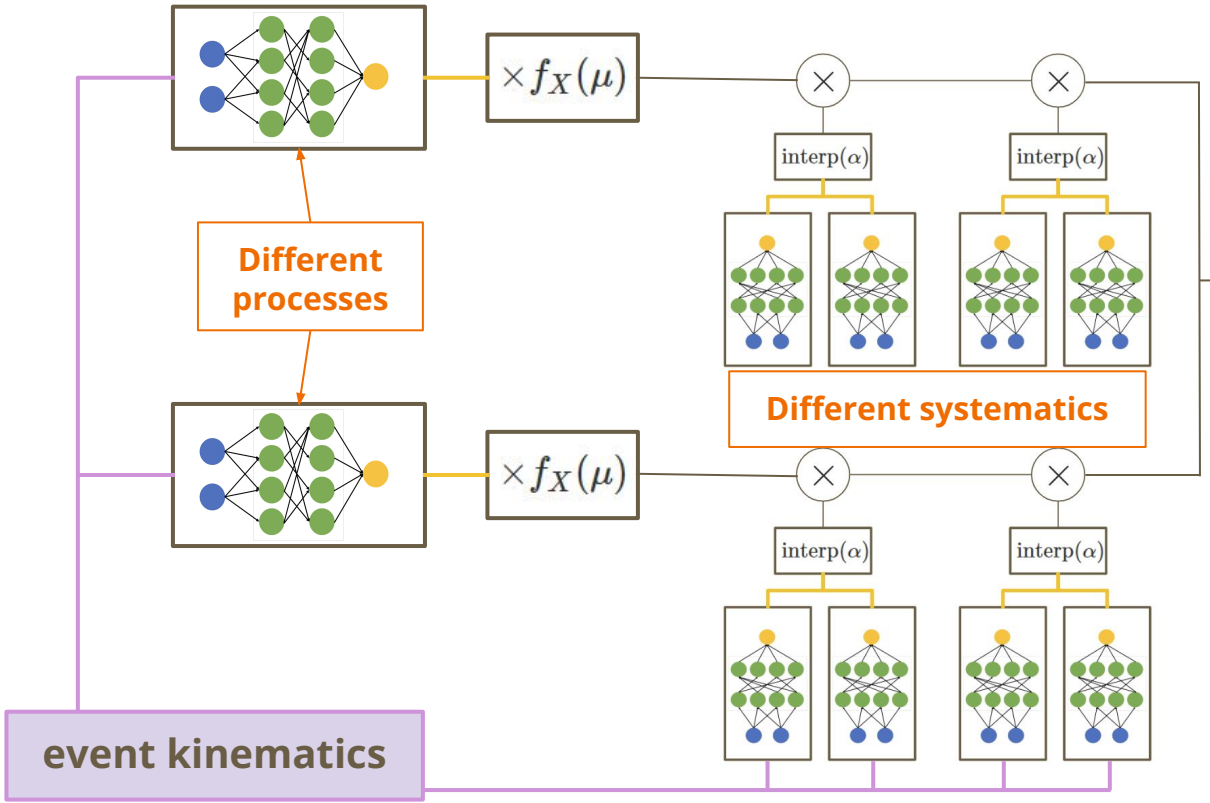
Profiling over ~10M events with ~10k observables



Very nice, but serious computational challenges

Image credit: 1805.00013

Parameter inference: challenges



NN training

- Large number of NN required to describe model and uncertainties
- O(10,000) deep NN each with O(1,000,000) parameters
  - **Between 1B - 10B trainable parameters in total**

Parameter inference

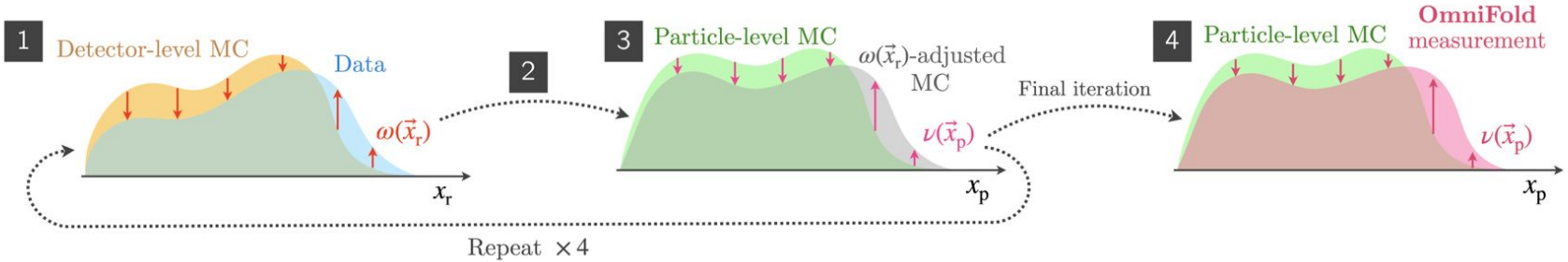
- O(10,000) NN outputs for O(1,000,000) events.
  - **Around 10B parameters (~100GB - 1TB of memory)**
- O(1,000) calls per minimization.
- Tricks / bookkeeping can simplify challenge for several applications

ML-based unbinned unfolding

A simultaneous unbinned differential cross section measurement of twenty-four Z+jets kinematic observables with the ATLAS detector

CDS CERN-EP-2024-132 | arXiv 2405.20041 | DOI 10.5281/zenodo.11507450 | launch binder | Open in Colab

These notebooks demonstrate how to interact with the unbinned, twenty-four-dimensional ATLAS Z+jets differential cross-section measurement presented in arXiv:2405.20041. This analysis uses OmniFold to mitigate detector effects in data.



https://gitlab.cern.ch/atlas-physics/public/sm-z-jets-omnifold-2024

ML-based unbinned unfolding

- Also based on likelihood ratio estimation (same challenge for stability and precision as for parameter estimation).
- The main result is itself, an NN (!)
  - Most (= all before this one) cross sections are presented as **histograms** and can be stored and interacted with on e.g. HEPData
  - Unbinned results **breaks HEPData**; how do you represent an unbinned object?
  - We published the unbinned data, along with Jupyter notebooks for users to interact with. However, it is non-trivial for a typical user - more flexibility/freedom comes with more **computational overhead**!

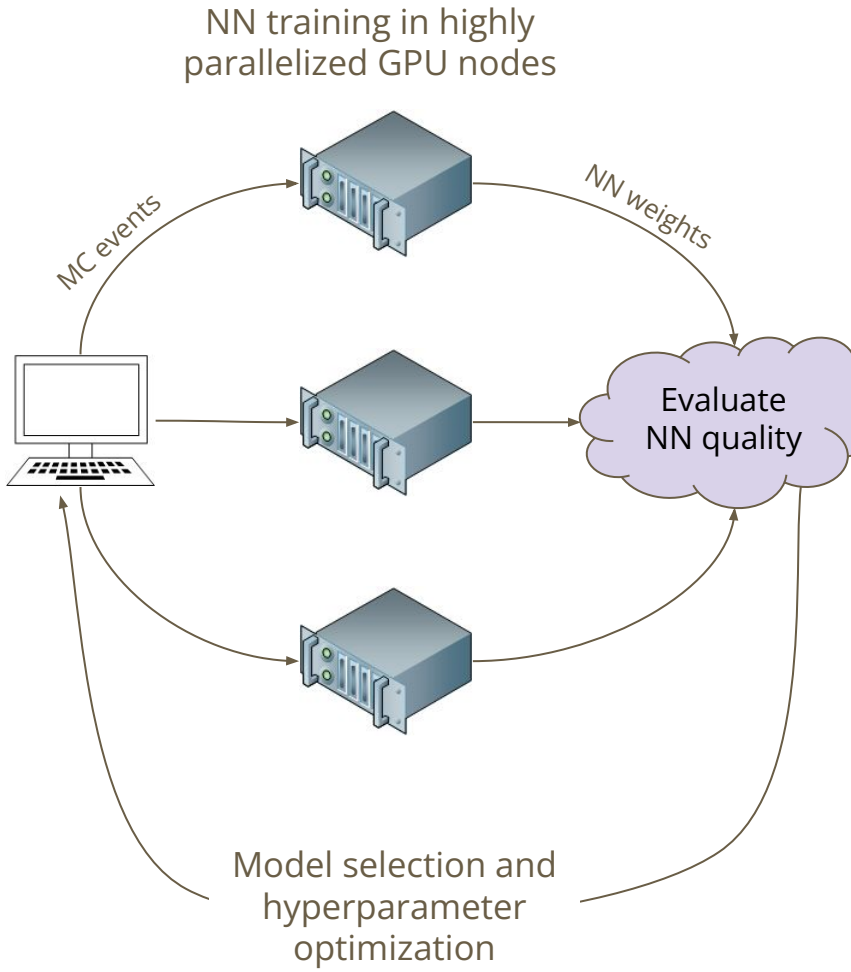
Can HPCs enable interaction with unbinned results? What about if we want to do parameter estimation on the unbinned data? (see previous slides...)

Opportunities with HPC: training

number of NN = (p processes) x (2s systematics)

x (n ensemble)

- Massive parallelization of training
  - Training of individual NN is fast*: O(hours) in modern GPUs
  - Large number of NN required.
- Implementation of metrics for automatic loop
  - Model selection (low bias, low variance, interpolation)
  - Hyperparameter optimization

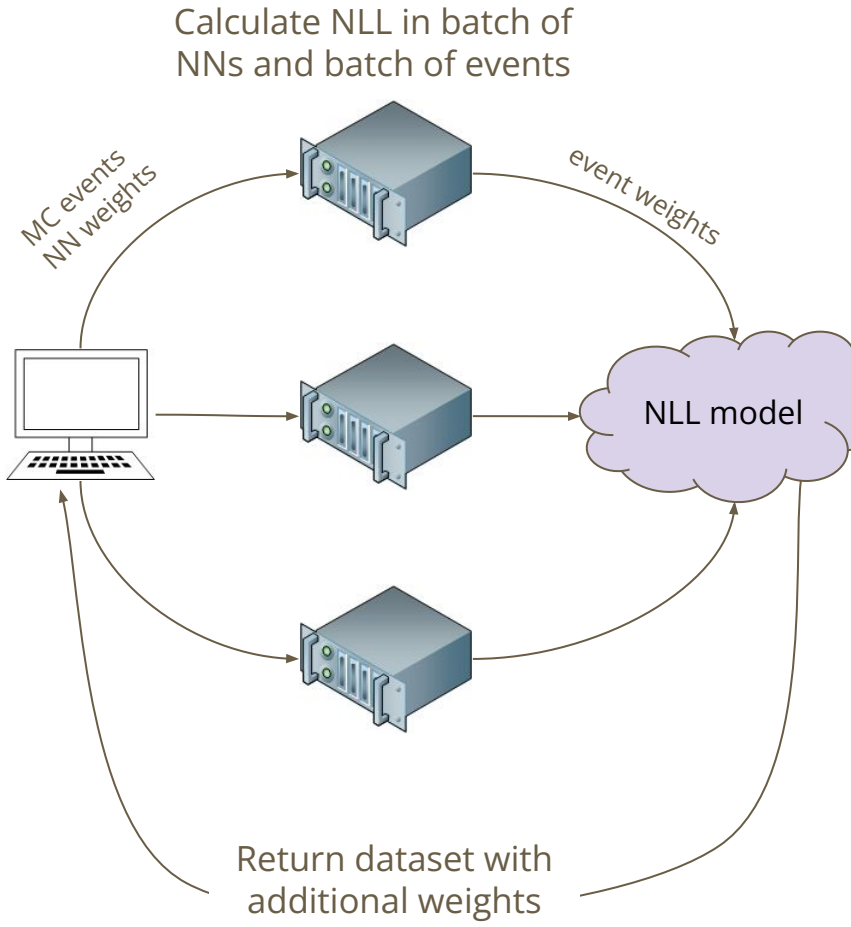


*Becomes more complicated for full phase space (more inclusive selections and/or more features)

Opportunities with HPC: NN evaluation

Parallelize both NNs and event batches

- Possible to generate bootstrap copies from ensembles to create alternative NLL models
- Possible to generate bootstrap copies from events to create alternative datasets (toys)

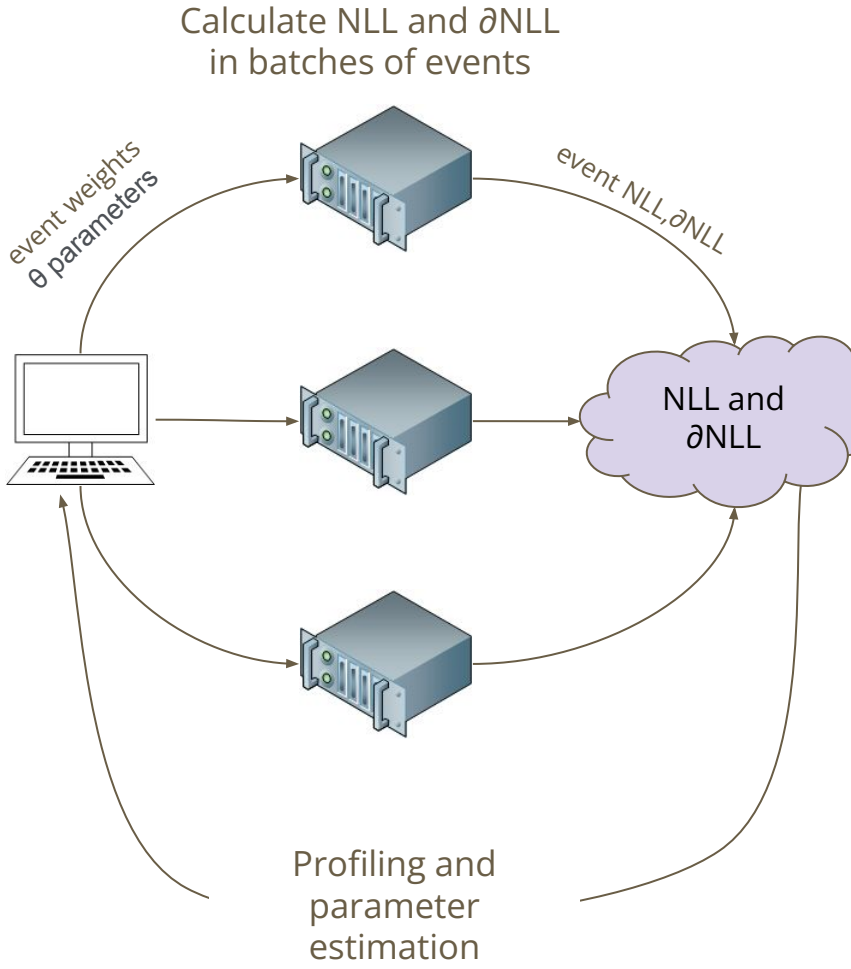


Opportunities with HPC: parameter inference

Very large memory requirements for NLL and ∂NLL calculation

Parallelize event batches to handle large LHC samples

- Possible to evaluate NN on the fly depending on the balance between size of NN and number of events.
- Calculate NLL and ∂NLL in different GPU nodes.
  - Integration in standard HEP tools (RooFit, pyHF, ...)
- Not embarrassingly parallel (unusual in HEP)



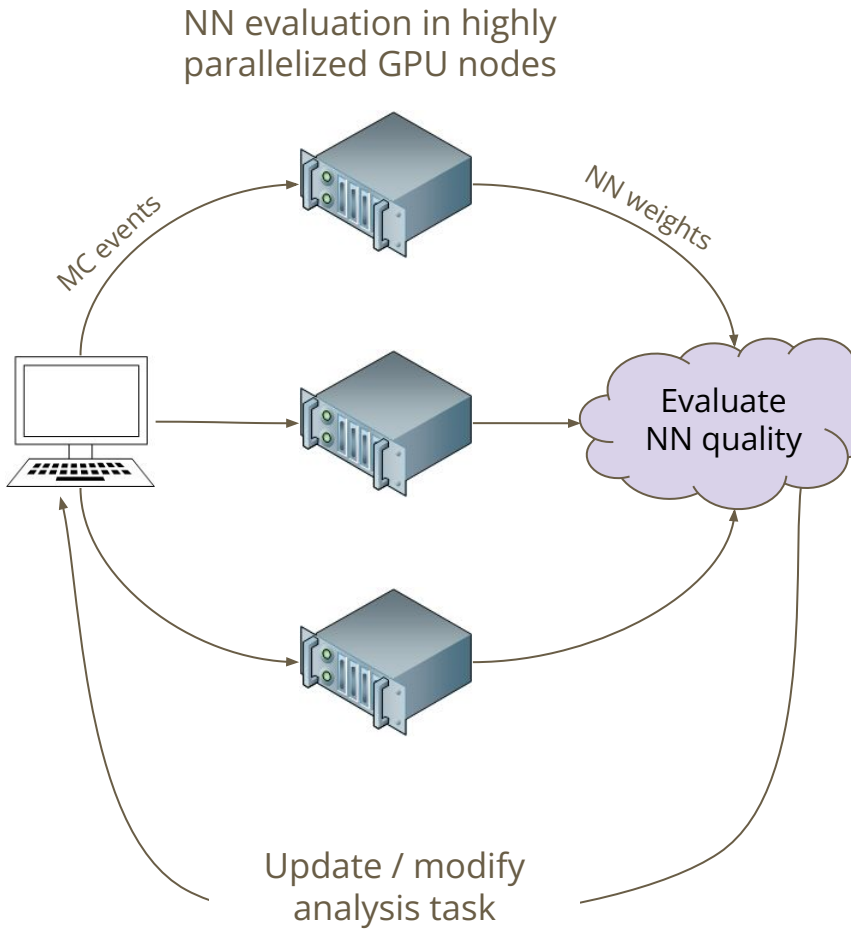
Simpler solutions possible for individual applications with tricks / bookkeeping

Opportunities with HPC: interoperability

The result is itself many NNs. How to interact with it?

- For parameter estimation and anomaly detection, the NNs are collaboration internal (as inference is in data space), but interacting with it is still essential.

- For unfolding, the result is public, but most users won't have the computational power or know-how to interact with it.



Conclusion

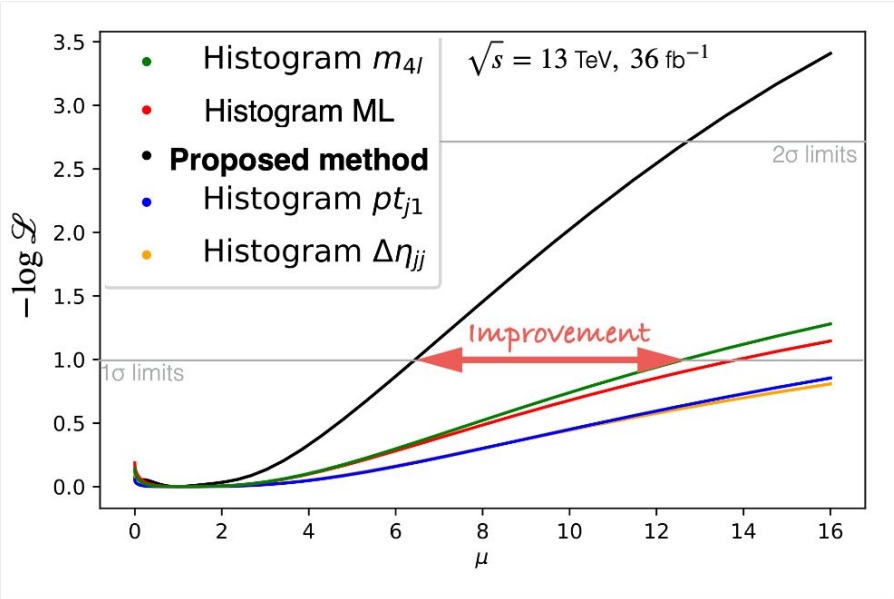
- Exciting new unbinned and high-dimensional analysis methods that provide more power and flexibility

- Methods require more computational infrastructure to use than traditional methods

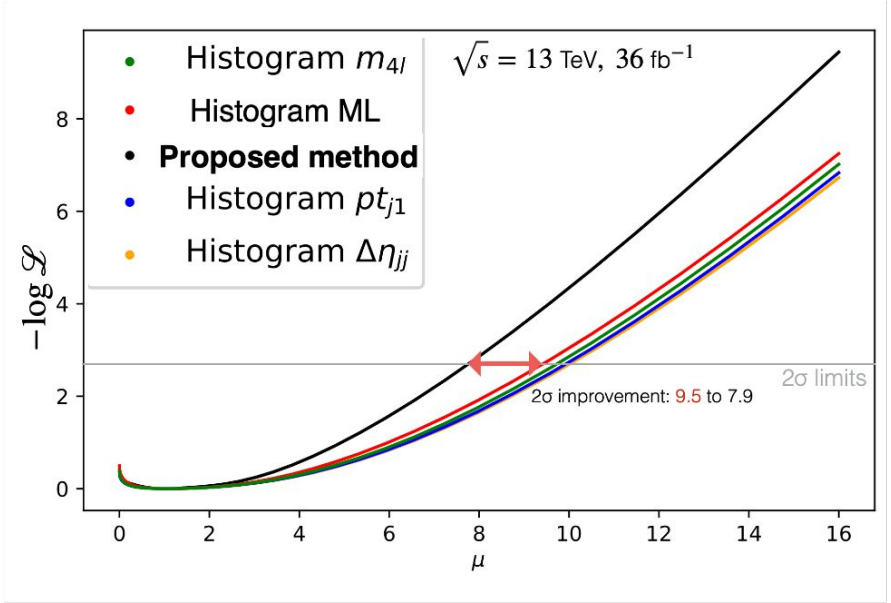
- A need to build such infrastructure as more analyses adopt these methods

The power of NSBI (parameter interference)

hal-02971995v3: **Aishik Ghosh**, David Rousseau



(a) SM, without rate



(b) SM with rate

The power of NSBI (unfolding)

Measure new observables or quantities constructed as a function of the input observables

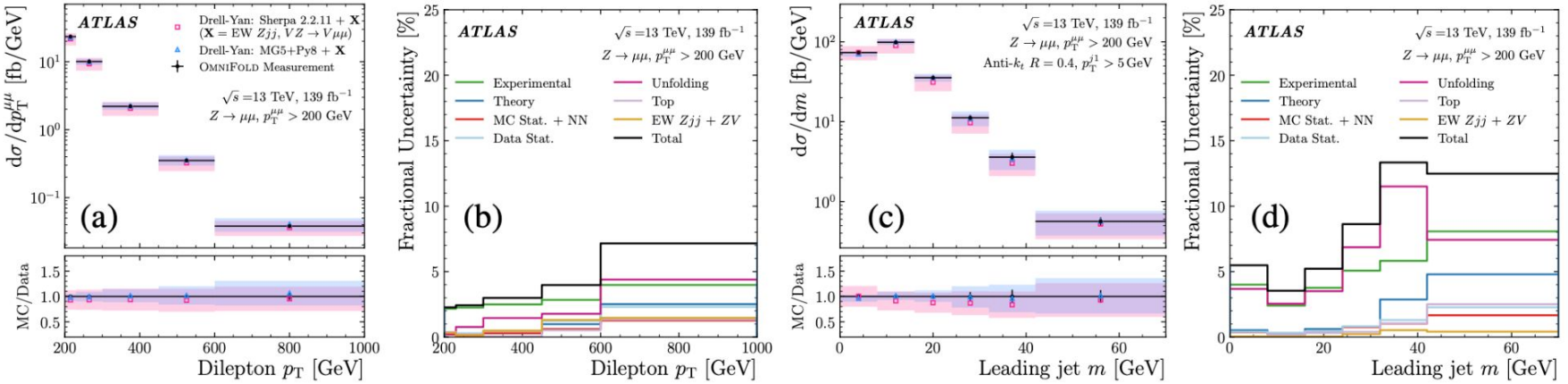


Figure 1: Measured differential cross sections compared with particle-level predictions from SHERPA and MADGRAPH for two of the 24 directly measured observables: (a) $p_{\mathrm{T}}^{\mu\mu}$ with its (b) associated uncertainty breakdown; and, (c) m_{j1} with its (d) associated uncertainty breakdown. For display purposes, binned (marginal) distributions are shown, though the measurement itself is unbinned and 24-dimensional.

https://gitlab.cern.ch/atlas-physics/public/sm-z-jets-omnifold-2024