

MACHINE LEARNING FOR NUCLEAR ASTROPHYSICS (A FIRST ATTEMPT)

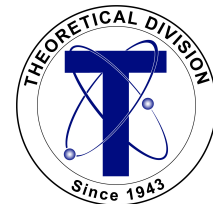


LA-UR-22-24945

MATTHEW MUMPOWER

JINA Frontiers

Friday May 27th 2022



Theoretical Division

ML/AI: TEXT → IMAGE



A bucket bag made of blue suede. The bag is decorated with intricate golden paisley patterns. The handle of the bag is made of rubies and pearls.

ML/AI: TEXT → IMAGE



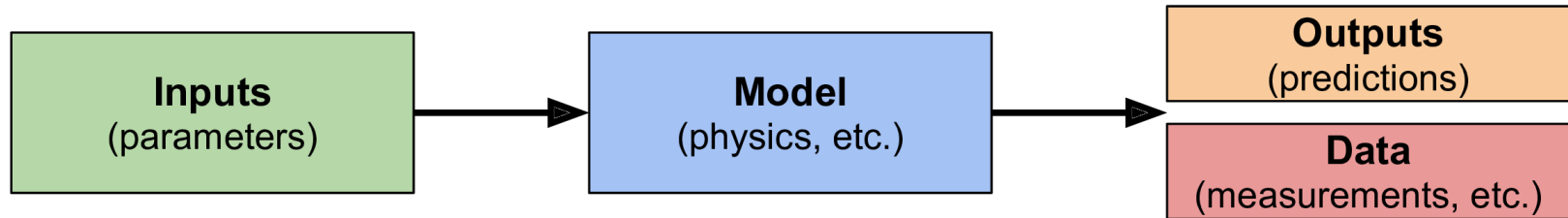
A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

ML/AI: PERSON GENERATION



Every refresh of this page generates a new person (who does not exist)

NUCLEAR PHYSICS FROM A FORWARD PROBLEM PERSPECTIVE



When we model nuclear properties we think of it as a 'forward' problem

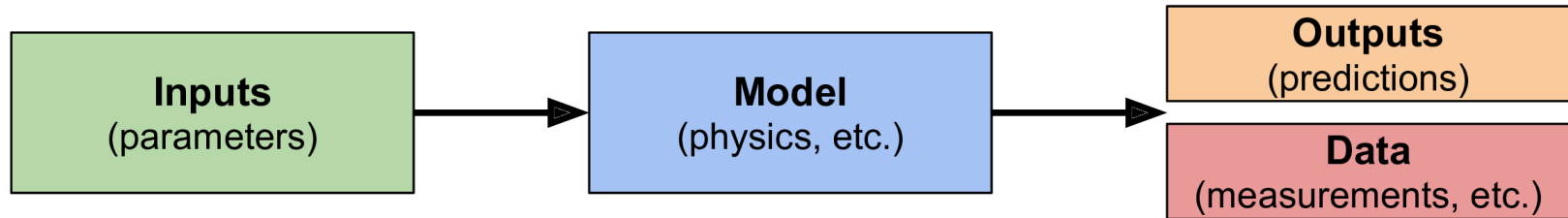
We start with our model and parameters and try to match measurements / observations

This is an extremely successful approach and is generally how we use theory models

$$\text{Mathematically: } f(\vec{x}) = \vec{y}$$

Where f is our model, \vec{x} are the parameters for the model and \vec{y} are the predictions

FORWARD PROBLEM... PROBLEMS...



There can be many challenges with this approach

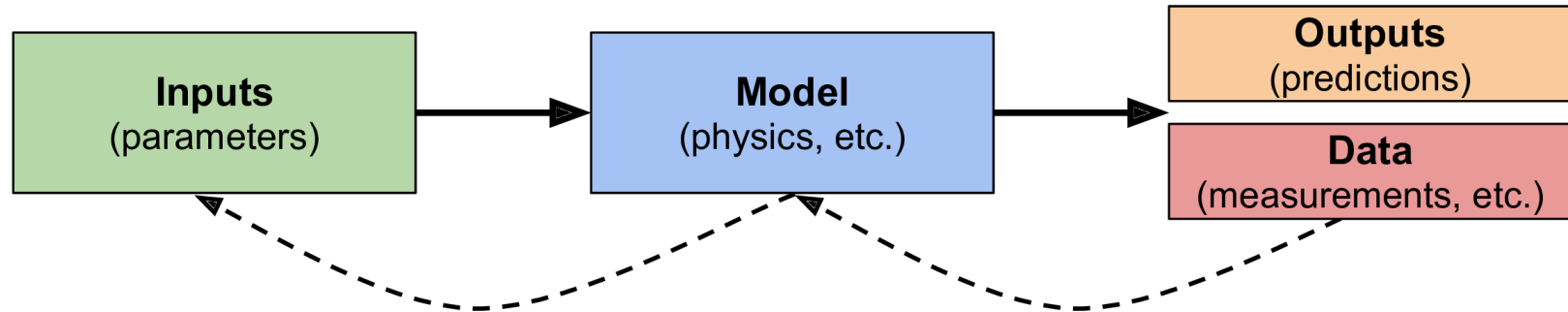
For instance: model computationally expensive or how do we update our model if we don't match data?

Or perhaps...

We don't always know what modifications are required or the physics could be very hard if not impossible to model given current computational limitations (e.g. many-body problem)

Treating this as an 'inverse problem' provides an alternative approach...

NUCLEAR PHYSICS FROM AN INVERSE PROBLEM PERSPECTIVE



Suppose we start with the nuclear data (and its associated uncertainties)

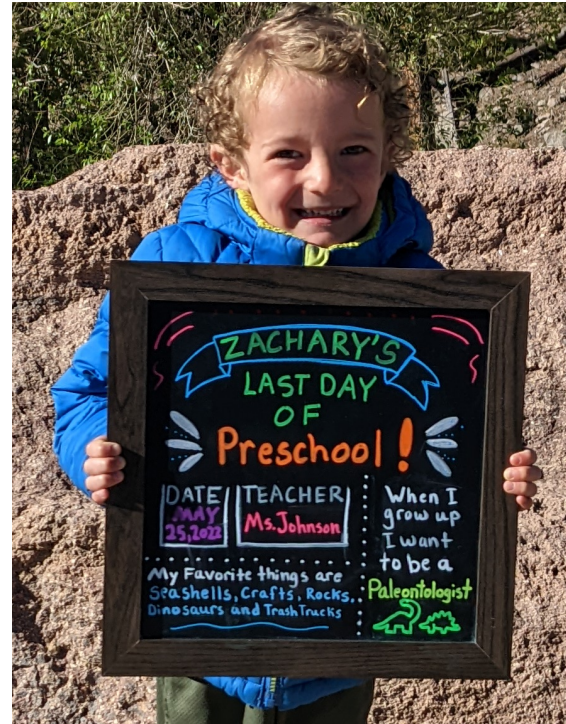
We can then try to figure out what models may match relevant observations

This approach allows us to *vary* and optimize *the model*!

Mathematically, still: $f(\vec{x}) = \vec{y}$; but we want to now find f ; fixing \vec{x}

Machine learning and artificial intelligence algorithms are ideal for this class of problems...

HOW DO WE FIND f ? AN EXAMPLE...



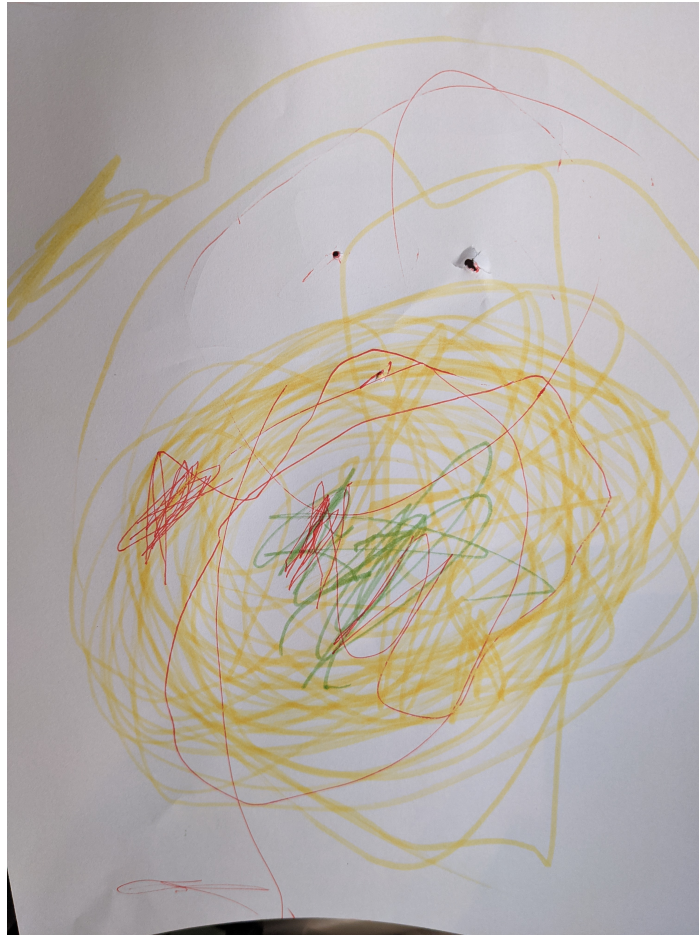
My son Zachary has been very keen on learning his letters from an early age

Letters are well known (this represents the data we want to match to)

But the model (being able to draw - eventually letters) must be developed, practiced and finally optimized

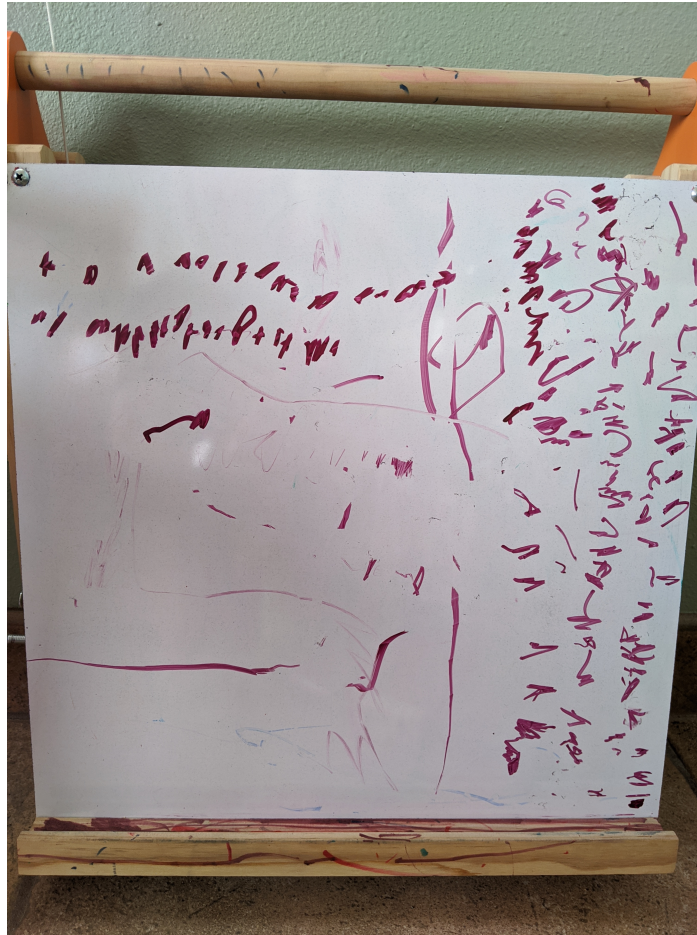
This is a complex, iterative process - it takes times to find ' f '!

FINDING f ... A FIRST ATTEMPT



Example from 1 years old; random drawing episode; time taken: 10 minutes

FINDING *f*... OKAY FOR REAL THIS TIME



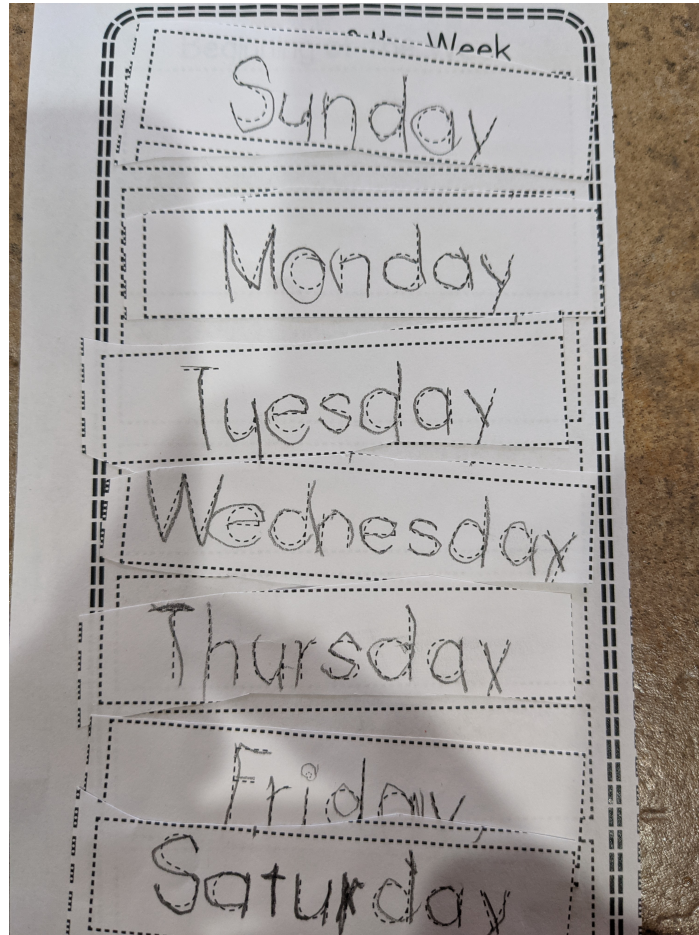
Example from 2.5 years old; first ever attempt to draw letters / alphabet; time taken: 2 hours!

FINDING f ... NOW WE'RE GETTING SOMEWHERE



Example from 3.5 years old; drawing scenery; time taken: 10 minutes

FINDING *f*... SOMETHING LEGIBLE!



Example from 4 years old; tracing the days of the week; time taken: 30 minutes

UNDERSTANDING OF NUCLEAR ASTROPHYSICS REQUIRES MANY NUCLEAR PROPERTIES

Nucleosynthesis masses, β -decay rates, reaction rates & branching ratios



LET'S APPLY THIS IDEA TO A BASIC PROPERTY...

A relatively easy choice is a scalar property; how about **atomic binding energies (masses)**?

But... writing down the many-body quantum mechanical Hamiltonian is hard; want to find ground state

$$\text{Recall: } f(\vec{x}) = \vec{y}$$

f will be a neural network (a model that can change)

\vec{y}_{obs} are the observations (e.g. Atomic Mass Evaluation) we want to match \vec{y} with

But what about \vec{x} ?

Why not a physically motivated feature space!?

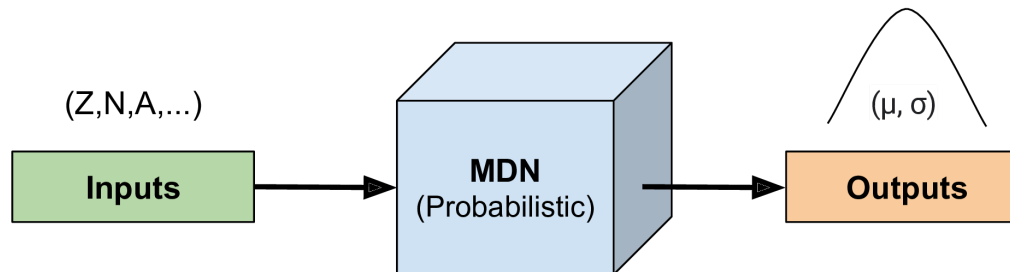
proton number (Z), neutron number, (N), nucleon number (A), etc.

NETWORK FEATURE SPACES (\vec{x})

Model Name	Feature Space	Output
M2	N, Z	δM
M6	$N, Z, A, A^{2/3},$ $Z(Z-1)/A^{1/3}, (N-Z)^2/A$	δM
M8	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$	δM
M10	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$	δM
M12	$\Delta N, \Delta Z$ $N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$ $\Delta N, \Delta Z, N_{\text{shell}}, Z_{\text{shell}}$	δM
MS2	N, Z	$\delta M, S_n$
MS6	$N, Z, A, A^{2/3},$ $Z(Z-1)/A^{1/3}, (N-Z)^2/A$	$\delta M, S_n$
MS8	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$	$\delta M, S_n$
MS10	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$	$\delta M, S_n$
MS12	$\Delta N, \Delta Z$ $N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3},$ $(N-Z)^2/A, Z_{EO}, N_{EO}$ $\Delta N, \Delta Z, N_{\text{shell}}, Z_{\text{shell}}$	$\delta M, S_n$

We use a variety of possible physically motivated features as inputs into the model

MIXTURE DENSITY NETWORK



We take a Bayesian approach: our network inputs are sampled based on nuclear data uncertainties

Our network outputs are therefore **statistical**

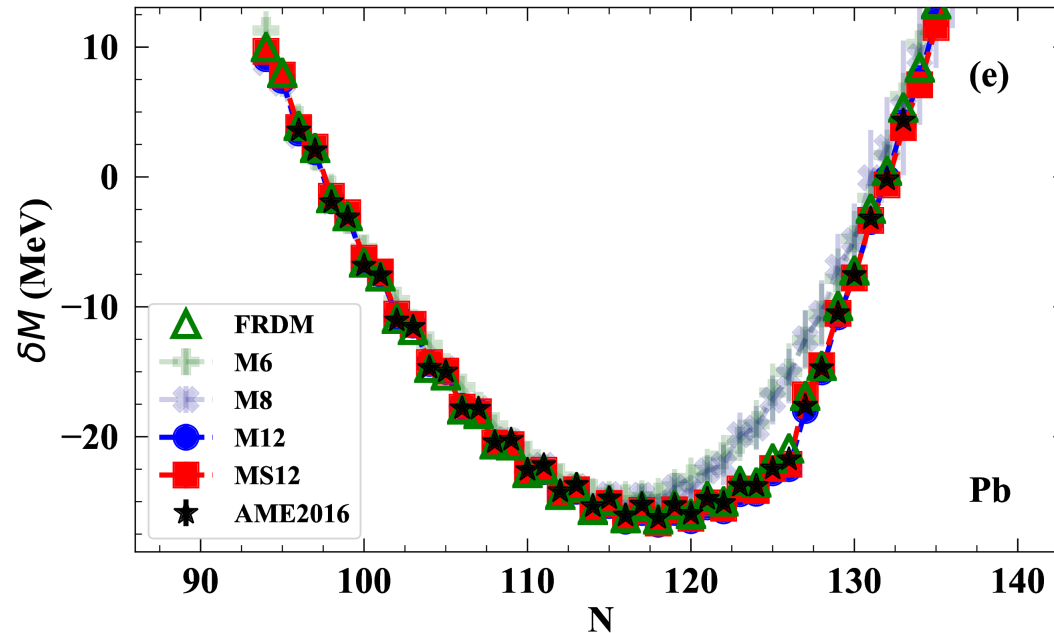
We can represent outputs by a collection of Gaussians (for masses we only need 1)

Our network also provides a well-quantified estimate of uncertainty (fully propagated through the model)

Network topology: fully connected ~ 4 -8 layers; 8 nodes; regularization via weight-decay

Last layer converts output to Gaussian ad-mixture

RESULTS: MODELING ATOMIC MASSES

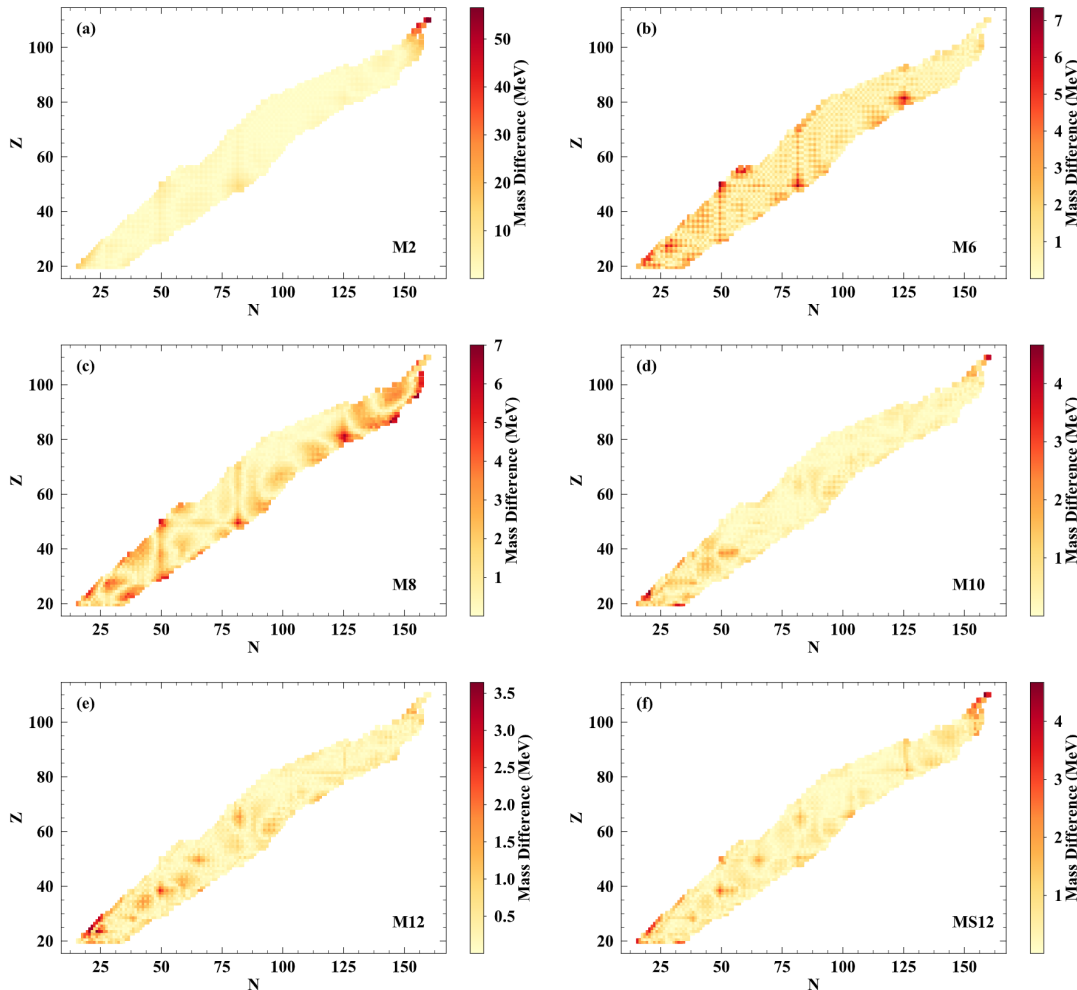


Here we show results along the lead (Pb) [$Z = 82$] isotopic chain

Data from the Atomic Mass Evaluation (AME2016) in black stars

Increasing sophistication of the model feature space indicated by M##

RESULTS: MODELING ATOMIC MASSES



Results across the chart of nuclides with increasing feature space complexity

Note the decrease in the maximum value of the scale

Lovell ([FRIB-TA early career award winner!](#)), Mohan, Sprouse & Mumpower arXiv:2201.00676 (2022)

RESULTS: MODELING ATOMIC MASSES

A summary of results

Model	δM	σ_{RMS} (MeV)	S_n	σ_{RMS} (MeV)
M2		3.90		—
MS2		2.43		1.25
M6		1.57		—
MS6		2.07		1.21
M8		1.66		—
MS8		2.21		0.57
M10		0.58		—
MS10		0.76		0.57
M12		0.56		—
MS12		0.64		0.47

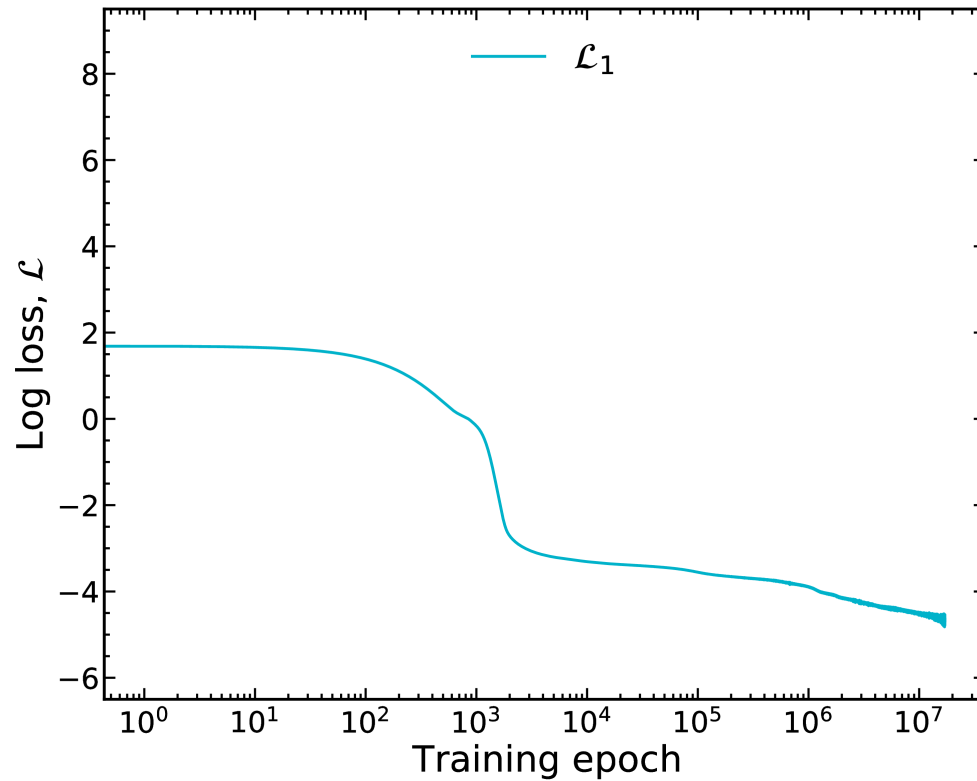
Some lessons learned:

More physics added to the feature space → the better our description of atomic masses

Attempting to fit and predict other mass related quantities (e.g. separation energies) does impact our results

Although marginal compared to choice of feature space

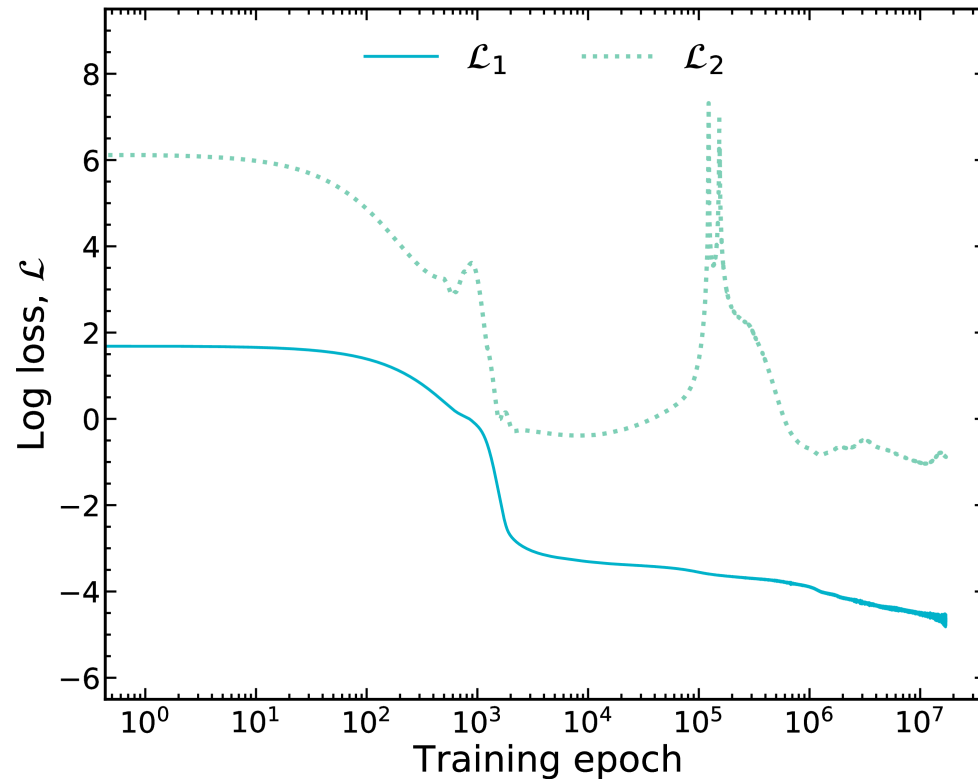
TRAINING ON MASS DATA



We train on mass data using a log-loss, \mathcal{L}_1

Training takes a considerably long time

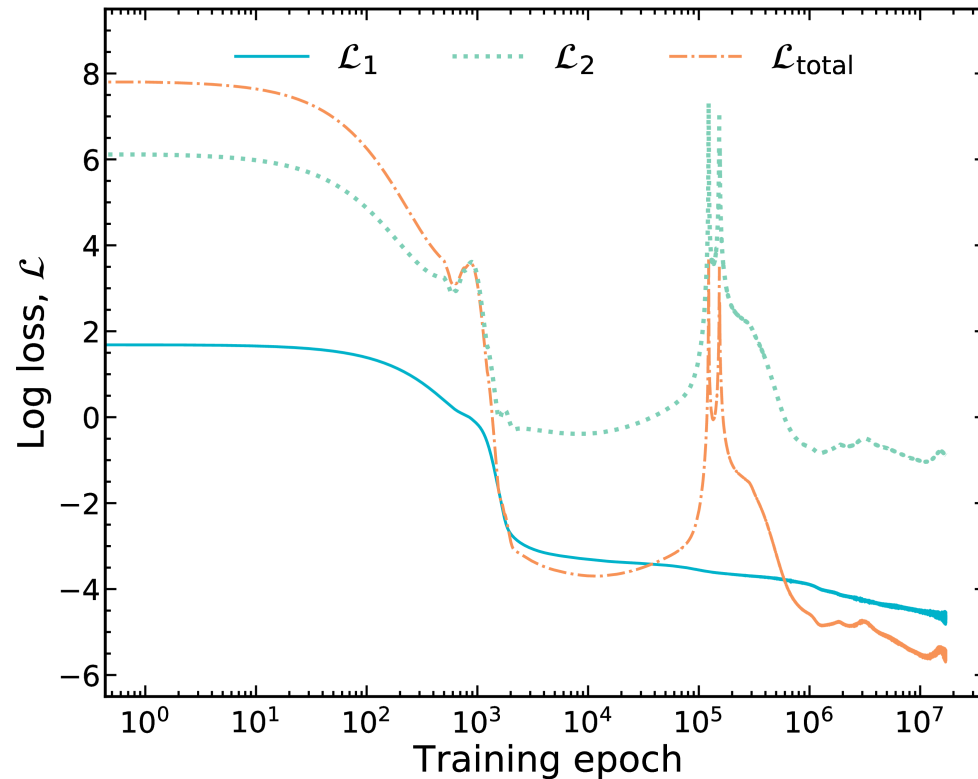
INTRODUCE PHYSICAL CONSTRAINT VIA LOSS FUNCTION



Now we optimize data, \mathcal{L}_1 , **and** introduce a second (physical) constraint, \mathcal{L}_2

We have tried many physics-based losses, this particular one (Garvey-Kelson relations) performs well

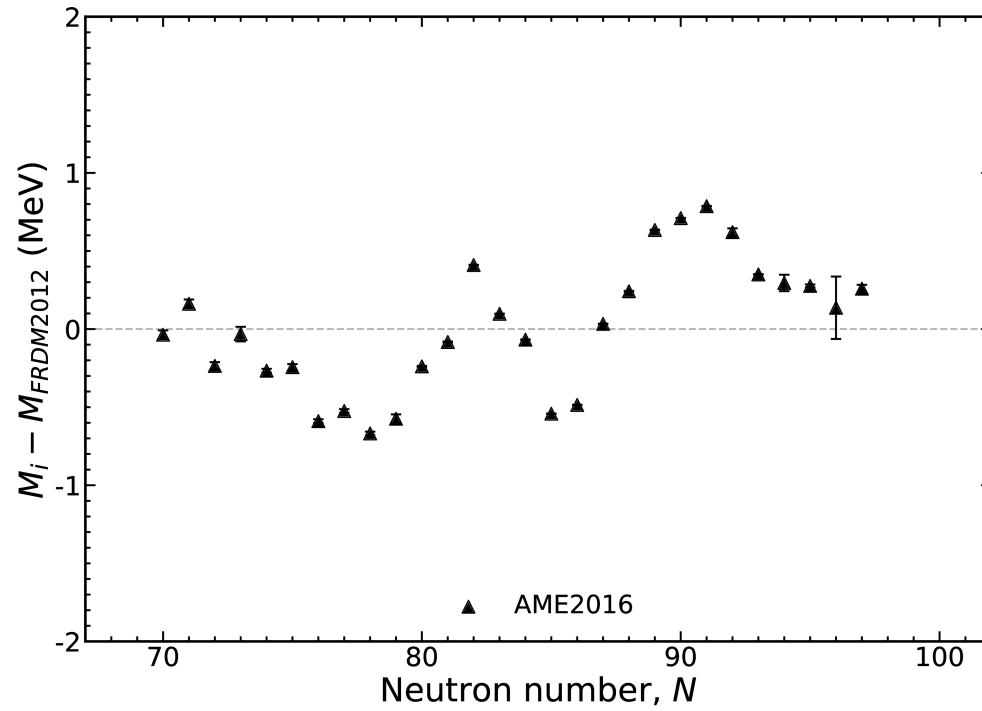
TOTAL LOSS: SUM THE TWO LOSSES



$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \lambda_{\text{phys}} \mathcal{L}_2 ; \lambda_{\text{phys}} = 1 \text{ in this case}$$

Good match to data does not imply good physics (although by happenstance it is in the end in this case)!

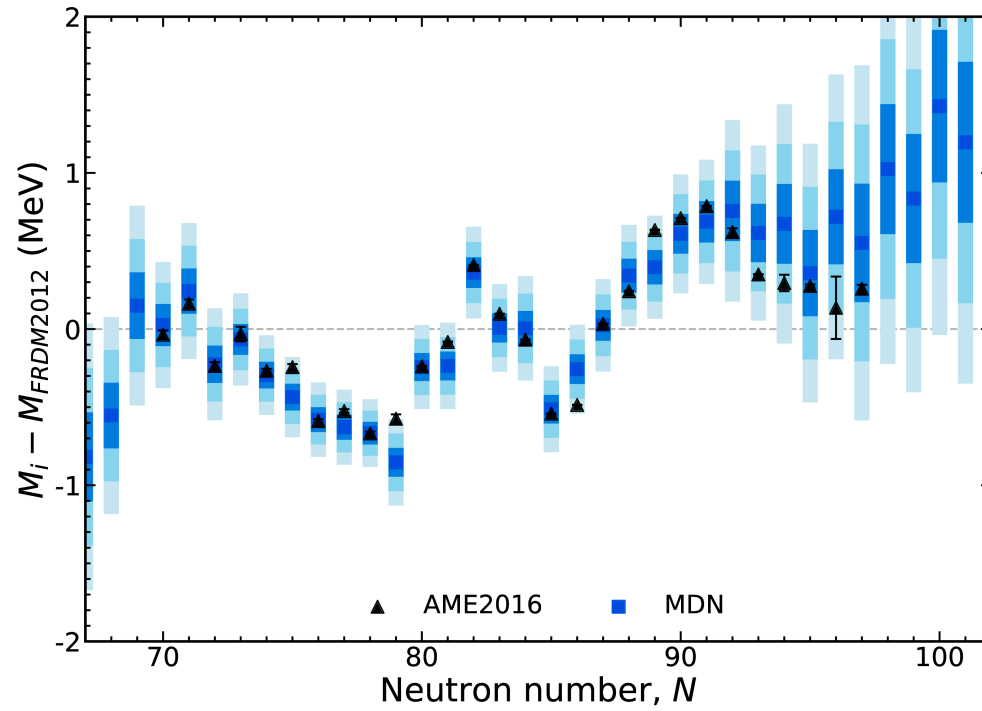
MASS TRENDS ALONG Nd ISOTOPIC CHAIN, $Z = 60$



Results plotted relative to the Los Alamos FRDM model

Modern mass models draw a 'straight line' through data

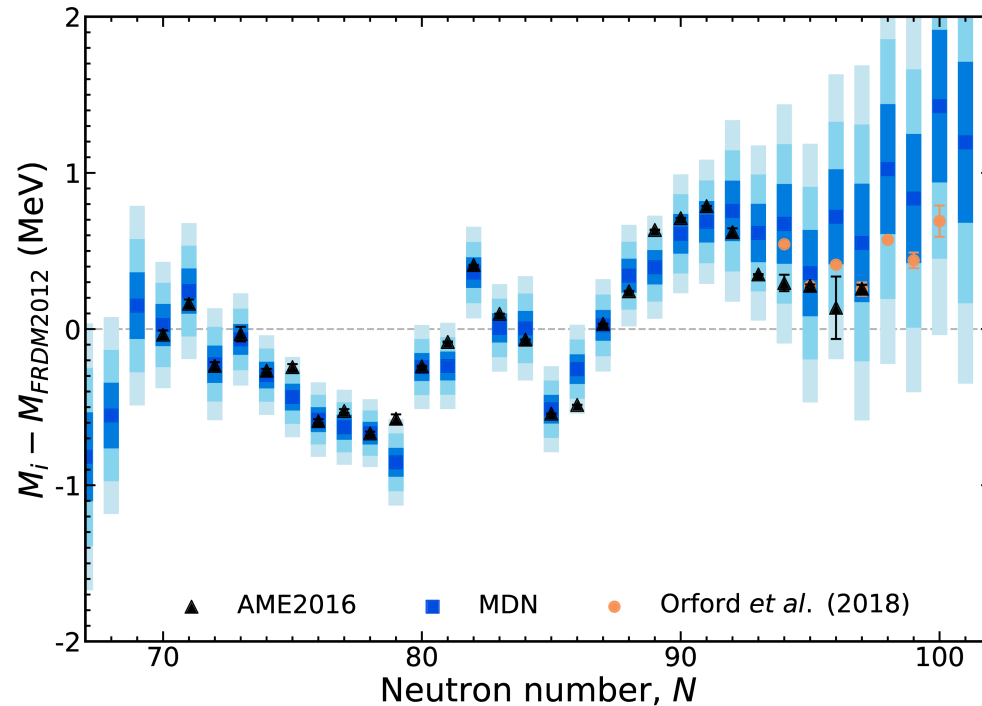
RESULTS OF NEW MODEL Nd, $Z = 60$



Probabilistic network \rightarrow well-defined uncertainties predicted for every nucleus (■ 1σ , ■ 2σ ■ 3σ)

Trends in data are captured far better than any mass model

RESULTS OF NEW MODEL Nd, $Z = 60$

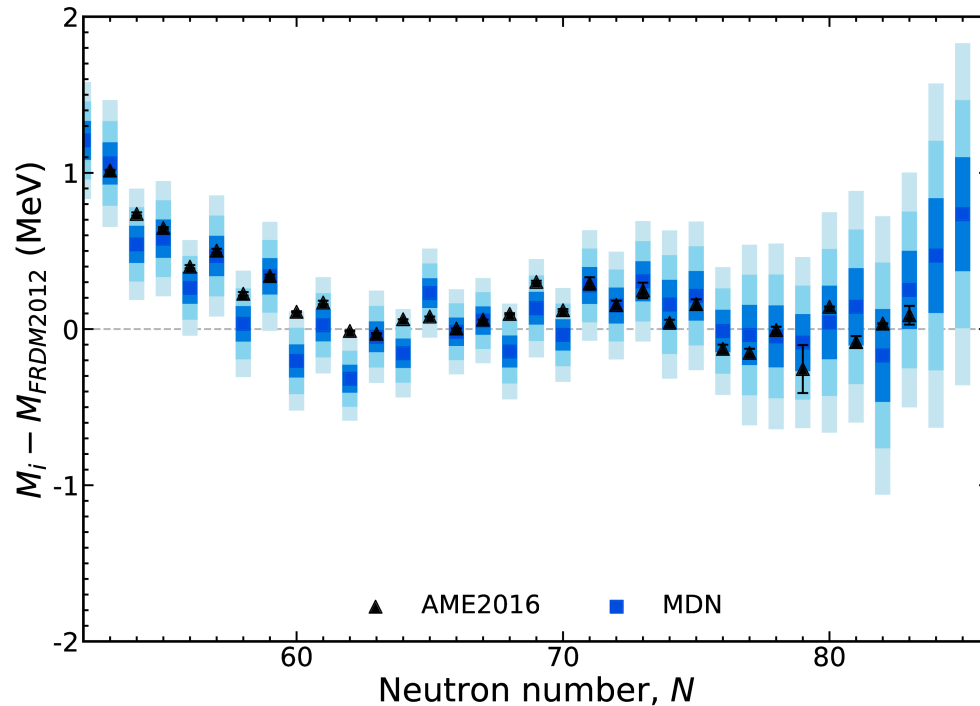


New data (unknown to the model) are in agreement with the MDN model predictions

Model is now trained on only 20% of the AME and predicting 80%!

The training set is random; different datasets perform differently

RESULTS OF NEW MODEL In, $Z = 49$



Our results are not limited to any mass region - all of them are described equally well ($Z \geq 20$)

Results are better than any existing mass model; uncertainties inherently increase with extrapolation

We can predict new measurements and capture trends (important for astrophysics)

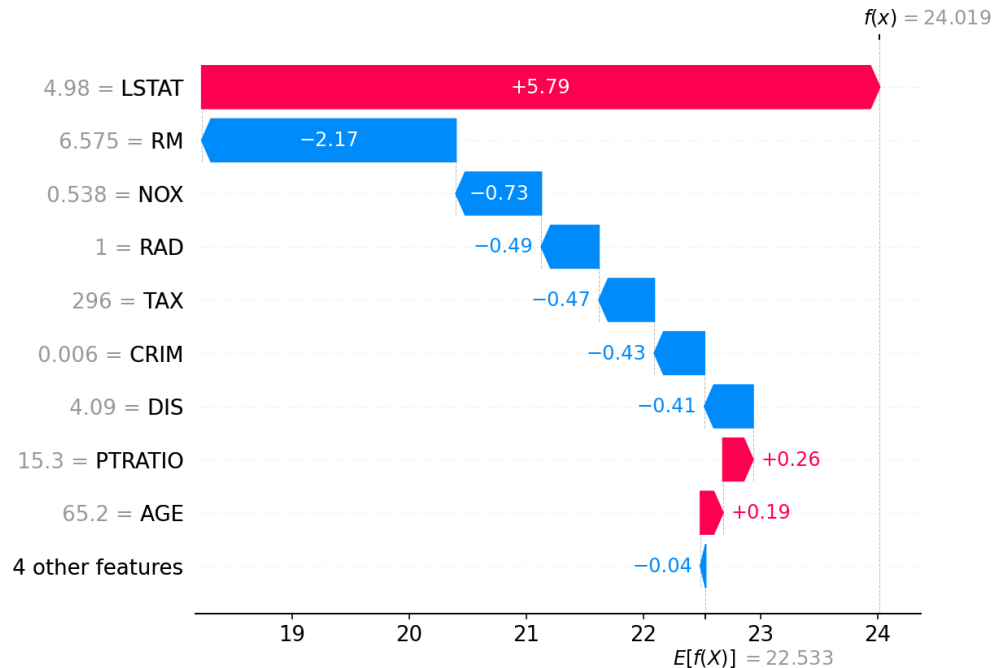
HOW CAN WE INTERPRET THESE RESULTS?

One method is to calculate **Shapley values** (Shapely 1953 • Lundberg & Lee 2017)

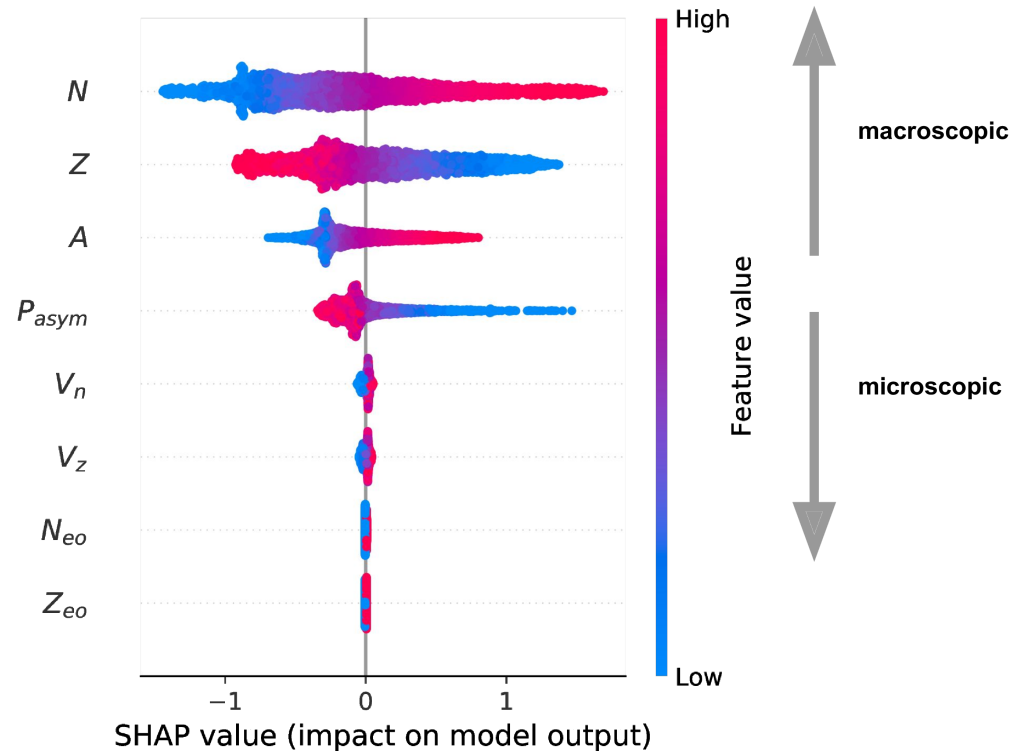
Idea arises in Game Theory when a coalition of players (ML - features) optimize gains relative to costs

The players may contribute unequally, but the payoff is mutually beneficial

Applicable to many applications, ML in particular to understand feature importance



RANKING FEATURE IMPORTANCE



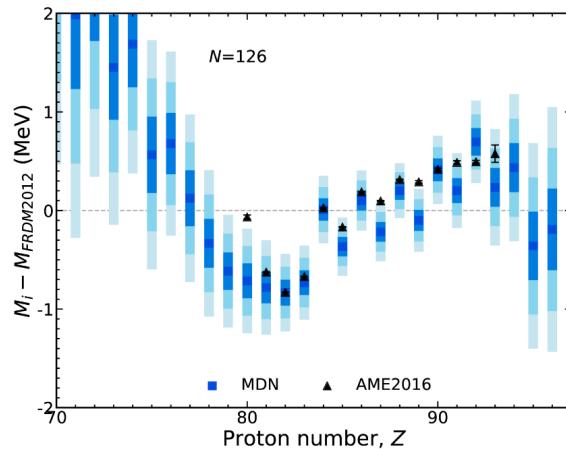
SHAP calculated for all of the AME (2020) [predictions!]

The model behaves like a theoretical mass model

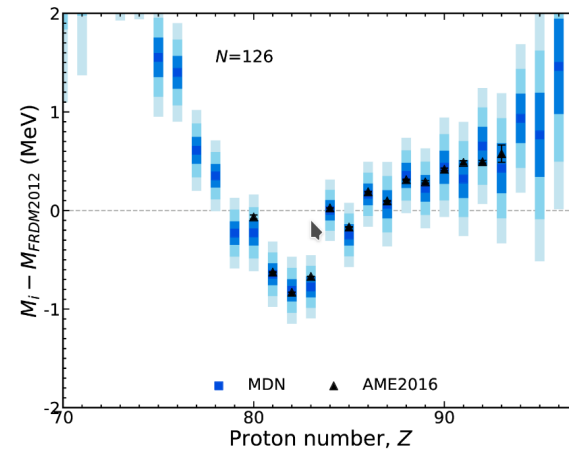
Macroscopic variables most important followed by quantum corrections

WHAT HAPPENS WHEN THERE'S NEW DATA? (GENERALIZATION)

Before



After



The network is able to incorporate this knowledge and updated predictions via Bayes' theorem

New information provided at closed shell $N = 126$

SPECIAL THANKS TO

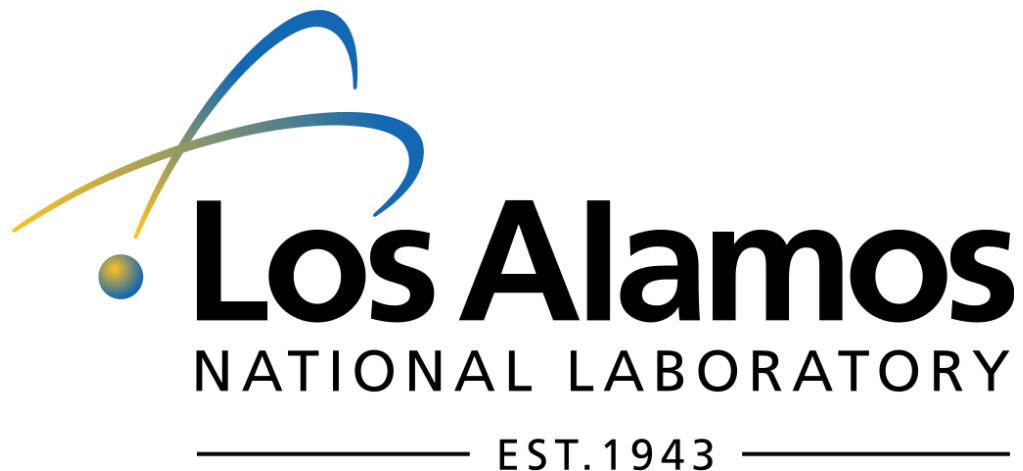
My collaborators

A. Lovell, A. Mohan, & [T. Sprouse](#)

▣ Postdoc



And to my son Zachary for the use of his drawings



LOS ALAMOS NATIONAL LABORATORY CAVEAT

The submitted materials have been authored by an employee or employees of Triad National Security, LLC (Triad) under contract with the U.S. Department of Energy/National Nuclear Security Administration (DOE/NNSA).

Accordingly, the U.S. Government retains an irrevocable, nonexclusive, royaltyfree license to publish, translate, reproduce, use, or dispose of the published form of the work and to authorize others to do the same for U.S. Government purposes.

SUMMARY

We have developed an interpretable neural network capable of predicting masses

Advances:

We can predict masses **accurately** and suss out **trends** in data

Our network is **probabilistic**, each prediction has an associated **uncertainty**

Our model can **extrapolate** and retains the physics

The **quality** of the training data is important (not shown)

We can **classify** the information content of existing and future measurements with this type of modeling

We are now ready to study the impact of this modeling in astrophysics!

Results / Data / Papers @ [MatthewMumpower.com](https://matthewmumpower.com)

