

# Product units in gated recurrent units improve nuclear-mass prediction

Ziyuan Li<sup>1,2</sup>, Paulo S.A. Freitas<sup>3</sup>, John W. Clark<sup>3,4</sup>, and Babette Dellen<sup>1</sup>

<sup>1</sup> Department of Mathematics, Informatics and Technology, University of Applied Sciences Koblenz, Joseph-Rovan-Allee 2, 53424 Remagen, Germany

dellen@hs-koblenz.de

<sup>2</sup> Technical University of Munich, Munich, Germany

<sup>3</sup> Department of Mathematics, University of Madeira, Campus Universitário da Pentecada, Funchal, 9020-105, Portugal

<sup>4</sup> Department of Physics, Washington University in St. Louis, 1 Brookings Drive, St. Louis, 63130, MO, USA

**Abstract.** The prediction of masses of atomic nuclei using machine learning can complement theoretical models and advance the exploration of poorly known domains of the nuclear chart. We propose a machine learning technique based on gated recurrent units (GRU), which have demonstrated competitive performance in nuclear-mass prediction by exploiting long-term dependencies. By integrating multiplicative interactions and product-unit transformations within recurrent units, we report significant improvements in nuclear-mass prediction. Computations are performed in the complex domain to jointly capture amplitude and phase dynamics. For interpolation and temporal-extrapolation tasks based on the atomic mass evaluation (AME2016 and AME2020), the complex additive-multiplicative product-unit gated recurrent unit (AM-PU-GRU) model consistently achieves the lowest prediction errors, with an interpolation RMSE of  $0.227 \pm 0.004$  MeV and an extrapolation RMSE of  $0.179 \pm 0.015$  MeV. These results surpass other state-of-the-art machine learning models and also outperform the real-valued GRU baseline and product-unit ablation variants, while remaining robust to different theoretical priors, including WS4 and SEMF. Our findings establish complex-valued product-unit recurrent networks as a new benchmark for sequence-based nuclear-mass prediction.

**Keywords:** Nuclear mass prediction · Gated recurrent units · Complex-valued product-unit neural networks.

## 1 Introduction

Accurate prediction of nuclear masses is a fundamental problem in nuclear physics, with important implications for nuclear-structure theory, nucleosynthesis pathways, and applications in nuclear energy and astrophysics. Experimental evaluations such as AME2016 [9, 10] and AME2020 [6, 21] provide precise measurements for many nuclides, yet large regions of the nuclear chart remain

inaccessible. This motivates theoretical and data-driven approaches capable of both interpolation in known regions and extrapolation to unknown nuclides.

Traditional mass models, such as the Weizsäcker–Skyrme Model (Version 4) (WS4) [22], achieve high accuracy through physics-motivated parametrizations, but their predictive power in unexplored regions is limited [8]. In parallel, machine learning methods have emerged to complement nuclear-mass prediction, notably recurrent neural network (RNN) [8], gated recurrent unit (GRU) [1, 8], and mixture density network (MDN) [18, 17, 12] have achieved strong interpolation accuracy within experimentally known regions, sometimes comparable to physics-based models. These approaches benefit from their ability to learn complex nonlinear relationships directly from data and have been shown to reduce prediction errors when sufficient experimental measurements are available. However, many of these architectures remain limited in extrapolation capability, as they primarily rely on additive representations and struggle to capture the higher-order nonlinear dependencies inherent in nuclear-mass systematics.

The product-unit (PU) [4, 11, 3, 2, 13, 15, 14] approach to machine learning was introduced as an alternative to summation-based formulations, enabling multiplicative interactions that provide compact representations of polynomial and power-law relations. More recently, complex-valued PU extensions [2, 13] have been explored, enabling joint modeling of amplitude and phase dynamics.

Building on these advances, we propose complex-valued GRU extensions for nuclear mass prediction. By integrating multiplicative interactions and product-unit transformations into recurrent frameworks, we obtain two novel architectures, the multiplicative-interaction product-unit GRU (MI-PU-GRU) and the additive-multiplicative product-unit GRU (AM-PU-GRU).

We next evaluate these architectures on interpolation and temporal extrapolation based on AME2016 and AME2020, comparing against real-valued baselines, ablations, and prior-informed setups. The complex AM-PU-GRU model achieves the lowest error on both tasks, setting a new benchmark for sequence-based nuclear-mass prediction.

## 2 Related Work

Product units (PUs) were introduced as an alternative to summation-based neurons in neural networks. Instead of computing a weighted sum followed by a nonlinear activation, a PU models multiplicative interactions by raising each input to a learnable exponent and taking their product according to

$$y = \prod_{i=1}^n x_i^{w_i} = \exp\left(\sum_{i=1}^n w_i \log |x_i|\right). \quad (1)$$

This formulation substantially enhances the expressive capacity of neural networks, enabling compact representation of polynomial, power-law, and rational relationships. As a result, PU-based architectures have been shown to improve extrapolation in scientific prediction tasks such as image classification and function approximation [3, 2].

More recently, complex-valued PU networks have been explored for signals with amplitude and phase. In Eq. (1), the complex extension combines  $\log|x_i|$  with  $\arg(x_i)$  in the exponent, introducing both amplitude scaling and phase rotation via the exponential mapping. This richer inductive bias has improved robustness in MRI reconstruction and nuclear-mass prediction [2, 13, 14].

### 3 Methodology

#### 3.1 Baseline: Gated Recurrent Unit

The GRU is a recurrent neural network variant designed to capture long-term dependencies by mitigating the vanishing gradient problem. Two gating mechanisms are introduced: an update gate and a reset gate, which regulate the information flow and memory update within each recurrent unit.

Given an input vector  $x_t \in \mathbb{R}^d$  and the previous hidden state  $h_{t-1} \in \mathbb{R}^H$  at time step  $t$ , a GRU computes the update gate  $z_t \in \mathbb{R}^H$ , reset gate  $r_t \in \mathbb{R}^H$ , candidate hidden state  $\tilde{h}_t \in \mathbb{R}^H$ , and the new hidden state  $h_t \in \mathbb{R}^H$  as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad \text{update gate} \quad (2)$$

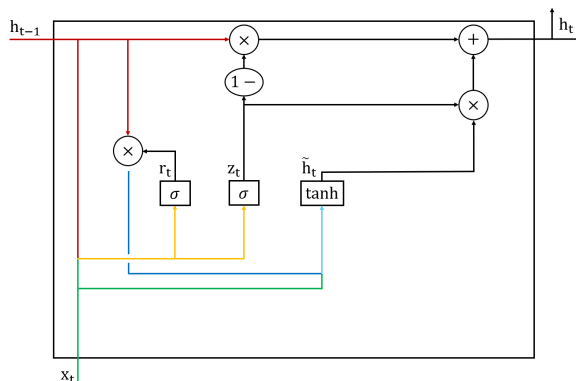
$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad \text{reset gate} \quad (3)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \quad \text{candidate hidden state} \quad (4)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad \text{new hidden state.} \quad (5)$$

Here,  $\sigma(\cdot)$  denotes the sigmoid activation function,  $\tanh(\cdot)$  is the hyperbolic tangent, and  $\odot$  represents element-wise multiplication. The learnable parameters satisfy  $W_{\{z,r,h\}} \in \mathbb{R}^{H \times d}$  and  $U_{\{z,r,h\}} \in \mathbb{R}^{H \times H}$ . Bias terms are used in our implementation but omitted for notational simplicity.

Figure 1 illustrates the internal structure of a standard GRU cell. The diagram is consistent with the above equations: The input  $x_t$  and previous hidden state  $h_{t-1}$  are jointly used to compute the reset gate  $r_t$  and update gate  $z_t$ . The reset gate  $r_t$  controls how much of  $h_{t-1}$  contributes to the candidate activation



**Fig. 1.** Computational graph of a standard GRU cell.

$\tilde{h}_t$ . The final hidden state  $h_t$  is computed as a convex combination of  $h_{t-1}$  and  $\tilde{h}_t$ , governed by the update gate  $z_t$ . The parameters of the model are defined through the weight matrices associated with the update gate, reset gate, and candidate hidden state.

### 3.2 Multiplicative-Interaction Product-Unit GRU (MI-PU-GRU)

To enhance the nonlinear modeling capacity of GRUs, we introduce the MI-PU-GRU cell, which incorporates two major innovations: a multiplicative interaction (MI) branch and a product-unit (PU) transformation in the candidate-state computation. Figure 2 shows the MI-PU-GRU computational graph.

The MI-PU-GRU case maintains the standard GRU gating structure but modifies the candidate hidden state  $\tilde{h}_t$  as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad \text{update gate} \quad (6)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad \text{reset gate} \quad (7)$$

$$\text{MI}_t = \text{MI}(x_t, r_t \odot h_{t-1}) \quad \text{multiplicative interaction} \quad (8)$$

$$\tilde{h}_t = \text{PU}([x_t, r_t \odot h_{t-1}, \text{MI}_t]) \quad \text{PU-transformed candidate} \quad (9)$$

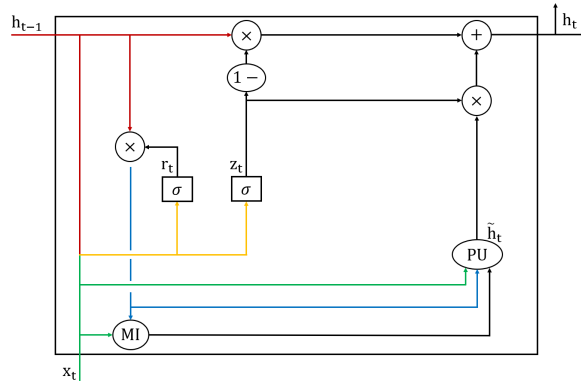
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad \text{new hidden state.} \quad (10)$$

The MI term is computed using an element-wise exponential interaction between the input and hidden representations:

$$\text{MI}(x, h) = \exp((W_{\text{mi}}x) \odot (U_{\text{mi}}h) + b_{\text{mi}}), \quad (11)$$

where  $W_{\text{mi}}$  and  $U_{\text{mi}}$  are learnable projection matrices and  $b_{\text{mi}}$  is a bias term. The MI branch computes a feature-wise exponential interaction between the input  $x_t$  and the reset-gated previous state  $r_t \odot h_{t-1}$ , producing a multiplicative representation  $\text{MI}_t$ . The exponential function ensures that the output is strictly positive, which is necessary for the subsequent PU transformation. This representation, along with  $x_t$  and  $r_t \odot h_{t-1}$ , is fed into a Product-Unit (PU) transformation:

$$\text{PU}(x) = \exp(W_{\text{pu}} \cdot \log(\max(x, \text{softplus}(\theta) + 10^{-7})) + b_{\text{pu}}), \quad (12)$$



**Fig. 2.** Computational graph of the MI-PU-GRU cell.



### 3.4 Complex-Valued MI-PU-GRU

To extend MI-PU-GRU to the complex domain, we adopt a fully complex-valued formulation. All learnable weights, hidden states, and intermediate activations are complex-valued. The real-valued input  $x_t \in \mathbb{R}^d$  is embedded into the complex vector space via a zero-imaginary extension  $\tilde{x}_t = x_t + i0 \in \mathbb{C}^d$ , and the hidden state satisfies  $h_{t-1} \in \mathbb{C}^H$ .

The update and reset gates are kept real-valued to ensure stable and interpretable gating; specifically, we apply the sigmoid to the real part of complex pre-activations, yielding  $z_t, r_t \in \mathbb{R}^H$ :

$$z_t = \sigma(\Re(W_z \tilde{x}_t + U_z h_{t-1})), \quad (21)$$

$$r_t = \sigma(\Re(W_r \tilde{x}_t + U_r h_{t-1})), \quad (22)$$

$$\mathbf{MI}_t = \mathbf{MI}_{\mathbb{C}}(\tilde{x}_t, r_t \odot h_{t-1}), \quad (23)$$

$$\tilde{h}_t = \mathbf{PU}_{\mathbb{C}}([\tilde{x}_t, r_t \odot h_{t-1}, \mathbf{MI}_t]), \quad (24)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (25)$$

Using real-valued gates (bounded in  $[0, 1]$ ) provides a numerically stable and interpretable mixing of complex hidden states without introducing additional phase rotations.

The MI branch performs feature-wise exponential interactions in the complex domain:

$$\mathbf{MI}_{\mathbb{C}}(x, h) = \exp((W_{\text{mi}}x) \odot (U_{\text{mi}}h) + b_{\text{mi}}), \quad (26)$$

where all parameters are complex-valued. Since  $\exp(a + ib) = \exp(a)(\cos b + i \sin b)$ , the interaction introduces both amplitude scaling and phase rotation, enhancing expressiveness.

The PU transformation in the complex domain is defined as

$$\mathbf{PU}_{\mathbb{C}}(x) = \exp(W_{\text{pu}}(\log r(x) + i\phi(x)) + b_{\text{pu}}), \quad (27)$$

where  $r(x) = \max(|x|, \text{softplus}(\theta) + \epsilon)$ ,  $\phi(x) = \arg(x)$ , and  $\epsilon = 10^{-7}$ . Here, the logarithm is applied to the stabilized magnitude  $r(x)$  and the phase term  $\phi(x)$  is handled explicitly.

Since our downstream task requires real-valued predictions, the final output of the complex-valued recurrent network is mapped back to the real domain by extracting the real part of the hidden state. This mapping ensures compatibility with standard loss functions defined over  $\mathbb{R}$ .

### 3.5 Complex-Valued AM-PU-GRU

As with standard AM-PU-GRU, complex AM-PU-GRU builds upon complex MI-PU-GRU by integrating an additive path and a fusion gate to balance linear

and nonlinear dynamics in  $\mathbb{C}$ . The update procedure is given by:

$$z_t = \sigma(\Re(W_z \tilde{x}_t + U_z h_{t-1})), \quad (28)$$

$$r_t = \sigma(\Re(W_r \tilde{x}_t + U_r h_{t-1})), \quad (29)$$

$$g_t = \sigma(\Re(W_g \tilde{x}_t + U_g h_{t-1})), \quad (30)$$

$$h_{\text{add}} = \tanh(\Re(W_{\text{add}} \tilde{x}_t + U_{\text{add}}(r_t \odot h_{t-1}))), \quad (31)$$

$$\text{MI}_t = \text{MI}_{\mathbb{C}}(\tilde{x}_t, r_t \odot h_{t-1}), \quad (32)$$

$$h_{\text{mul}} = \text{PU}_{\mathbb{C}}([\tilde{x}_t, r_t \odot h_{t-1}, \text{MI}_t]), \quad (33)$$

$$\tilde{h}_t^{\text{add}} = h_{\text{add}} + i0, \quad (34)$$

$$\tilde{h}_t = g_t \odot h_{\text{mul}} + (1 - g_t) \odot \tilde{h}_t^{\text{add}}, \quad (35)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (36)$$

Here,  $h_{\text{add}} \in \mathbb{R}^H$  is recast to  $\mathbb{C}^H$  before fusion. The use of real-valued gates ensures numerically stable and interpretable mixing while allowing the hidden state to evolve in the full complex plane.

### 3.6 Task Formulation

Our goal is to predict the mass excess of an atomic nucleus from sequential nuclear structure data, which we formulate as a sequence-to-one regression problem. For each target nucleus with proton number  $Z^*$  and neutron number  $N^*$ , we construct a short ordered sequence of  $T = 5$  neighboring nuclides based on proximity in the  $(Z, N)$  chart.

For a target nucleus  $(Z^*, N^*)$  with mass number  $A^* = Z^* + N^*$ , the first  $T - 1 = 4$  elements are selected from the corresponding reference set of experimentally measured nuclei. Specifically, excluding the target nucleus itself, we consider nuclei satisfying  $A \leq A^*$  and select the four nearest ones in the  $(Z, N)$  plane as context nuclei; their measured mass excess values are used as input features. The final element  $x_T$  corresponds to the target nucleus; its mass excess value in the input is initialized by the WS4 mass model (i.e.,  $\text{ME}_T = \text{ME}^{\text{WS4}}(Z^*, N^*)$ ) to provide a physically informed prior.

Given  $X$ , the model learns a nonlinear mapping  $f : \mathbb{R}^{T \times 3} \rightarrow \mathbb{R}$  to predict the ground-truth mass excess  $y = \text{ME}^{\text{AME}}(Z^*, N^*)$  of the target nucleus. This design allows the model to exploit local continuity over neighboring nuclides while combining empirical measurements and theoretical priors.

To capture the complex nonlinear dependencies inherent in nuclear-mass systematics, we propose complex-valued extensions of the MI-PU-GRU and AM-PU-GRU architectures, which integrate multiplicative interactions and product-unit transformations into the recurrent framework.

In addition to these primary models, we also implement their real-valued counterparts (MI-PU-GRU and AM-PU-GRU) to provide a controlled comparison between real and complex formulations. Furthermore, we design ablation variants in the complex domain, including PU-GRU, MI-GRU, AM-GRU, and the baseline GRU, by selectively disabling specific architectural components.

These ablations allow us to isolate and quantify the contribution of multiplicative interactions, product-unit transformations, and additive–multiplicative fusion. Finally, to evaluate robustness with respect to theoretical priors, we perform prior stability experiments where the target nucleus is initialized using the semi-empirical mass formula (SEMF), which incorporates volume, surface, Coulomb, asymmetry, and pairing terms. The SEMF-based mass excess estimate is computed using a simplified version of the Weizsäcker formula [23]:

$$ME = (ZM_p + NM_n + ZM_e - B - A) \cdot 931.5 \times 10^3 \text{ keV}, \quad (37)$$

where  $B$  is the total binding energy, which is calculated as

$$B = a_v A - a_s A^{2/3} - a_c \frac{Z(Z-1)}{A^{1/3}} - a_a \frac{(A-2Z)^2}{A} + \delta, \quad (38)$$

with  $A = Z + N$  and the pairing term  $\delta$  defined as  $\delta = +a_p/\sqrt{A}$  for even–even nuclei,  $\delta = -a_p/\sqrt{A}$  for odd–odd nuclei, and  $\delta = 0$  otherwise.

## 4 Experiments and Results

### 4.1 Experimental Setup

*Experiment I: Interpolation.* This experiment assesses the model’s interpolation capability within the domain of experimentally known nuclides. We randomly split the AME2020 dataset into training, validation, and test subsets using a 70%-15%-15% ratio. During sequence construction for the validation and test sets, only the experimentally measured mass-excess values from the training set are used to provide input features for precursor nuclei. That is, all non-target entries in a sequence are filled using training set information, ensuring no data leakage. Model selection is based on validation performance: the version of the model that achieves the lowest validation loss during training is retained and used for final evaluation on the test set.

*Experiment II: Temporal Extrapolation.* To test the extrapolation capability of our models, we simulate a temporal generalization scenario by training on AME2016-era nuclides and testing on nuclides added in the subsequent AME2020 release. This setup allows us to evaluate the model’s ability to generalize to newly measured nuclei that were previously unknown. In addition, we reserve 5% of the AME2016 data as an interpolation test set to provide a comprehensive assessment of the model’s performance on both interpolation and extrapolation tasks within the same experimental framework. As in Experiment I, all sequences for both extrapolation and interpolation test sets are constructed using only nuclei from the training set to provide precursor mass excess values.

*Training Setup.* All models are trained in PyTorch for 3000 epochs using MSE loss and the RAdam optimizer (initial learning rate  $10^{-3}$ ). We use a step learning-rate schedule that halves the learning rate every 500 epochs, and set the batch size to 64.

## 4.2 Main Results

To evaluate the predictive capability of the proposed architectures, we first compare the baseline real-valued GRU with the complex-valued MI-PU-GRU and AM-PU-GRU under the WS4 prior. For a fair comparison, we control the parameter scale by adjusting network depth: the baseline GRU is implemented with 120 stacked recurrent layers, while the complex MI-PU-GRU and AM-PU-GRU are implemented with 60 and 50 layers, respectively. This configuration results in a comparable number of trainable parameters across all models. For complex-valued architectures, each learnable weight is represented as  $a + ib$ , where  $a$  is the real and  $b$  the imaginary part of the complex weight, and thus counts as two trainable parameters when reporting parameter sizes.

Each network is trained independently five times without fixing the random seed, in order to account for stochasticity in initialization and training dynamics. We report the mean and standard deviation of the RMSE over these runs.

The interpolation results of Experiment I are summarized in Table 1. As can be seen, both complex-valued PU-GRU variants consistently outperform the real-valued GRU baseline. In particular, the complex AM-PU-GRU model achieves the lowest interpolation error (0.227 MeV) and the lowest sample standard deviation (0.004 MeV), highlighting the effectiveness of combining additive and multiplicative candidate paths within the complex domain.

**Table 1.** Principal results of **Experiment I** using WS4 estimates. Here, RV denotes the real-valued model, CV the complex-valued model. Results are reported as the mean  $\pm$  sample standard deviation (MeV).  $N_{\text{param}}$  indicates the number of trainable parameters. IntRMSE denotes interpolation RMSE (and ExtRMSE denotes extrapolation RMSE, when applicable). The best result in each column is highlighted in bold.

Model	$N_{\text{param}}$	IntRMSE (MeV)
RV GRU	45121	$0.242 \pm \mathbf{0.004}$
CV MI-PU-GRU	38165	$0.236 \pm 0.005$
CV AM-PU-GRU	37605	$\mathbf{0.227} \pm \mathbf{0.004}$

The results of Experiment II are summarized in Table 2. Compared to the real-valued GRU baseline, both complex PU-GRU variants yield lower extrapolation errors, with the complex AM-PU-GRU again achieving the best overall performance. In particular, AM-PU-GRU attains the lowest interpolation RMSE of  $0.253 \pm 0.003$  MeV and the lowest extrapolation RMSE of  $0.179 \pm 0.015$  MeV. These findings indicate that the proposed complex architectures improve both interpolation and temporal-extrapolation performance under the present experimental setup. In particular, the extrapolation capability of the complex AM-PU-GRU is significantly stronger, indicating its effectiveness in capturing long-range structural dependencies in nuclear mass systematics. Moreover, AM-PU-GRU exhibits the most stable extrapolation performance, as reflected by the lowest standard deviation among all models.

The lower extrapolation RMSE in Table 2 does not mean that extrapolation is generally easier. In Experiment II, the interpolation set is a 5% hold-out from AME2016, whereas the extrapolation set consists of nuclei newly added in

**Table 2.** Main results of **Experiment II** using WS4 estimates.

Model	IntRMSE (MeV)	ExtRMSE (MeV)
RV GRU	0.261 $\pm$ <b>0.003</b>	0.205 $\pm$ 0.018
CV MI-PU-GRU	0.254 $\pm$ <b>0.003</b>	0.186 $\pm$ 0.020
CV AM-PU-GRU	<b>0.253 <math>\pm</math> 0.003</b>	<b>0.179 <math>\pm</math> 0.015</b>

AME2020, so the two test sets differ in composition. Since prediction depends on local neighborhood structure in the  $(Z, N)$  chart and the WS4 prior, some AME2020-added nuclei may be easier to predict than the held-out AME2016 subset.

We further compare our best-performing models with existing approaches from the literature, as summarized in the Discussion section.

### 4.3 Comparative Analysis: Real vs. Complex

To assess the effect of complex-valued modeling, we compare the performance of real-valued PU-GRU variants with their complex-valued counterparts under the WS4 prior. Specifically, we implement a 90-layer real-valued MI-PU-GRU and a 75-layer real-valued AM-PU-GRU, such that their parameter counts are comparable to those of the 60-layer complex MI-PU-GRU and the 50-layer complex AM-PU-GRU, respectively. Each real-valued network is trained independently three times, and the reported performance is the average RMSE across runs.

The results, summarized in Tables 3 and 4, indicate that the complex-valued PU-GRU variants consistently outperform their real-valued counterparts in both interpolation and extrapolation settings. This demonstrates that extending PU-GRU architectures into the complex domain yields richer representational capacity by jointly modeling amplitude and phase dynamics, leading to improved predictive accuracy.

**Table 3.** Results of **Experiment I** for real-valued variants.

Model	$N_{\text{param}}$	IntRMSE (MeV)
RV MI-PU-GRU	42580	0.240
RV AM-PU-GRU	41710	0.233

**Table 4.** Results of **Experiment II** for real-valued variants.

Model	IntRMSE (MeV)	ExtRMSE (MeV)
RV MI-PU-GRU	0.255	0.196
RV AM-PU-GRU	0.258	0.194

### 4.4 Ablation Study

To isolate the contributions of individual architectural components, we have performed an ablation study in the complex domain. We implement four variants: a complex-valued GRU with 80 layers, a 70-layer complex MI-GRU (without

PU and additive path), a 60-layer complex AM-GRU (without PU but with additive multiplicative fusion), and an 80-layer complex PU-GRU (retaining PU but without explicit MI or AM mechanisms). All models are configured to have comparable parameter counts for a fair comparison.

The results are presented in Tables 5 and 6. We observe that neither the complex MI-GRU nor the complex AM-GRU improves upon the baseline complex GRU, and in fact both exhibit degraded performance. By contrast, the complex PU-GRU model achieves a substantial reduction in error, clearly outperforming the baseline GRU. This indicates that the product-unit transformation is the key factor driving improvements, while multiplicative interactions or additive-multiplicative fusion alone are insufficient to enhance predictive accuracy.

**Table 5.** Ablation results of **Experiment I** with complex-valued GRU, MI-GRU, AM-GRU, and PU-GRU implementation.

Model	$N_{\text{param}}$	IntRMSE (MeV)
CV GRU	40482	0.400
CV MI-GRU	41582	0.801
CV AM-GRU	38522	2.051
CV PU-GRU	40482	0.261

**Table 6.** Ablation results of **Experiment II** with complex-valued GRU, MI-GRU, AM-GRU and PU-GRU strategies.

Model	IntRMSE (MeV)	ExtRMSE (MeV)
CV GRU	0.493	0.259
CV MI-GRU	0.843	0.241
CV AM-GRU	1.316	0.282
CV PU-GRU	0.261	0.197

#### 4.5 Prior Stability Analysis

Finally, we evaluate robustness to the choice of the theoretical baseline estimate used to initialize the target nucleus (WS4 vs. SEMF). In this experiment, the network configurations remain the same as in the main results, i.e., a 120-layer real-valued GRU baseline, a 60-layer complex MI-PU-GRU, and a 50-layer complex AM-PU-GRU. The only difference lies in the initialization of the target nucleus, where we replace the WS4 estimate with the SEMF, which incorporates volume, surface, Coulomb, asymmetry, and pairing terms.

The results are summarized in Tables 7 and 8. Compared to the WS4-based experiments, the relative ranking of models is unchanged: both complex PU-GRU variants outperform the real-valued GRU, and the complex AM-PU-GRU version continues to achieve the best overall accuracy. These findings confirm that the proposed architectures are robust to the choice of baseline initialization model (WS4 vs. SEMF) and maintain stable capability under this change.

**Table 7.** Prior stability results of **Experiment I** under SEMF initialization.

Model	IntRMSE (MeV)
RV GRU	0.414
CV MI-PU-GRU	0.402
CV AM-PU-GRU	<b>0.401</b>

**Table 8.** Prior stability results of **Experiment II** under SEMF initialization.

Model	IntRMSE (MeV)	ExtRMSE (MeV)
RV GRU	0.481	0.598
CV MI-PU-GRU	0.460	0.505
CV AM-PU-GRU	0.455	<b>0.477</b>

## 5 Discussion

Beyond comparisons among our proposed variants, it is important to contextualize the performance of the PU-GRU architectures with respect to existing approaches in the literature. Numerous machine-learning methods have been previously applied to nuclear mass prediction, including MDN, categorical gradient boosting trees (CatBoost), and fully connected neural networks (FCNN). We summarize representative results reported in prior studies and compare them against our best-performing model, the complex AM-PU-GRU. Table 9 summarizes the interpolation performance of various machine learning models and our models on the AME2020 dataset (Experiment I). All models are evaluated on the task of predicting nuclear mass excess values using various input feature sets and network architectures.

**Table 9.** Comparison with existing approaches in **Experiment I**. All RMSE values are reported in MeV. The column “Selection criterion” specifies the nuclide selection rule,  $N_{\text{feat}}$  denotes the number of input features per nuclide, and  $E_{\text{measure}}$  refers to the experimental uncertainty of the measured mass-excess values.

Model	Selection criterion	$N_{\text{feat}}$	IntRMSE (MeV)
XGBoost [20]			3.702
MLP Regressor [20]		4	3.128
RFR [20]			3.089
MISR [19]	$12 \leq Z \leq 50$	10	0.99
MDN [17]	$Z, N \geq 5$	9	0.395
MDN [12]	$Z \geq 10$	11	0.246
CatBoost [5]	$8 \leq Z \leq 108$	7	0.189
	$8 \leq N \leq 162$		
PI-FCNN [7]	$Z, N > 20$ $E_{\text{measure}} < 100 \text{ keV}$	13	0.122
CPUN [2]		10	0.438
RNN [8]			0.601
LSTM [8]	$Z, N \geq 8$	11	0.557
GRU [8]			0.459
CV AM-PU-GRU [ours]	$Z, N \geq 8$	3	0.227

As shown in Table 9, Jalili et al. [8] reported results for several recurrent architectures, including GRU, RNN, and LSTM, but their models utilize broader input features, as well as standard recurrent designs without product-unit augmentation. Our complex-valued AM-PU-GRU model achieves superior performance with a more compact and physics-informed input representation, achieving an RMSE of 0.227 MeV for mass-excess prediction.

We also compare with the CPUN proposed by Dellen et al. [2], which yields an RMSE of 0.438 MeV. While their model also incorporates product units and complex-valued representations, it is based on a fully connected feedforward architecture and lacks the temporal modeling capability of our recurrent design. In contrast, our PU-GRU framework integrates multiplicative interactions into a recurrent paradigm and explicitly exploits sequence-level dependencies among neighboring nuclides, leading to improved performance over CPUN [2].

Nevertheless, our model does not outperform the best results reported in the literature, notably CatBoost [5] (0.189 MeV) and Physics-Informed FCNN [7] (0.122 MeV). This gap is partly due to differences in data selection and domain-specific constraints. For example, [7] evaluates only nuclides with  $Z, N > 20$  and measured error  $< 100$  keV, excluding many difficult edge cases, while [5] imposes upper bounds of  $Z \leq 108$  and  $N \leq 162$ , which also simplifies the task. By contrast, our model is evaluated over a broader nuclide range without such filtering, making the problem more challenging. In addition, unlike tree-based methods such as CatBoost [5], which are not end-to-end and often depend on feature engineering, our PU-GRU follows an end-to-end paradigm that maps raw nuclear sequences directly to mass excess values while retaining modular interpretability.

Table 10 summarizes the interpolation performance on AME2016 and temporal extrapolation on AME2020 of various machine-learning models and our models (Experiment II). Across both interpolation and extrapolation tasks, our complex-valued AM-PU-GRU model consistently achieves the lowest prediction error (0.253 MeV and 0.179 MeV) among all compared methods. Furthermore, its performance even surpasses certain non-end-to-end approaches, such as physics-informed FCNN [7] and CNN-WS4 [16], highlighting the effectiveness of learning nuclear mass systematics directly from sequential data.

**Table 10.** Comparison with existing approaches in **Experiment II**. All RMSE values are reported in MeV.

Model	Selection criterion	$N_{\text{feat}}$	IntRMSE	ExtRMSE
GPR [24]	$Z, N \geq 8$	12	0.26	0.67
SVR [24]			0.39	0.74
MDN [18]	$Z \geq 20$	8	0.316	0.336
CNN-WS4 [16]	$Z, N > 8$	9	-	0.211
PI-FCNN [7]	$Z, N > 20$ and $E_{\text{measure}} < 100 \text{ keV}$	13	-	0.191
CV AM-PU-GRU [ours]	$Z, N \geq 8$	3	0.253	0.179

## 6 Conclusion

We have presented complex-valued extensions of the MI-PU-GRU and AM-PU-GRU architectures for nuclear mass prediction. By embedding multiplicative interactions and product-unit transformations into recurrent units and extending them to the complex domain, our models capture nonlinear dependencies and long-range structural correlations that are difficult to represent with standard additive recurrent architectures.

Through systematic evaluation on interpolation and extrapolation tasks using AME2016 and AME2020 datasets, we have shown that the complex AM-PU-GRU model achieves the best overall performance, consistently yielding the lowest RMSE across all scenarios. Comparative analyses further reveal that complex-valued models outperform their real-valued counterparts, while ablation studies identify the product-unit transformation as the critical component driving accuracy improvements. Moreover, prior stability experiments confirm that the proposed architectures maintain consistent predictive capability under both WS4 and SEMF initialization, demonstrating robustness with respect to theoretical priors.

Beyond outperforming existing end-to-end machine learning approaches, our models also surpass certain non-end-to-end correction schemes, underscoring the potential of product-unit recurrent architectures as powerful tools for scientific prediction tasks. Future work will explore extending complex PU-based recurrent networks to other nuclear observables and to broader domains where extrapolation capability is essential.

## References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. Dellen, B., Jaekel, U., Freitas, P.S., Clark, J.W.: Predicting nuclear masses with product-unit networks. *Physics Letters B* **852**, 138608 (2024)
3. Dellen, B., Jaekel, U., Wolnitza, M.: Function and pattern extrapolation with product-unit networks. In: *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part II* 19. pp. 174–188. Springer (2019)
4. Durbin, R., Rumelhart, D.E.: Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation* **1**(1), 133–142 (1989)
5. Guo, J.L., Wang, H.L., Zhang, Z.Z., Liu, M.L.: Probing the refined performance of the categorical-boosting algorithm to the hartree-fock-bogoliubov mass model with several skyrme forces. *Physical Review C* **111**(5), 054322 (2025)
6. Huang, W., Wang, M., Kondev, F.G., Audi, G., Naimi, S.: The ame 2020 atomic mass evaluation (i). evaluation of input data, and adjustment procedures. *Chinese Physics C* **45**(3), 030002 (2021)
7. Huang, Y., Chen, J., Jia, J., Liu, L.M., Ma, Y.G., Zhang, C.: Validation and extrapolation of atomic masses with a physics-informed fully connected neural network. *Physical Review C* **111**(3), 034329 (2025)

8. Jalili, A., Pan, F., Chen, A.X., Draayer, J.P.: Deep learning approaches for nuclear binding energy prediction: a comparative study of rnn, gru and lstm models. arXiv preprint arXiv:2503.19348 (2025)
9. Kondev, F., Naimi, S.: The ame2016 atomic mass evaluation (i). evaluation of input data; and adjustment procedures. Chinese physics C **41**(3), 030002 (2017)
10. Kondev, F., Naimi, S.: The ame2016 atomic mass evaluation (ii). tables, graphs and references. Chinese Physics C **41**(3), 030003 (2017)
11. Leerink, L., Giles, C., Horne, B., Jabri, M.: Learning with product units. Advances in neural information processing systems **7** (1994)
12. Li, M., Sprouse, T.M., Meyer, B.S., Mumpower, M.R.: Atomic masses with machine learning for the astrophysical r process. Physics Letters B **848**, 138385 (2024)
13. Li, Z., Jaekel, U., Dellen, B.: Data-driven 3d shape completion with product units. In: International Conference on Computational Science. pp. 302–315. Springer (2024)
14. Li, Z., Jaekel, U., Dellen, B.: Advancing complex-valued neural networks with product units for mri reconstruction. In: International Conference on Neural Information Processing. pp. 540–554. Springer (2025)
15. Li, Z., Jaekel, U., Dellen, B.: Deep residual learning with product units. arXiv preprint arXiv:2505.04397 (2025)
16. Lu, Y., Shang, T., Du, P., Li, J., Liang, H., Niu, Z.: Nuclear mass predictions based on a convolutional neural network. Physical Review C **111**(1), 014325 (2025)
17. Mumpower, M., Li, M., Sprouse, T.M., Meyer, B.S., Lovell, A.E., Mohan, A.T.: Bayesian averaging for ground state masses of atomic nuclei in a machine learning approach. Frontiers in physics **11**, 1198572 (2023)
18. Mumpower, M.R., Sprouse, T.M., Lovell, A.E., Mohan, A.T.: Physically interpretable machine learning for nuclear masses. Physical Review C **106**(2), L021301 (2022)
19. Munoz, J.M., Udrescu, S.M., Garcia Ruiz, R.F.: Discovering nuclear models from symbolic machine learning. Communications Physics **8**(1), 101 (2025)
20. Pandey, B., Giri, S., Pant, R.D., Jalan, M., Chaudhary, A., Adhikari, N.P.: Prediction of binding energy using machine learning approach. AIP Advances **14**(10) (2024)
21. Wang, M., Huang, W.J., Kondev, F.G., Audi, G., Naimi, S.: The ame 2020 atomic mass evaluation (ii). tables, graphs and references. Chinese Physics C **45**(3), 030003 (2021)
22. Wang, N., Liu, M., Wu, X., Meng, J.: Surface diffuseness correction in global mass formula. Physics Letters B **734**, 215–219 (2014)
23. Weizsäcker, C.v.: Zur theorie der kernmassen. Zeitschrift für Physik **96**(7), 431–458 (1935)
24. Yüksel, E., Soydaner, D., Bahtiyar, H.: Nuclear mass predictions using machine learning models. Physical Review C **109**(6), 064322 (2024)