

Nuclear masses learned from a probabilistic neural networkA. E. Lovell ^{1,*}, A. T. Mohan,² T. M. Sprouse,¹ and M. R. Mumpower ¹¹*Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*²*Computational Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

(Received 15 September 2021; accepted 8 June 2022; published 13 July 2022)

Machine learning methods and uncertainty quantification have been gaining interest throughout the last several years in low-energy nuclear physics. In particular, Gaussian processes and Bayesian neural networks have increasingly been applied to improve mass model predictions while providing well-quantified uncertainties. In this work, we use the probabilistic Mixture Density Network (MDN) to directly predict the mass excess of the 2016 Atomic Mass Evaluation within the range of measured data, and we extrapolate the inferred models beyond available experimental data. The MDN provides not only mean values but also full posterior distributions both within the training set and extrapolated testing set. We show that the addition of physical information to the feature space increases the accuracy of the match to the training data as well as provides for more physically meaningful extrapolations beyond the limits of experimental data.

DOI: [10.1103/PhysRevC.106.014305](https://doi.org/10.1103/PhysRevC.106.014305)**I. INTRODUCTION**

The mass of a nucleus is one of the most basic properties that it possesses. Predicting this key quantity for the many thousands of atomic nuclei that exist in nature represents a major hurdle for theoretical nuclear physics modeling. Many applications such as those in astrophysics [1–4], reaction modeling [5,6], and nuclear data evaluations [7] are sensitive to the accurate description of nuclear masses.

Several theoretical approaches afford the prediction of nuclear masses. Phenomenological models like the liquid drop model combine experimental observations with physically interpreted parameters and are successful in predicting bulk properties across the nuclear chart. More advanced mass formulas, such as the Duflo-Zuker model, incorporate microscopic effects via analytic expressions [8,9]. A more detailed picture can be provided by macroscopic-microscopic models which tack on a microscopic component to the bulk liquid drop terms, in the form of shell and pairing corrections [10]. Such modeling has proven to be robust in predicting the properties of atomic nuclei, offering the best accuracy of masses at the present time [11,12]. Recent advances in computation open the opportunity for a more microscopic picture of the nucleus, as with the approach of using energy density functionals which are capable of predicting many properties beyond masses [13,14]. At this time, to achieve accuracy on the order of macroscopic-microscopic models, fully microscopic models rely on the addition of phenomenological parameters [15–17].

Over the past several years, quantifying uncertainties on nuclear models has become a focus of the community (e.g., Ref. [18]). There has been a large push to understand un-

certainties coming from parameter optimization and model truncation in nuclear theory, from *ab initio* methods [19,20] to few-body methods [21–25] and across effective field theories [26–30] and density functional theories [31,32]. Much of the focus has been in moving the field from standard χ^2 minimization and covariance propagation to Bayesian methods. However, there has also been significant effort put into emulators such as Gaussian processes (GPs) [33,34], Bayesian neural networks (BNNs) [35–37], a committee of neural networks [38], and most recently eigenvector continuation [20,39–43] for speeding up parameter optimization, improving the accuracy of nuclear models, and uncertainty quantification. In particular, there have been significant developments surrounding the application of GPs and BNNs to improving nuclear mass predictions [44–47].

Model calibration, or the fitting of model parameters to data, is an important aspect of this type of research. In [44–47], the GP and BNN emulators were trained on the differences between the Atomic Mass Evaluation and various mass models (the residuals). These trained residuals were used to “correct” each of the mass models to lie within the range of the Atomic Mass Evaluation, as well provide uncertainties on the theoretical predictions. However, there are some challenges using both the BNN and GP. Neufcourt *et al.* [44] found that the GP is much more stable than the BNN, in particular when the size of the training set available for the neural network training is limited. In addition, they note that by using statistical arguments, the size of the neural network in the BNN cannot exceed certain dimensions due, again, to the size of the training set and it is further limited by the number of free parameters given available data. (However, some studies, e.g., Ref. [48], show that as the complexity of a neural network increases, it will first perform worse and then perform better and that there are certain regimes where increasing the number of training points worsens the performance.) The GP also

*lovell@lanl.gov

has some downsides, the major one being that its predictions return to the mean value after one or two correlation lengths, so there is no straightforward way to extrapolate the residual corrections. However, it is worthwhile to note that one must be careful any time that an extrapolation is performed, regardless of the machine learning method used.

In this work, we take a different approach than either the GP or BNN and instead use a probabilistic machine learning technique, the Mixture Density Network (MDN) [49]. Unlike most neural networks which directly map the input feature space to a deterministic output space, the MDN describes the output as a mixture of Gaussian functions and uses the neural network to learn the weights, means, and standard deviations of these distributions. The Gaussian distribution gives not only mean values but also uncertainties associated with each prediction, and the mixture allows full posterior distributions to be predicted. This method allows us to take into account discrepant values in the training set without having to simply average them. We explored the ability of the MDN to reliably propagate uncertainties on fission fragment mass yields from the training set to predictions in Ref. [50] for both interpolations within the training set and extrapolations beyond it. The uncertainties were found to reliably propagate from the training set to the predictions. It was found that very small experimental uncertainties became about 10% larger in the predictions compared to the training set, but this deviation was caused mainly by the several orders of magnitude that the fission yields span. This challenge is completely mitigated in the current work because the mass excesses do not cover multiple orders of magnitude.

This paper is divided into the following sections: in Sec. II, we discuss the details of the MDN, the training and testing sets, as well as the neural network architecture; in Sec. III, we discuss the results of the various training sets that we investigated and how adding more physics information into the feature space improves the training and extrapolation; finally, we conclude and describe some future directions for this work in Sec. IV.

II. THEORY

In this section, we describe our choice of probabilistic neural network, the MDN, our training and testing sets, and the neural network architecture.

A. Mixture Density Network (MDN)

In a standard feed-forward neural network (NN), the goal is to optimize a complex, nonlinear mapping between an input feature space \mathbf{x} and an output space \mathbf{y} though $\mathbf{y} = f(\mathbf{x})$. This mapping is typically deterministic—in that stochasticity is not taken into account—and these networks tend to be incapable of well reproducing probabilistic training sets. To mitigate these challenges, and in particular to take uncertainties on the training set into account, we employ a probabilistic machine learning technique, the Mixture Density Network [49]. Instead of using the NN to directly map the input to the output of the training set, the MDN describes the output as a mixture

of Gaussian functions,

$$\mathbf{y}(\mathbf{x}) = \sum_{i=1}^m \alpha_i(\mathbf{x}) \mathcal{N}[\mu_i(\mathbf{x}), \sigma_i(\mathbf{x})], \quad (1)$$

where \mathcal{N} is the normal distribution and the weights, means, and standard deviations, $\alpha_i(\mathbf{x})$, $\mu_i(\mathbf{x})$, and $\sigma_i(\mathbf{x})$, are learned by the standard feed-forward NN. The loss function changes from a simple mean-squares estimate to a log-likelihood loss,

$$\mathcal{L} = -\ln \left[\sum_{i=1}^m \frac{\alpha_i(\mathbf{x})}{(2\pi)^{m/2} \sigma_i(\mathbf{x})} \exp \left\{ -\frac{\|\mathbf{t} - \mu_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2} \right\} \right], \quad (2)$$

where \mathbf{t} is the vector of training outputs and m is the total number of Gaussian mixtures. Instead of minimizing the absolute root-mean-square error, the loss function of Eq. (2) minimizes the difference in the probability distributions between the true distribution of the training set and the predicted posterior distribution.

To better represent probabilistic data sets than traditional deterministic neural networks, the MDN has the additional benefit of predicting the exact posterior distribution of each predicted value instead of simply predicting a mean and a standard deviation—then leaving it to the user to make a decision on the shape of the posterior distribution. The mixture of Gaussians also takes into account discrepant points in the training set (one such example is shown in Ref. [50]), without just taking a weighted average, as the MDN has the flexibility to form a joint distribution over the discrepant sets. More details about the MDN can be found in Ref. [49].

Our MDN is written in PYTORCH [51] and can be run on both CPU and GPU, with a significant speedup when using the GPU implementation due to the parallel nature of the computational workload. The hyperparameters that set the number of Gaussian mixtures in the MDN and the number of layers and nodes in the feed-forward neural network can be specified for each run without any change in the source code. In addition, the implementation is generalized to allow for any number of features in the input vector and for the prediction of any number of outputs, which is explored in this work. In the current implementation, the number of Gaussian mixtures is constrained to be the same for each output; however, this could be easily generalized.

B. The training and testing sets

While the goal of this work is to directly predict the mass excess, δM , instead of predicting a correction to any one specific mass model, we primarily focus on how the features included in the input space lead to better predictions within the training set and when extrapolating beyond it. To that end, there are several different training sets that we define for this work, which we will describe in this section, each successive one adding more information about the nuclei with the goal to decrease the discrepancy between the predicted values and the experimentally evaluated mass excesses. In addition, we also explore how constraining multiple outputs improves the training, particularly predicting both the mass excess and one-neutron separation energy, S_n . To clearly denote these distinctions, we indicate mass predictions only with

TABLE I. Description of the training sets used in this work. Additional details can be found in the text.

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	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3}, (N-Z)^2/A, Z_{EO}, N_{EO}, \Delta N, \Delta Z$	δM
M12	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3}, (N-Z)^2/A, Z_{EO}, N_{EO}, \Delta N, \Delta Z, N_{\text{shell}}, Z_{\text{shell}}$	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	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3}, (N-Z)^2/A, Z_{EO}, N_{EO}, \Delta N, \Delta Z$	$\delta M, S_n$
MS12	$N, Z, A, A^{2/3}, Z(Z-1)/A^{1/3}, (N-Z)^2/A, Z_{EO}, N_{EO}, \Delta N, \Delta Z, N_{\text{shell}}, Z_{\text{shell}}$	$\delta M, S_n$

M, both mass and separation energy predictions with MS, and the number that follows indicates the length of the feature space.

In our base model, M2, the input feature space consists of just the number of neutrons, N , and the number of protons, Z , and we are only predicting the mass excess. The inputs for model M6 take inspiration from the standard semiempirical mass model [52]. In addition to N and Z , we include other bulk properties in the feature space: the mass number, A , and $A^{2/3}$ (from the volume and surface terms), $Z(Z-1)/A^{1/3}$ from the Coulomb term, and $(N-Z)^2/A$ from the asymmetry term. In addition, M8 also includes pairing information, Z_{EO} and N_{EO} where Z_{EO} (N_{EO}) is zero if Z (N) is even and 1 if Z (N) is odd. Then M10 and M12 include information about magic numbers and the nuclear shells. M10 adds ΔN and ΔZ to the features of M8, which are defined as the number of neutrons or protons on top of a magic number or away from a magic number (whichever is smaller), i.e., the valance nucleons. For M12, N_{shell} and Z_{shell} are the shell model orbitals of the last neutron and proton, starting counting at zero, as defined in, e.g., Ref. [53]. We additionally construct an equivalent MS training set for each feature space. An overview of the MDN models can be found in Table I.

We construct our training sets from the 2016 Atomic Mass Evaluation (AME2016). For each isotope where the separation energy can be defined (e.g., where the isotope with $N-1$ neutrons has also been measured, 2074 nuclei), 25 values of δM are sampled from a normal distribution, $\mathcal{N}(\delta M(N, Z), \Delta M(N, Z))$, where $\Delta M(N, Z)$ is the experimentally evaluated uncertainty on $\delta M(N, Z)$. The same is used to calculate $S_n(N, Z)$ except that the masses are sam-

pled from both $\mathcal{N}(\delta M(N, Z), \Delta M(N, Z))$ and $\mathcal{N}(\delta M(N-1, Z), \Delta M(N-1, Z))$ with

$$S_n(N, Z) = \Delta_n - \delta M(N, Z) + \delta M(N-1, Z), \quad (3)$$

where $\Delta_n = 8.0713 \text{ MeV}/c^2$ is the neutron mass excess. Sampling from the distribution on both nuclei allows us to include an uncertainty on the one-neutron separation energy that comes from the evaluated uncertainty on both masses that are needed to calculate this quantity.

The testing set consists of all of the nuclei predicted by the 2012 version of the finite-range droplet model (FRDM) [10], 8668 nuclei. This set of nuclei was chosen to span a large range in N and Z , beyond the experimentally measured nuclei. The FRDM predictions in unmeasured regions of the chart have been consistent over the years, and therefore, a comparison of the MDN predictions to FRDM is representative of the quality of the extrapolation.

Training ensues with the minimization of the loss function defined in Eq. (2). After training, we quantify the goodness of fit in the standard way by calculating the average standard deviation between the MDN prediction and the quantity of interest through the σ_{RMS} . For example, σ_{RMS} between the predicted MDN mass excess, δM^{MDN} , and the AME mass excess, δM^{AME} , is defined as

$$\sigma_{\text{RMS}} = \sqrt{\frac{\sum_i (\delta M_i^{\text{AME}} - \delta M_i^{\text{MDN}})^2}{K}}, \quad (4)$$

where K is the total number of nuclei that were included in the prediction. The same quantity can also be defined for S_n .

C. Network architecture

In this work, we are investigating how to improve network predictions by including more physics in the input feature space. Although other studies (e.g., Ref. [54]) have shown that deeper networks are better able to capture complexities in the data sets of interest compared to NNs with a single layer, deeper networks do not *a priori* provide any insight into why a model is or is not training well. More complex networks also do not lead to any new understanding of the underlying physics, even though they are exceedingly useful for building emulators. To that purpose, we have constructed the neural network to be as small as possible while still providing very good agreement between the training set and the MDN predictions (what is meant by good agreement will be discussed as the results are presented). To that end, our network consists of a single hidden layer with only six nodes. As in previous work [50], because the training set was pulled from a single Gaussian distribution at each mass, our MDN consists of only a single Gaussian mixture. The MDN predictions do not change significantly when multiple Gaussians are used—up to some point when the training becomes unstable. When multiple Gaussians are used, only one Gaussian has a nonzero weight for each mass, which further supports our decision to include only a single Gaussian in our MDN.

Our MDN in PYTORCH uses an Adam optimizer with a learning rate of 0.001 (the default), with a hyperbolic tangent as the activation function. The weights of the NN are

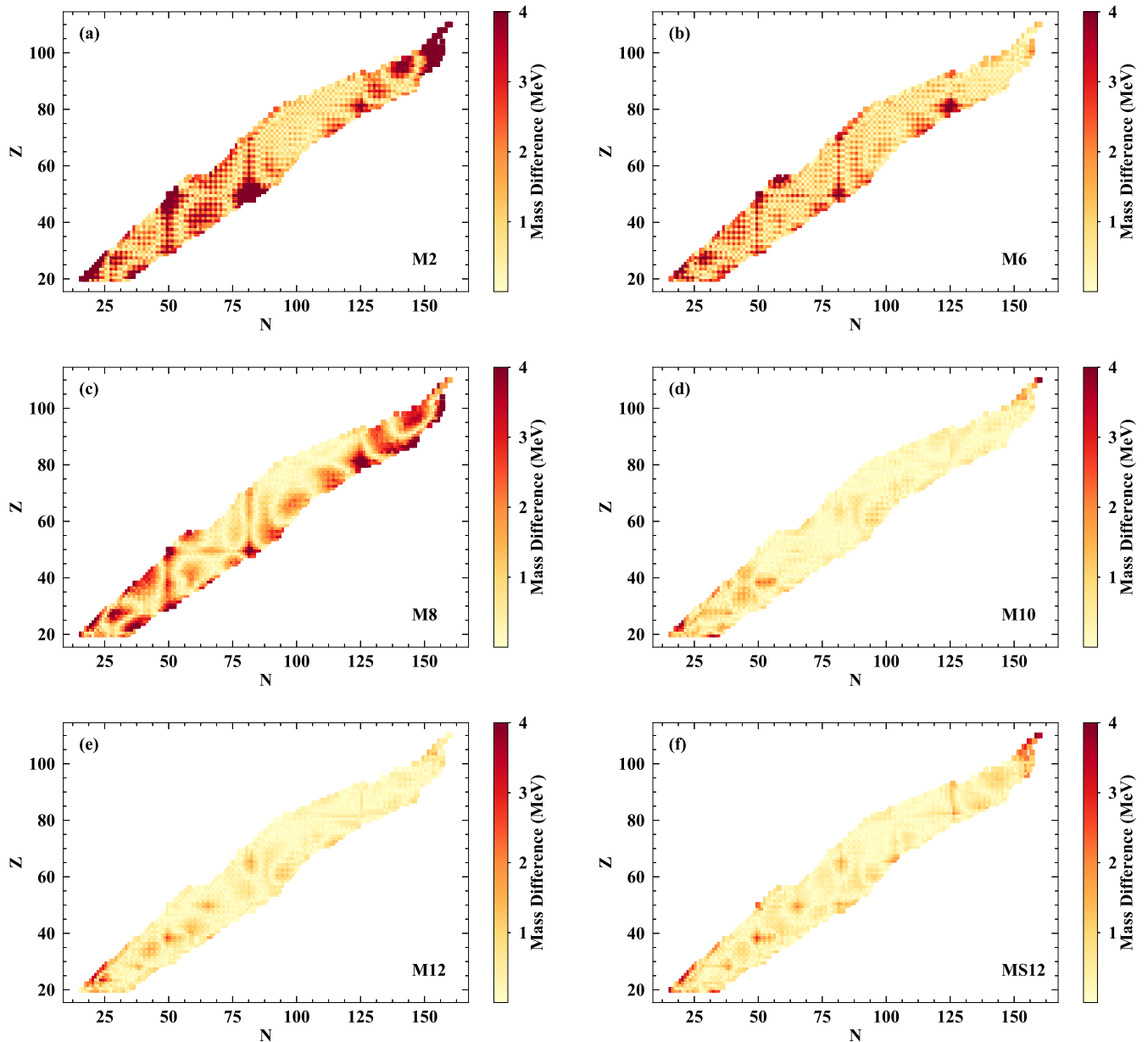


FIG. 1. Mass difference between the MDN predictions and the AME2016 (in MeV) across the nuclear chart for the six models studied in this work: (a) M2, (b) MS2, (c) MS6, (d) MS8, (e) MS10, and (f) MS12. Darker colors indicate regions of larger differences between the AME and the MDN predictions.

initialized randomly. The MDN was found to be converged after training for 100 000 epochs, which is used throughout this work.

III. RESULTS AND DISCUSSION

A. Feature space engineering

Figure 1 shows an overview of the absolute value of the mass difference between the AME2016 masses and the output of the MDN for most of the models in Table I (note that the only MS model that we show is MS12, as the residuals of each of the MN and MSN show similar features). Beginning with Fig. 1(a) for model M2, which only includes the number of neutrons and protons in the input feature space and trains

on just the mass excess, we immediately see that such a small network cannot adequately reproduce the training set, particularly not to the level of accuracy that most mass models reach. This is quantified in the first row of Table II which lists the σ_{RMS} of the MDN predictions with respect to the AME2016. For model M2, $\sigma_{\text{RMS}} = 3.90$ MeV whereas most mass models with phenomenological corrections are on the order of 0.5 MeV, e.g., the Hartree-Fock-Bogoliubov approach [16].

The clearest trend that is seen in the M2 model is that the difference in the masses increase as the nuclei become heavier—most noticeably in the highest- Z region of the nuclear chart. This trend leads us to model M6, which introduces several factors from the semiempirical mass model into the feature space. Although we could improve the training by

TABLE II. σ_{RMS} for the mass excess (column 2) and one-neutron separation energy (column 3) for each of the models described in this work (column 1).

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

tuning the hyperparameters of the neural network (including more layers and nodes), this only serves to lower σ_{RMS} and does not improve the extrapolation beyond the training set or the consistency of the extrapolations (discussed later in Secs. III B and III C). Instead, by including these four values into the feature space, σ_{RMS} is reduced to 1.57 MeV, a decrease of a factor of almost 3. The residuals between the AME2016 masses and the masses predicted by M6 are shown in Fig. 1(b). Now, as the residuals have drastically decreased in the high- Z region of the nuclear chart, it is much easier to see patterns in the residuals. The residuals are noticeably larger at the magic numbers, a prominent feature seen in the mass models studied in Ref. [44] for the two-neutron separation energies. In addition, there is clear odd-even staggering seen in the residuals, meaning that the MDN predictions do not reproduce the staggering that is observed in the data.

We first take into account the even-odd staggering. For each nucleus, we include in the feature space N_{EO} and Z_{EO} , as described in Sec. II B. The mass residuals when this feature space is trained on, M8, are shown in Fig. 1(c). Although there is not an overall decrease in σ_{RMS} —it is now 1.66 MeV, which is actually a slight increase from M6—the mass residuals no longer show the odd-even staggering that is seen in Fig. 1(b). We then can take into account the shell closures, by including in the feature space the distance to the shell closure, as in model M10. As can be seen in Fig. 1(d), the large differences between the AME2016 masses and the MDN predictions at the magic numbers are completely removed. In addition, σ_{RMS} again decreases by almost a factor of 3, to 0.58 MeV, which is on par with theoretical nuclear mass models.

The last two models shown, M12 and MS12, are then aimed at improving the extrapolation of the masses beyond the training set, discussed in detail in Sec. III B. To improve the extrapolation, we first include the shell model orbitals that each N and Z falls in, as described in Sec. II B. The difference between the AME2016 masses and the MDN predicted masses are shown in Fig. 1(e). Within the training space, there is not much difference between models M10 and M12, and indeed, σ_{RMS} only decreases by 0.02 MeV when the shells are included.

In the last model shown, MS12, we include in the prediction both the mass excess, δM , and the one-neutron separation energy, S_n . Because S_n is defined by the differences in masses, as in Eq. (3), including S_n in the loss function should provide an additional constraint on the slope of the mass surface, in addition to the values of the masses. When including this additional constraint, the σ_{RMS} on δM increase by ≈ 0.1 MeV to 0.64 MeV, but the trends across the measured region of the nuclear chart remain similar to those for M12, as shown in Fig. 1(f). In addition, σ_{RMS} for the one-neutron separation energies of the MDN predictions compared to the AME2016 is 0.47 MeV. Even though we have not shown the difference between models MN and MSN for $N = 2, 6, 8, 10$, the results shown for the difference between M12 and MS12—where the extrapolation of the mass difference is improved with the addition of including the neutron separation energy in the output vector—are illustrative of the differences between each of these training pairs. The extrapolation is discussed further in Sec. III B.

Additionally, the addition of features to the input space reduces the mass uncertainty that is predicted by the MDN. Particularly for model M2, outside of the training set, the predicted uncertainties on the masses quickly become so large as to be unusable. Naturally, the MDN predicts increased uncertainties as the nuclei become more exotic (either neutron rich or proton rich). An increase in uncertainties away from the training set is exactly what we would expect from any robust prediction of uncertainties. In addition, as more features are included in the training set, not only are the predictions within the range of the training set more accurate, but the uncertainties within the training set and extrapolated region become smaller.

B. Extrapolation beyond the training set

Along with reducing the σ_{RMS} between the MDN predictions and the AME2016, the second reason for including more features in the input space is to improve the predictions in the extrapolated region where experimental measurements have not been made or are recent enough to have not been included in the AME2016. To look at the MDN predictions, we show the mass differences, δM , along a few select mass chains in Figs. 2(a)–2(f) (Fe, Mo, Sn, W, Pb, and Sg, respectively). We note that these mass chains span nearly the entire range of elements included in the training set, from $Z = 26$ to $Z = 106$, and are representative of the nuclear chart for $A \geq 20$. For each mass chain, we plot four of the MDN calculations, M6, M8, M12, and MS12, along with the AME2016 to compare to the data, and FRDM2012 to show reasonable values for the extrapolated mass differences. Here, we can see in more detail the benefits of adding more information to the feature space of the training set. Going from M6 to M8—which is particularly clear in Fig. 2(f) due to the poor predictive power of the MDN for these two training sets at higher Z values—we see that there is no odd-even staggering along the mass chain for M6, but that this feature appears in M8 when a feature was included for whether the given nucleus has an odd or even number of neutrons/protons, as is seen in the reproduction of the training set as well. In addition, we see that the

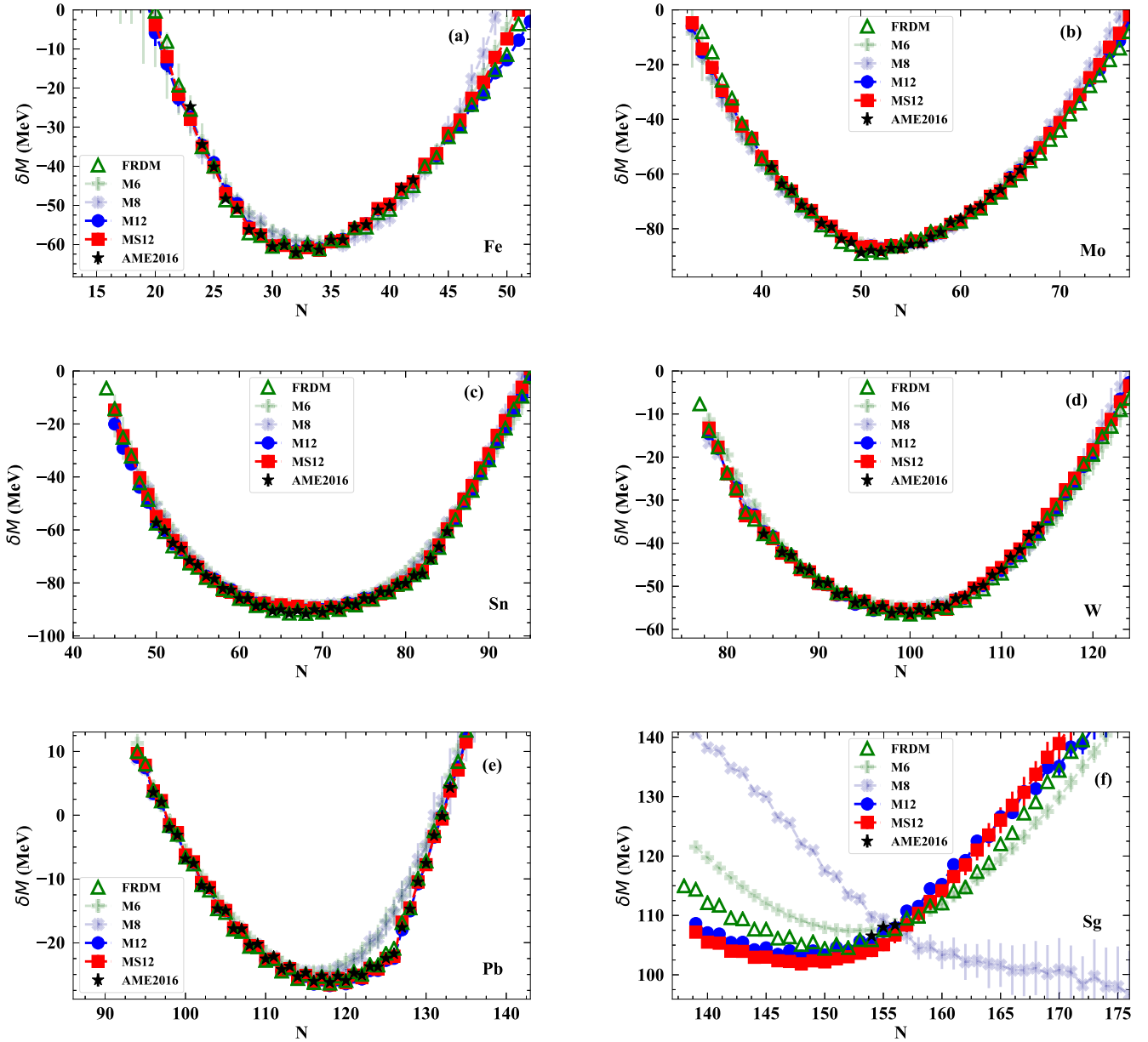


FIG. 2. Comparison of several MDN calculations—M6 (green pluses), M8 (dark blue exes), M12 (blue circles), MS12 (red squares)—to the AME2016 (black stars) and FRDM2012 (green open triangles) for selected mass chains: (a) Fe, (b) Mo, (c) Sn, (d) W, (e) Pb, and (f) Sg.

odd-even staggering is present even for the extrapolated mass predictions, showing that the MDN algorithm has learned that the odd-even staggering is an important feature for predicting the masses. Incorporating these additional features into the input space, even when the overall trend in the predictions is incorrect, is crucial for reproducing the physics trends in the predicted MDN values.

In the difference between M8 and M12/MS12, we also see the importance of including information about the magic numbers and shell closures. It is particularly clear in Fig. 2(e) (but also noticeable in the remaining panels) that the masses around the shell closures are better reproduced when information about the magic numbers is included in the feature space. Instead of smoothly following a parabolic shape (with or without corrections for the even-odd staggering), the larger gaps

between δM values at the shell closures are seen in the mass chains. In addition, one of the greatest benefits of including more features in the training set is to improve and stabilize the extrapolations, particularly noticeable in Fig. 2(f), where only three masses from the chain are available to be included in the training set. For the M8 training set, the extrapolated values of δM do not even follow physical trends, simply decreasing as Z increases, instead of following a parabolic shape. Interestingly, the M6 feature set provides a better prediction for this mass chain than the M8. However, this discrepancy is due more to the instability of the MDN predictions when the feature space is small (MDN calculations with similar values for σ_{RMS} compared to the AME2016 can lead to widely discrepant mass predictions in the extrapolated region) rather than the predictions truly becoming worse as

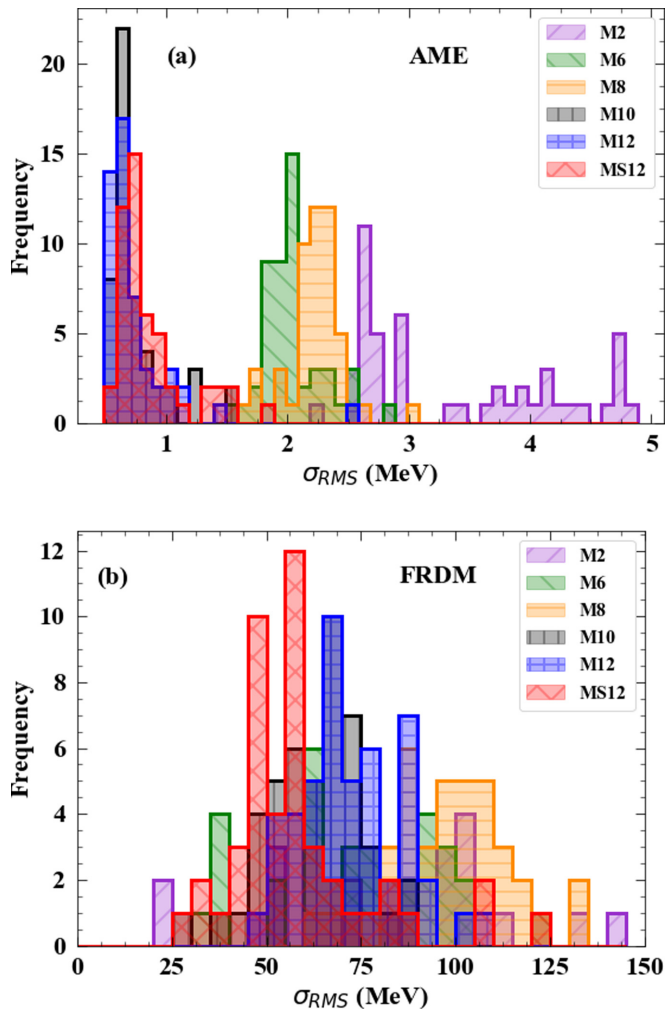


FIG. 3. Distribution of σ_{RMS} values for each of the six MDN model predictions, in (a) the training region, compared to the AME, and (b) in the extrapolated region, compared to the FRDM.

the feature space is increased, as will be discussed further in Sec. III C.

C. Reproducibility of the MDN predictions

We run the MDN training and prediction for each model 50 times to understand the changes in the reproducibility of the MDN algorithm as the size of the feature space is increased. In Fig. 3, we show the distribution of the σ_{RMS} values for δM (in MeV) for each of the models, compared to the AME in Fig. 3(a) (includes only the masses in the training set) and the FRDM in Fig. 3(b) (includes the masses in the training set and the extrapolated region). In Fig. 3(a), models M10, M12, and MS12 have very similar values for σ_{RMS} , as can be seen in Table II for the single calculation analysis. In addition, the widths of these three distributions are also fairly similar, with the tail of the MS12 distribution extending about 0.5 MeV farther than the distributions of M10 and M12. (However, the number of counts in this extended tail is less than 5 out of 50.) When reproducing the training set, the addition of features after the even-odd distinction—M10 and further feature

additions—does not improve the reproduction of the training set, or the reproducibility across multiple MDN trainings.

It is in the extrapolation region that we see a marked improvement with the addition of the extended feature space, including training jointly on the mass excesses and one-neutron separation energies (again, only MS12 is shown, as the other feature spaces where both mass excess and neutron separation energy are included in the output space display the same trends between the MN and MSN feature spaces). In Fig. 3(b), we show the distributions of the σ_{RMS} values for six of the MDN models with respect to the FRDM, to show the overall reproducibility when the extrapolated region is taken into account. On the scale of Fig. 3(b), each of the MDN models has a similar mean σ_{RMS} compared to the FRDM. However, the MDN models where more input features are included have a lower average σ_{RMS} across the 50 iterations of the training along with a narrower distribution. [The narrower distribution of predictions with each increased feature space indicates that the prediction of M8 in Fig. 2(f) is an outlier, but the training sets with a larger feature space lead to more reproducible predictions.]

Here, we see also the benefit of including both the mass excess and one-neutron separation energy in the output of the train, instead of just the mass excess. In particular, the distribution of results from the MS12 model has a lower average σ_{RMS} than the distribution from the M12 model. This result indicates, both, that having a more complete feature space for the network training improves the extrapolation of the observable of interest and that simultaneously predicting two related observables better constrains the extrapolation. Even though the σ_{RMS} does not decrease from M12 to MS12, the extrapolated mass values are more consistent with one another.

IV. CONCLUSION

In conclusion, we have used for the first time the probabilistic machine learning technique, the Mixture Density Network (MDN), to directly learn the nuclear mass excesses using a series of features related to the properties of the nuclear mass models, with well-quantified uncertainties. As the number of features was increased, the σ_{RMS} between the MDN predictions and the 2016 Atomic Mass Evaluation (AME) was greatly improved: from nearly 4 MeV when only neutron and proton number were included in the feature set, down almost 500 keV when the full suite of features were included in the input space. In addition, as more features were added to the input space, the extrapolation of masses beyond the AME was improved, in terms of the shape of each mass chain being more physical. We also saw that simultaneously predicting both the mass excess and the one-neutron separation energy using the MDN further improved the quality of the extrapolated masses and the reproducibility of the MDN prediction. Thus, including more physics information into the input space for the network training improved the performance of the network.

A logical next step for this work would be to test whether this expanded input feature space allows for fewer masses to be included in the training data set while keeping the σ_{RMS} of

the mass excess compared to the AME close to or below 500 keV. In addition, instead of including additional physics information into the input feature space, it is possible to include more physics directly into the neural network itself, either with further constraints built into the loss function or using a specially designed network architecture (e.g., recurrent neural networks are typically used to learn patterns in sequential data, and such a network design could possibly better capture the parabolic-like behavior of each mass chain).

Finally, one of the benefits of using the MDN is to provide realistic uncertainty estimates for each predicted value based on the uncertainty included in the training set. To this end, the MDN could be used in a similar way as Gaussian processes and Bayesian neural networks to perform Bayesian averaging

of multiple mass models to predict the limits of stability (e.g., Refs. [45,47]). It would be interesting to compare the results from the MDN to predictions of other techniques, both in terms of the physics content and the quality of the uncertainty quantification.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract No. 89233218CNA000001. We gratefully acknowledge the support of the U.S. Department of Energy through the LANL/LDRD Program and the Center for Non Linear Studies.

-
- [1] M. R. Mumpower, R. Surman, G. C. McLaughlin, and A. Aprahamian, *Prog. Part. Nucl. Phys.* **86**, 86 (2016).
- [2] C. J. Horowitz, A. Arcones, B. Côté, I. Dillmann, W. Nazarewicz, I. U. Roederer, H. Schatz, A. Aprahamian, D. Atanasov, A. Bauswein *et al.*, *J. Phys. G: Nucl. Part. Phys.* **46**, 083001 (2019).
- [3] N. Vassh, G. C. McLaughlin, M. R. Mumpower, and R. Surman, *Astrophys. J.* **907**, 98 (2021).
- [4] J. J. Cowan, C. Sneden, J. E. Lawler, A. Aprahamian, M. Wiescher, K. Langanke, G. Martínez-Pinedo, and F.-K. Thielemann, *Rev. Mod. Phys.* **93**, 015002 (2021).
- [5] T. Kawano, S. Chiba, and H. Koura, *J. Nucl. Sci. Technol.* **43**, 1 (2006).
- [6] T. Kawano, R. Capote, S. Hilaire, and P. Chau Huu-Tai, *Phys. Rev. C* **94**, 014612 (2016).
- [7] M. B. Chadwick, M. Herman, P. Obložinský, M. E. Dunn, Y. Danon, A. C. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla *et al.*, *Nucl. Data Sheets* **112**, 2887 (2011).
- [8] J. Duflo and A. P. Zuker, *Phys. Rev. C* **52**, R23 (1995).
- [9] M. Liu, N. Wang, Y. Deng, and X. Wu, *Phys. Rev. C* **84**, 014333 (2011).
- [10] P. Möller, A. J. Sierk, T. Ichikawa, and H. Sagawa, *At. Data Nucl. Data Tables* **109-110**, 1 (2016).
- [11] H. Koura, T. Tachibana, M. Uno, and M. Yamada, *Prog. Theor. Phys.* **113**, 305 (2005).
- [12] H. Zhang, J. Dong, N. Ma, G. Royer, J. Li, and H. Zhang, *Nucl. Phys. A* **929**, 38 (2014).
- [13] S. Goriely, N. Chamel, and J. M. Pearson, *Phys. Rev. Lett.* **102**, 152503 (2009).
- [14] N. Schunck and L. M. Robledo, *Rep. Prog. Phys.* **79**, 116301 (2016).
- [15] S. Goriely, F. Tondeur, and J. M. Pearson, *At. Data Nucl. Data Tables* **77**, 311 (2001).
- [16] S. Goriely, N. Chamel, and J. M. Pearson, *Phys. Rev. C* **88**, 024308 (2013).
- [17] S. Goriely and R. Capote, *Phys. Rev. C* **89**, 054318 (2014).
- [18] D. R. Phillips, R. J. Furnstahl, U. Heinz, T. Maiti, W. Nazarewicz, F. M. Nunes, M. Plumlee, M. T. Pratola, S. Pratt, F. G. Viens *et al.*, *J. Phys. G: Nucl. Part. Phys.* **48**, 072001 (2021).
- [19] S. R. Stroberg, J. D. Holt, A. Schwenk, and J. Simonis, *Phys. Rev. Lett.* **126**, 022501 (2021).
- [20] S. König, A. Ekström, K. Hebeler, D. Lee, and A. Schwenk, *Phys. Lett. B* **810**, 135814 (2020).
- [21] A. E. Lovell and F. M. Nunes, *J. Phys. G: Nucl. Part. Phys.* **42**, 034014 (2015).
- [22] A. E. Lovell, F. M. Nunes, J. Sarich, and S. M. Wild, *Phys. Rev. C* **95**, 024611 (2017).
- [23] G. B. King, A. E. Lovell, and F. M. Nunes, *Phys. Rev. C* **98**, 044623 (2018).
- [24] A. E. Lovell and F. M. Nunes, *Phys. Rev. C* **97**, 064612 (2018).
- [25] G. B. King, A. E. Lovell, L. Neufcourt, and F. M. Nunes, *Phys. Rev. Lett.* **122**, 232502 (2019).
- [26] M. Schindler and D. Phillips, *Ann. Phys.* **324**, 682 (2009).
- [27] E. A. Coello Pérez and T. Papenbrock, *Phys. Rev. C* **92**, 064309 (2015).
- [28] R. J. Furnstahl, N. Klco, D. R. Phillips, and S. Wesolowski, *Phys. Rev. C* **92**, 024005 (2015).
- [29] S. Wesolowski, N. Klco, R. J. Furnstahl, D. R. Phillips, and A. Thapaliya, *J. Phys. G: Nucl. Part. Phys.* **43**, 074001 (2016).
- [30] J. A. Melendez, S. Wesolowski, and R. J. Furnstahl, *Phys. Rev. C* **96**, 024003 (2017).
- [31] N. Schunck, J. D. McDonnell, D. Higdon, J. Sarich, and S. M. Wild, *Eur. Phys. J. A* **51**, 169 (2015).
- [32] J. D. McDonnell, N. Schunck, D. Higdon, J. Sarich, S. M. Wild, and W. Nazarewicz, *Phys. Rev. Lett.* **114**, 122501 (2015).
- [33] J. Novak, K. Novak, S. Pratt, J. Vredevogd, C. E. Coleman-Smith, and R. L. Wolpert, *Phys. Rev. C* **89**, 034917 (2014).
- [34] E. Sangaline and S. Pratt, *Phys. Rev. C* **93**, 024908 (2016).
- [35] R. Utama, W.-C. Chen, and J. Piekarewicz, *J. Phys. G: Nucl. Part. Phys.* **43**, 114002 (2016).
- [36] R. Utama, J. Piekarewicz, and H. B. Prosper, *Phys. Rev. C* **93**, 014311 (2016).
- [37] Z.-A. Wang, J. Pei, Y. Liu, and Y. Qiang, *Phys. Rev. Lett.* **123**, 122501 (2019).
- [38] R.-D. Lasserri, D. Regnier, J.-P. Ebran, and A. Penon, *Phys. Rev. Lett.* **124**, 162502 (2020).
- [39] D. Frame, R. He, I. Ipsen, D. Lee, D. Lee, and E. Rrapaj, *Phys. Rev. Lett.* **121**, 032501 (2018).
- [40] A. Sarkar and D. Lee, *Phys. Rev. Lett.* **126**, 032501 (2021).
- [41] A. Ekström and G. Hagen, *Phys. Rev. Lett.* **123**, 252501 (2019).
- [42] S. Wesolowski, I. Svensson, A. Ekström, C. Forssén, R. J. Furnstahl, J. A. Melendez, and D. R. Phillips, *Phys. Rev. C* **104**, 064001 (2021).

- [43] R. Furnstahl, A. Garcia, P. Millican, and X. Zhang, *Phys. Lett. B* **809**, 135719 (2020).
- [44] L. Neufcourt, Y. Cao, W. Nazarewicz, and F. Viens, *Phys. Rev. C* **98**, 034318 (2018).
- [45] L. Neufcourt, Y. Cao, W. Nazarewicz, E. Olsen, and F. Viens, *Phys. Rev. Lett.* **122**, 062502 (2019).
- [46] L. Neufcourt, Y. Cao, S. Giuliani, W. Nazarewicz, E. Olsen, and O. B. Tarasov, *Phys. Rev. C* **101**, 014319 (2020).
- [47] L. Neufcourt, Y. Cao, S. A. Giuliani, W. Nazarewicz, E. Olsen, and O. B. Tarasov, *Phys. Rev. C* **101**, 044307 (2020).
- [48] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, [arXiv:1912.02292](https://arxiv.org/abs/1912.02292).
- [49] C. M. Bishop, Technical Report, Aston University, Department of Computer Science and Applied Mathematics, 1994 (unpublished).
- [50] A. E. Lovell, A. T. Mohan, and P. Talou, *J. Phys. G: Nucl. Part. Phys.* **47**, 114001 (2020).
- [51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, in *Advances in Neural Information Processing Systems* 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Vancouver, Canada, 2019), p. 8024, <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [52] C. F. V. Weizsäcker, *Z. Phys.* **96**, 431 (1935).
- [53] I. J. Thompson and F. M. Nunes, *Nuclear Reactions for Astrophysics: Principles, Calculation and Applications of Low-Energy Reactions* (Cambridge University Press, Cambridge, UK, 2009).
- [54] Z.-A. Wang and J. Pei, *Phys. Rev. C* **104**, 064608 (2021).