Taming nucleon density distributions with deep neural network method

Zu-Xing Yang\textsuperscript{1,2}∗, Wei Zuo\textsuperscript{1,2}†, Peng Yin\textsuperscript{1,3}, and Xiao-Hua Fan\textsuperscript{4}

\textsuperscript{1}Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China
\textsuperscript{2}School of Nuclear Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
\textsuperscript{3}Department of Physics and Astronomy, Iowa State University, Ames, IA 50011, USA
\textsuperscript{4}School of Physical Science and Technology, Southwest University, Chongqing 400715, China

We investigate the density distributions of finite nuclei employing a well-designed deep neural network method. We calculate the target nucleon density distributions with Skyrme density functional theories, which are used to train the networks. We find that the training with only about 10% nuclei (300 – 400) is sufficient to describe the nucleon density distributions of all the nuclear chart within 2% relative error. The relative error comes to 5% when about 200 proton(neutron) density distributions are used for training. We obtained very similar results for different Skyrme density functional theories. Therefore the ability to train networks is weakly dependent on the theoretical model. Moreover, in the process of machine learning, there is a turning point showing the transition from the Fermi-like distribution to the realistic Skyrme distribution, which provides significant properties of convergence process.

I. INTRODUCTION

Since the discovery of the atomic nucleus\textsuperscript{1}, density information has played a crucial role in the study of nuclear structure, especially in explaining the nature of nuclear force and the fundamental properties of nuclear matter\textsuperscript{2}. For example, in studies of high-momentum tails(HMT) induced by short-range correlations(SRC), it has been found that the percentage of tails is highly dependent on the density\textsuperscript{3–5}. The asymmetry potential is density-dependent, which is influenced by the Pauli blocking of the ∆-decay and the thickness of the neutron skin\textsuperscript{6}. Moreover, The equation of state(EoS) of nuclear matter is used to describe the neutron stars(NS) mergers\textsuperscript{7}.

In the 1950s, Hofstadter first measured the charge density of protons using electron scattering experiments and described the density distribution of some nuclei on that basis\textsuperscript{8}. Whereafter, two-parameter Fermi (2pF) / three-parameter Fermi (3pF) distributions\textsuperscript{9,11} or Fourier-Bessel expansions\textsuperscript{11,12} have been used to roughly describe the shape of nuclear density profile. The density functional theory (DFTs), e.g., Skyrme-Hartree-Fock (SHF) method, is one of the most efficient and widely used models to study the bulk properties of nuclei, especially the density distribution\textsuperscript{13–15}. With the creation of the Bardeen-Cooper-Schrieffer(BCS) theory\textsuperscript{16}, the understanding of superconductivity theory has been subverted and new blood has been injected into nuclear physics. Bogoliubov imported the generalized Bogoliubov transform to include both the particle-hole and particle-particle parts of nuclear force\textsuperscript{17}, which, of course, made for a more refined description of density. Currently, almost all models describing nuclear structure can calculate density distributions. In 2016, density information for the short-lived nucleus $^{34}$Si was also obtained by comparing the experimental measurements for the removal of an $l = 0$ proton with the results of reaction theories\textsuperscript{18}. Some simulations, based on transport models, also have a positive effect on the discovery of density profile. For example, using density profile as a initialization-condition, the bubble structure was researched by heavy ion collisions(HIC) simulation of p+$^{48}$Si\textsuperscript{19}.

Back propagation neural network (BPNN) method\textsuperscript{20–27}, which is the most popular and powerful machine learning tools, has been remarkably developed in computer vision (CV)\textsuperscript{28,29} and natural language processing (NLP)\textsuperscript{30,31} in recent years. In the application to the nuclear physics, it is also becoming prevailing. By training neural networks with the X-ray radiation from neutron stars, the relationship between pressure and mass density has been estimated\textsuperscript{32}. Combining the Hartree-Fock-Bogoliubov(HFB) and multilayer neural network methods, the nuclear complexity was tamed, where excited state energy, nuclear deformation etc. of nuclei are predicted\textsuperscript{33}. Excellent results have been obtained using Bayesian neural network (BNN) methods for the extrapolation of binding energies from stable nuclei to drip-line region\textsuperscript{34}. The artificial neural network method was used to extrapolate the ground state properties of nuclei calculated by ab-initio approach to large basis space\textsuperscript{35,36}. Until recently, Google proposed a hybrid quantum-classical machine learning model for training beyond classical data types\textsuperscript{37}. Inspired by these successful applications of the machine learning method, we briefly modify the traditional DNN model by using the determined mass number as a physical condition.

II. THE NEURAL NETWORK METHOD AND MACHINE LEARNING PROCESS

This work is aimed at mapping out the nucleon density distributions of almost all the nuclei employing the
neural network method by training networks with density distributions of very limited nuclei. This is a logistic regression problem in supervised learning, which statistically can be seen as a maximum likelihood estimation (MLE).  

\[
\begin{align*}
\mathbf{w}_{\text{MLE}} &= \arg \max_{\mathbf{w}} \log P(\mathbf{D}|\mathbf{w}) \\
&= \arg \max_{\mathbf{w}} \sum_{i} \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}),
\end{align*}
\]

where \(\mathbf{w}\) and \(\mathbf{w}_{\text{MLE}}\) denote the weights of neural networks during learning and the weights of neural networks where the mapping is implemented with the greatest probability, respectively. \(\mathbf{x}_i\) and \(\mathbf{y}_i\) represent the input and output, respectively. In this work, the input is \(\{Z_{\text{num}}, N_{\text{num}}, \tau\}\) where \(Z_{\text{num}}\) and \(N_{\text{num}}\) denote the proton and neutron numbers, respectively. \(\tau\) corresponds to the proton (\(\tau = 1\)) or neutron (\(\tau = 2\)). The output is the nucleon (\(\tau = 1\) for proton and \(\tau = 2\) for neutron) density distribution of the nucleus (\(Z_{\text{num}}, N_{\text{num}}\)). That is, the neural network is to produce the proton or neutron density distribution of a nucleus for a given set of \(\{Z_{\text{num}}, N_{\text{num}}, \tau\}\).

BPNN, which was proposed by Rumelhart and McClelland in 1986, is adopted to achieve Eq. (1). The method adjusts multi-layer feedforward neural network with back propagation algorithm of error, by which the weight of the link chain converges toward its stated goal \(\mathbf{w}_{\text{MLE}}\).

\textit{network structure} Neural networks typically contain a large number of interconnected neurons in the input layer, output layer, and several hidden layers. We used about 860,000 parameters located in eight layers (\(l \in [1, 8]\)),

\[
\begin{align*}
\hat{z}_i^{(l+1)} &= b_i^{(l)} + \sum_{j=1}^{S_l} w_{ij}^{(l)} \hat{a}_j^{(l)} , \\
a_i^{(l)} &= g(\hat{z}_i^{(l)}),
\end{align*}
\]

where \(a_i^{(l)}\) and \(\hat{z}_i^{(l+1)}\) denote the input and output of the \(l\)-th layer; \(w_{ij}^{(l)}\) and \(b_i^{(l)}\) are weight and bias of the \(l\)-th layer. In the input layer and each hidden layer (\(l \in [1, 7]\)), the nonlinear activation function \(g(x) = \text{ReLU}(x) = \max(0, x)\), which has been shown to have extraordinary effects\cite{9, 10}, is employed. In the output layer (\(l = 8\)), the activation function \(g(x) = \text{Sigmoid}(x) = \frac{1}{1+e^{-x}}\) is adopted to smooth the results.

The loss function is a key component in network regulation. The mean squared error (MSE) is widely used for regression problems, which can be obtained naturally by taking a normal distribution for \(P(\mathbf{D}|\mathbf{w})\) in Eq. (1). However, a physical constraint \(\int_{0}^{\infty} 4\pi \rho(r)r^2dr = X\) has to be taken into account in this work, with \(X = Z_{\text{num}}\) or \(N_{\text{num}}\). \(\rho(r)\) denotes the corresponding nucleon density distribution. We therefore introduce a correction to the MSE and define the normalized mean square error (NMSE) as follows,

\[
NMSE = \frac{\sum_{\alpha=1}^{M} \frac{1}{N} \sum_{i=1}^{N} (\lambda \rho_{\text{pre}}^{\alpha}(r_i) - \rho_{\text{tar}}^{\alpha}(r_i))^2}{\int_{0}^{\infty} 4\pi \rho_{\text{pre}}(r)r^2dr},
\]

where \(\alpha\) runs over all the nuclei which are adopted to train the networks. \(M\) corresponds to the number of nuclei in a training set. \(\iota\) runs over all the radial grids for a specified nucleus \(\alpha\). In this work we consider the radial range from \(r = 0\) fm to \(r = 15\) fm and divide \(r\) into \(N = 150\) grids for all the nuclei. \(\rho_{\text{pre}} = \mathbf{a}_i^{(8)}\) refers to the nucleon density distribution of a nucleus obtained with the current neural network method during the learning process, and \(\rho_{\text{tar}}\) denotes the target nucleon density distribution of the same nucleus, which is calculated with the Skyrme DFTs. With the correction in Eq. (3), absolute density becomes irrelevant. The conserved quantity allows our networks to be more targeted in capturing scale Features. The networks hence can be well-converged even with very limited training dataset.

Combining Eq. (3) with Eq. (1), mapping out the density distributions of a set of nuclei for a training set is equivalent to solving the following equation,

\[
\mathbf{w}_{\text{MLE}} = \arg \min_{\mathbf{w}} NMSE(\mathbf{w}).
\]

During the machine learning, we employ the adaptive moment estimation(Adam)\cite{11} for the optimizer to operate gradient descent. We manipulate the learning rate according to the variation in error at different epochs. We take the Keras with a Tensorflow backend\cite{42} to train the networks. The training results become well converged within 2000 epochs. One epoch represents that all the neurons have been trained once.

\textit{training process} About 3400 nuclei have been discovered in laboratories to date. Various theoretical approaches have been used to calculate the nucleon density distributions of these nuclei. In this section, the
We found that the NMSE and MSE decreases with epoch. For the training set, we use 10% of the entire dataset to train the networks as shown in Fig. 1. The red symbols in Fig. 1 corresponds to the nuclei which are not used in the training process. We present the locations of corresponding nuclei in Fig. 1. Note that we choose these 10% of the dataset randomly. Therefore both the proton and neutron density distributions of about 3400 nuclei. We calculate density using the following equation:

$$\rho_{\tau}(r) = \sum_{n,\beta, j, l, m} w_{\beta} \frac{2j + 1}{4\pi} \left( \frac{R_{\beta}}{r} \right)^2$$

(5)

where $w_{\beta}$ and $R_{\beta}$ denote the pairing weight and radial wave function for each single particle state, respectively. The total wave function reads:

$$\varphi_{\beta}(r) = \frac{R_{\beta}(r)}{r} \psi_{j, l, m}(\theta, \phi).$$

(6)

The functions $\psi_{j, l, m}(\theta, \phi)$ are spinor spherical harmonics.

In our first application, we train the networks with 10% of the entire dataset, formed by about 6800 sets of neutron/proton density distributions of about 3400 nuclei. We present the locations of corresponding nuclei in Fig. 1. Note that we choose these 10% of the dataset randomly. Therefore both the proton and neutron density distributions of some nuclei (blue symbols in Fig. 1) are adopted in these 10% data, whereas only one of the two nucleon density distributions (either the proton (yellow symbols in Fig. 1) or neutron density distribution (blue symbols in Fig. 1)) is employed for other adopted nuclei. The red symbols in Fig. 1 corresponds to the nuclei which are not used in the training process.

In Fig. 2 we present the loss function NMSE and MSE as functions of epoch, for both the training set and the validation set. For the training set, we use 10% of the entire dataset to train the networks as shown in Fig. 1. We found that the NMSE and MSE decreases with epoch simultaneously during the training process, which indicating that the results tend to be converged with networks trained sufficiently. Meanwhile, another 10% of entire dataset was evaluated after each training epoch as the validation test. It is clear that, both for NMSE and MSE, there is a high consistency of errors on training sets and validation sets. This shows that no under-fitting or over-fitting during the training process, which illustrates that the neural network method is able to describe the nucleon density distributions of the entire nuclear chart by training the networks with only 10% of the dataset. Note that NMSEs are much smaller than MSEs in Fig. 2 since we train the networks with the NMSE other than the MSE. We therefore treat MSE as an amplifier for NMSE. With the help of this amplifier, we notice that the two significant convergence processes emerge during the training process: the first one is at about 50 epochs and the second one is from 500 to 700 epochs.

What do the two convergences represent? In Fig. 3 we show the evolution of the neutron density distribution of a random nuclide $^{125}$In on the test set during the training process in Fig. 2. The first significant convergence brings the density distribution from chaos to order, which is achieved rapidly within around 50 epochs. The red block at 100th epoch, which is Fermi-like, is an exemplary distribution after the first significant convergence. The second significant convergence emerges from about 500th epoch, after which the Fermi-like distribution evolves to that calculated by the SKM*-SHF method. In order to make the value of the loss function as small as possible, the network naturally gets the Fermi-like distribution, which is like the process of human understanding of nuclear density profile in the past 70 years, from naive to refined. Epoch 500 is a crossover point, which is the...
networks for the SLy4-HFB model. Moreover, we use the dataset, which are all even-even nuclei, to train the networks for the theoretical model. One of the Skyrme methods using the SLy4 interaction, which we refer to as SLy4-HFB. As for the SKM*-SHF model, nine neural network models based on the SKM*-SHF and SLy4-HFB to guarantee their consistency.

We present in Fig. 4 the error distribution obtained with the nine neural network models based on the SKM*-SHF data training. The error distribution is the standard deviation between the result of the neural network method, \( \lambda_{\text{pre}}(r)_i \), and the result obtained by the SKM*-SHF method, \( \rho_{\text{tar}}(r)_i \), and is written as,

\[
\sigma(r) = \sqrt{\frac{\sum_{i=1}^{A} (\lambda_{\text{pre}}(r)_i - \rho_{\text{tar}}(r)_i)^2}{A}}.
\]

The subscript \( i \) runs over all 100 nucleon density distributions (\( A = 100 \)) which are randomly selected. We find in Fig. 4 that the maximum value of the error distribution is within 0.0015 fm\(^{-3}\) for the train set with 10% dataset. Comparing with the saturation density of the proton or neutron, about 0.08 fm\(^{-3}\) (half of the nuclear saturation density, 0.16 fm\(^{-3}\)), the central relative error (CRE) is less than 2%. For the train set with less than 10% dataset, we can see the error increases dramatically. For the case with 5% dataset, the predicted CRE is between 5% and 6%. However, even with only 2% of the dataset used for training, the predicted CRE is less than 10%. The predicted CRE reduces as more data are included in the train set.

Fig. 5 shows four random proton density distributions and four random neutron density distributions predicted by the trained neural-nets with different fractions-2%, 5%, 10% and 20% of dataset. It is clear that the results predicted by the models trained with 10% and 20% of the dataset overlap almost exactly with the curves obtained from SKM*-SHF calculations. Of particular note is the green curve, which is almost a standard Fermi distribution. There was no second convergence during the training for this curve, i.e. the W-point did not occur. This is acceptable because the training data at this point is rather small. This curve corresponds to a CRE of approximately 10%. This value can also be considered as the CRE at the W-point during the training with more than 2% of dataset. Naturally, the corresponding error curve (the 2% one) in Fig. 4 can also be viewed as the error curve at the W-point. The predictions of the model trained with 5% of the dataset are intermediate between the 2% one and the 10% one. It has undergone the second significant convergence in part and provides an error of only about 5%.

In Fig. 6 we present a quantity \( \sigma_{\text{tot}} \), which represents the normalized total standard deviation and is written as

\[
\sigma_{\text{tot}} = \sqrt{\frac{\sum_{j=1}^{N} \sigma(r_j)^2}{N}},
\]

where \( N \) denotes the number of radial grids. For the results in the right part of Fig. 6, \( \sigma_{\text{tot}} \) decreases as the amount of training data increase, which is consistent with the results in Fig. 4.

III. RESULTS AND DISCUSSIONS

In the following applications, we use two Skyrme DFTs to calculate the target nucleon density distributions in order to investigate the dependence of the neural network predictions on the theoretical model. One of the Skyrme DFTs is the SKM*-SHF as we mentioned in the above section. The other one is the Hartree-Fock-Bogoliubov method using the SLy4 interaction, which we refer to as SLy4-HFB. As for the SKM*-SHF model, nine neural network models with 2%, 5% and 10%-70% of the dataset adopted to train the networks are used to investigate the convergence pattern. However, we take 10% of the dataset, which are all even-even nuclei, to train the networks for the SLy4-HFB model. Moreover, we use identical inputs for the training processes with SKM*-SHF and SLy4-HFB to guarantee their consistency.
FIG. 5. (Color online) The proton density distribution of four random nuclei: $^{55}$Ti(a), $^{114}$Nb(b), $^{154}$Nd(c), $^{212}$Bi(d) and the neutron density distribution of four random nuclei: $^{43}$Ca(e), $^{104}$Rh(f), $^{164}$Ta(g), $^{208}$At(h) in test set predicted by the trained neural-net with different fractions of dataset, and compared with original data from SKM*-SHF.

FIG. 6. (Color online) The error $\sigma_{\text{tot}}$ of predicted density distributions as a function of different percentages of the dataset with SKM*-SHF and SLy4-HFB.

In the left part of Fig. 6, we investigate the model dependence of the current neural network method. We randomly selected 340 nucleon density distributions of even-even nuclei to train the networks using the target nucleon density distributions calculated by the SKM*-SHF and SLy4-HFB methods. We notice that $\sigma_{\text{tot}}$ for the results obtained with the SKM*-SHF and SLy4-HFB methods are on the same order of magnitude. Therefore the predictive power of the current neural network method is weakly dependent on the theoretical models and interactions. Comparing the data for 10% of dataset in the left part of Fig. 6 with that in the right part, we find that $\sigma_{\text{tot}}$ for even-even nuclei is smaller than the result which does not distinguish the odd/even status of the proton or neutron numbers. This can be explained in terms of the complexity of information. Dataset with only even-even nuclei have lower complexity than the other case. During the training process, low complexity contains less features, which are easier to captured. Therefore, the neural network calculations with only even-even nuclei dataset lead to smaller $\sigma_{\text{tot}}$.

IV. SUMMARY

In this work we proposed a method to describe the nucleon density distributions of finite nuclei using neural network method. Due to the normalized condition in the loss function, the nucleon density distribution was always Fermi-like in the early stage of the machine learning and then tended to the target distribution after the second significant convergence. We therefore defined a W-point at which the second significant convergence emerged. We
found that the W-point emerged earlier with more data adopted in the training process. We did not observe the W-point for the machine learning with 2% of dataset.

We found that the machine learning with 10% of dataset was sufficient to predict the nucleon density distributions of all the nuclei with high precision. The corresponding CRE was less than 2%, which suggested that the neural network method adopted in this work was rather successful in the current application. By comparing the results using the target distributions calculated with two different Skyrme DFTs, SKM*-SHF and SLY4-HFB, we found that the predictive power of the neural network was weakly dependent on the adopted theoretical model.

So far, the ground-state density distribution of spherical nuclei has been adequately studied. Compared to traditional methods, the neural network method not only reduces the complexity of the study, allowing one to avoid complicated many-body problem, but also significantly improves the predictive power with less computational cost.

V. ACKNOWLEDGEMENTS

We acknowledge helpful discussions with Gao-Chan Yong and Jian-Min Dong. This work was supported by the National Natural Science Foundation of China (Grant Nos. 11435014, 11705240, 11975282) and the 973 Program of China (Grant No. 2013CB834405). This work was also partially supported by the CUSTIPEN (China-U.S. Theory Institute for Physics with Exotic Nuclei) funded by the U.S. Department of Energy, office of Science under Grant No. DE-SC0009971.

[1] E. Rutherford, Phil. Mag. 21, 669-688 (1911)
[2] A. R. Abdulghany, arXiv:2004.09727
[3] Z. X. Yang, X. L. Shang, G. C. Yong, W. Zuo, Y. Gao, Phys. Rev. C 100, 054325 (2019)
[4] P. Yin, J. Y. Li, P. Wang and W. Zuo, Phys. Rev. C 87, 014314 (2013).
[5] P. Yin, J. Dong and W. Zuo, Chin. Phys. C 41, 114102 (2017)
[6] C. Hartnack, A. Le Fevre, Y. Leifels, J. Aichelin, arXiv:1808.09868
[7] C. J. Horowitz, Ann. Phys. 411, 167992 (2019)
[8] R. Hofstadter, Rev. Mod. Phys. 28, 214-254 (1956)
[9] D. R. Yennie, D. G. Ravenhall, R. N. Wilson, Phys. Rev. 95, 500 (1954).
[10] M. A. Preston, Physics of the nucleus (Addison-Wesley, Reading (Mass.), 1962).
[11] N. Ullah, Pramana 43 No.2, 165-168(1994)
[12] H. Euteneuer, J. Friedrich and N. Voegler, Nucl. Phys. A 298, 452 (1978).
[13] T. Naito, X. Roca-Maza, G. Colo’, H. Z. Liang, Phys. Rev. C 99, 043003 (2019)
[14] Y. Zhang, Y. Chen, J. Meng, P. Ring, Phys. Rev. C 95, 014316 (2017)
[15] C. A. Bertulani, J. Valencia, Phys. Rev. C 100, 015802 (2019)
[16] J. Bardeen, L. N. Cooper, J. R. Schrieffer, Phys. Rev. 106, 162-164
[17] N. N. Bogoliubov, Soviet Physics JETP-USSR, 7, 41-46(1958)
[18] A. Mutscher et al., Nature Physics, Nature Publishing Group, 13(2), pp.142-146(2017)
[19] X. H. Fan, G. C. Yong, W. Zuo, Phys. Rev. C 99, 041101(R) (2019).
[20] B. B. Jiao, International Journal of Modern Physics E, 2050024(2020)
[21] I. Kaastra, M. Boyd, Neurocomputing 10 215 (1996)
[22] A. Parlagy, Y. Islamoglu, H. Yasar et al, Appl. therm. eng. 26 824 (2006)
[23] G. Jing, W. Du, Y. Guo, Desalination 291 78 (2012)
[24] B. H. M. Sadeghi, J. Mater. Process. Technol. 103 411 (2000)
[25] L. L. Zhao, Y. X. Yang, Y. M. Zheng et al, Journal of East China Institute of Technology 36 79 (2013)
[26] H. Z. Xu, S. Wu, N. Liu, Journal of East China Institute of Technology 3 293 (2008)
[27] X. X. Cheng, X. Y. Cheng, C. J. Liu, Journal of East China Institute of Technology 39 172 (2016)
[28] S. S. Mohanta, C. Assisi, arXiv:1906.03958
[29] P. Grohs, T. Wiatowski, H. Bölcskei, Proc. of IEEE International Symposium on Information Theory (ISIT), 1163-1167(2016)
[30] J. H. Mao, W. X. Y. Yang, J. Wang, Z. H. Huang, A. Yuille, International Conference on Learning Representations(ICLRR)(2015)
[31] J. Legrand, R. Collobert, International Conference on Learning Representations(ICLRR) (2015)
[32] Y. Fujimoto, K. Fukushima, K. Murase, Phys. Rev. D 101, 054016 (2020)
[33] R. D. Lasser, D. Regnier, J. P. Ebran, A. Penon, Phys. Rev. Lett. 124, 162502 (2020)
[34] J. D. Holt, S. R. Stroberg, A. Schwenk, J. Simonis, arXiv:1905.10475
[35] G. A. Negoita, G. R. Luecke, J. P. Vary, P. Maris, A. M. Shirotok, I. J. Shin, Y. Kim, E. G. Ng and C. Yang, Proceedings of the Ninth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking COMPUTATION TOOLS 2018 February 18-22, 2018, Barcelona, Spain.
[36] G. A. Negoita, J. P. Vary, G. R. Luecke, P. Maris, A. M. Shirotok, I. J. Shin, Y. Kim, E. G. Ng and C. Yang, Proceedings of the Ninth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking COMPUTATION TOOLS 2018 February 18-22, 2018, Barcelona, Spain.
[37] M. Broughton et al., TensorFlow Quantum: A Software Framework for Quantum Machine Learning, arXiv:2003.02989
[38] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, International Conference on Machine Learning (ICML)(2015)
[39] A. J. R. Simpson, arXiv:1509.05173 MSC Class: 68Txx
[40] MSC Classification Codes https://cran.r-project.org/web/classifications/MSC.html
[41] D. Kingma, J. Ba, International Conference on Learning Representations(ICLRR) (2015)
[42] M. Abadi et al, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[43] P. G. Reinhard, “The Skyrme-Hartree-Fock Model of the Ground State,” in *Computational Nuclear Physics 1: Nuclear Structure*, edited by K. Langanke, J. A. Maruhn, and S. E. Koonin, SpringerVerlag, New York, 1991, pp. 29-50.

[44] D. Filan, S. Hod, C. Wild, A. Critch, S. Russell, arXiv:2003.04881