

Analysis of a Skyrme energy density functional with deep learningN. Hizawa ¹, K. Hagino ¹, and K. Yoshida ^{2,3,4}¹*Department of Physics, Kyoto University, Kyoto 606-8502, Japan*²*Research Center for Nuclear Physics, Osaka University, Ibaraki, Osaka 567-0047, Japan*³*RIKEN Nishina Center for Accelerator-Based Science, Wako, Saitama 351-0198, Japan*⁴*Center for Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*

(Received 20 June 2023; accepted 10 August 2023; published 21 September 2023)

Over the past decade, machine learning has been successfully applied in various fields of science. In this study, we employ a deep learning method to analyze a Skyrme energy density functional (Skyrme-EDF), which is a Kohn-Sham type functional commonly used in nuclear physics. Our goal is to construct an orbital-free functional that reproduces the results of the Skyrme-EDF. To this end, we first compute energies and densities of a nucleus with the Skyrme Kohn-Sham + Bardeen-Cooper-Schrieffer method by introducing a set of external fields. Those are then used as training data for deep learning to construct a functional which depends only on the density distribution. Applying this scheme to the ²⁴Mg nucleus with two distinct random external fields, we successfully obtain a new functional which reproduces the binding energy of the original Skyrme-EDF with an accuracy of about 0.04 MeV. The rate at which the neural network outputs the energy for a given density is about 10⁵–10⁶ times faster than the Kohn-Sham scheme, demonstrating a promising potential for applications to heavy and superheavy nuclei, including the dynamics of fission.

DOI: [10.1103/PhysRevC.108.034311](https://doi.org/10.1103/PhysRevC.108.034311)**I. INTRODUCTION**

Recent progress of deep learning is quite remarkable. It has actually gained popularity in various fields of science and technology, such as natural language processing, computer vision, and speech recognition [1–5]. In several fields of physics, such as condensed-matter physics, a multitude of ideas to utilize machine learning are arising. For instance, in Ref. [6], an energy density functional (EDF) for electron systems that depends solely on an electron number density was constructed using the method developed in Ref. [7], in which an attempt was made with a neural network to predict the solution of a two-dimensional Schrödinger equation in a random potential. Other applications have already existed also for a variety of problems, including those in spin systems [8] and superconducting systems [9].

In contrast, an application of deep learning to nuclear physics has still been in its early stage [10–26]. We mention that nuclear physics continues to face numerous unresolved challenges that call for innovative solutions, including a description of large amplitude collective motions. Such problems may be solved efficiently by applying the machine-learning techniques developed in other fields of physics.

In particular, the recent application of deep learning to the Kohn-Sham type DFT [6] mentioned above could be readily applied also to nuclear physics. In nuclear physics, phenomenological models for a functional have often been employed [27]. The resultant Kohn-Sham type energy density functional (KS-EDF) is not an explicit functional of the particle number density only, but is parametrized together with other local densities, such as the kinetic-energy density,

the spin-orbit density, and the pair density when considering explicitly the nucleonic superfluidity. To calculate observables using the KS-EDF, such as the binding energy of a nucleus, one needs to solve a self-consistent differential equation of the same form as that in the mean-field theory many times, which is computationally expensive, especially for heavy systems. Therefore, it is desirable to develop an orbital-free EDF (OF-EDF) theory that depends solely on the particle number density, without explicitly depending on Kohn-Sham orbitals and other local densities such as the kinetic and the spin densities. Deep learning can be a powerful tool for that purpose [6]. Such theory will be based on a functional that depends solely on the particle number density. Notice that this is totally consistent with the original philosophy of density-functional theory (DFT).

The aim of this paper is to apply the method developed in Ref. [6] to a nuclear system and construct a deep-learning-based nuclear OF-EDF that reproduces results of the Skyrme-EDF. In applying the method of Ref. [6] to a nuclear system, one has to take into account several aspects that make nuclear systems different from electron systems. One obvious difference is that a nucleus is a self-bound attractive system. In an electron system without phonons, the only interaction between electrons is the repulsive Coulomb force, which causes two electrons to distribute as far as possible. In marked contrast, nucleons tend to get closer to each other due to a short-ranged attractive nuclear force, and thus the mechanism which determines the density distribution is quite different between electron and nucleon systems [28]. In addition, for electron systems, the KS-EDFs, which are inspired by the Hartree-Fock method, works well in

general. On the other hand, in nuclear systems, superfluidity plays a crucial role in open-shell nuclei, and observables are better explained using a KS-EDF that is inspired by the Hartree-Fock-Bardeen-Cooper-Schrieffer (BCS) or Hartree-Fock-Bogoliubov method rather than by the Hartree-Fock method. This leads to a technical difference in that a nuclear KS-EDF depends also on the pair density. It will be intriguing to investigate how well the deep learning method works in such attraction-dominated nuclear systems.

This paper is organized as follows: In Sec. II, we introduce the KS-EDF which we employ, and define a protocol for deep learning. We also discuss how to generate data sets to train neural networks. In Sec. III, we carry out the deep learning for the ^{24}Mg nucleus and discuss how well the data sets can be learned. We then summarize the paper in Sec. IV and discuss future perspectives.

II. FORMULATION

A. Skyrme energy density functional

We first introduce a Kohn-Sham type energy density functional (KS-EDF) for training on a neural network. Throughout this study, we consistently employ the following Skyrme-type EDF [29]:

$$E_{\text{tot}} = E_{\text{kin}} + E_{\text{int}} + E_{\text{pair}} + E_{\text{CoM}}, \quad (1)$$

with

$$E_{\text{kin}}[\tau] = \frac{\hbar^2}{2m} \left(1 - \frac{1}{A}\right) \sum_q \int d^3r \tau_q(\mathbf{r}), \quad (2)$$

$$\begin{aligned} E_{\text{int}}[\rho, \tau, \mathbf{J}] = & \int d^3r \left\{ \frac{b_0}{2} \rho^2 - \frac{b'_0}{2} \sum_q \rho_q^2 + \frac{b_3}{3} \rho^{\alpha+2} \right. \\ & - \frac{b'_3}{3} \rho^\alpha \sum_q \rho_q^2 + b_1 \rho \tau - b'_1 \sum_q \rho_q \tau_q \\ & - \frac{b_2}{2} \rho \nabla^2 \rho + \frac{b'_2}{2} \sum_q \rho_q \nabla^2 \rho_q \\ & \left. - b_4 \rho \nabla \cdot \mathbf{J} - b'_4 \sum_q \rho_q \nabla \cdot \mathbf{J}_q \right\}, \quad (3) \end{aligned}$$

$$E_{\text{pair}}[\rho, \tilde{\rho}] = \sum_q \frac{V_0^{(q)}}{4} \int d^3r \left\{ 1 - \left(\frac{\rho}{\rho_0}\right)^\gamma \right\} \tilde{\rho}_q^2, \quad (4)$$

and

$$E_{\text{CoM}}[\rho] = \frac{C}{2} \left(\int d^3r z \rho(\mathbf{r}) \right)^2, \quad (5)$$

where m is the nucleon mass and A is the mass number of a nucleus. E_{kin} , E_{int} , E_{pair} , and E_{CoM} are the kinetic energy, the interaction energy, the pairing energy, and a cost function for the center of mass, respectively. ρ , τ , \mathbf{J} , and $\tilde{\rho}$ are the particle number density, the kinetic density, the spin density, and the pair density, respectively, in which the subscript q refers to

neutron or proton. Those are defined as

$$\rho(\mathbf{r}) = 2 \sum_q \sum_{k>0} v_{q,k}^2 |\varphi_{q,k}(\mathbf{r})|^2, \quad (6)$$

$$\tau(\mathbf{r}) = 2 \sum_q \sum_{k>0} v_{q,k}^2 |\nabla \varphi_{q,k}(\mathbf{r})|^2, \quad (7)$$

$$\mathbf{J}(\mathbf{r}) = 2 \sum_q \sum_{k>0} v_{q,k}^2 \varphi_{q,k}^*(\mathbf{r}) (-i \nabla \times \boldsymbol{\sigma}) \varphi_{q,k}(\mathbf{r}), \quad (8)$$

$$\tilde{\rho}(\mathbf{r}) = -2 \sum_q \sum_{k>0} u_{q,k} v_{q,k} |\varphi_{q,k}(\mathbf{r})|^2, \quad (9)$$

where $\varphi_{q,k}(\mathbf{r})$ is the k th Kohn-Sham orbital in a spinor form with isospin q , and $v_{q,k}^2 = 1 - u_{q,k}^2$ is the occupation probability for the k th orbital. Notice that we take the BCS approximation for the treatment of the pairing correlation.

In the interaction part of the functional, b_i and b'_i ($i = 1-4$) as well as α are the Skyrme parameters. In this paper, we use the SLy4 parameter set [30] for these parameters. For simplicity, we ignore the Coulomb interaction, even though the entire Coulomb interaction term can be explicitly described as a functional of the proton number density if the Slater approximation is introduced to the exchange term.

For the pairing part, we employ a surface-type functional of the density-dependent delta-interaction (DDDI) [31] [see Eq. (4)], which contains the parameters $V_0^{(q)}$, ρ_0 , and γ . In this study, we take $\gamma = 1$ and $\rho_0 = 0.16 \text{ fm}^{-3}$ and determine $V_0^{(q)}$ so that the average pairing gap coincides with the empirical pairing gap, $\Delta_q = 12/\sqrt{A} \text{ MeV}$ [29,32]. The zero-range pairing interaction has to be supplemented with an energy cutoff. In this paper, the sharp cutoff energy of 60 MeV is introduced to the single-particle energy of the Kohn-Sham orbitals. The resultant strengths for the pairing are $V_0^{(n)} = V_0^{(p)} = -683.344 \text{ MeV fm}^3$.

In addition to the ordinal Skyrme EDF, we introduce a functional $E_{\text{CoM}}[\rho]$ to fix the center-of-mass position in the z direction, even though this can be done also by constraining the $Q_{10} = \sum_i z_i$ operator. This is necessary because we introduce external fields (see Sec. II C below) to generate various density distributions. By fixing the center-of-mass position, one can prevent a nucleus from localizing around the edges of the box, which is useful to generate various deformed states in a small box. In this study, we take 0.625 MeV/fm^2 for the value of C .

In this paper, we consider only the ^{24}Mg nucleus. This choice of a nucleus is convenient, because this nucleus has equal numbers of protons and neutrons, and thus the proton and the neutron densities coincide with each other when the Coulomb interaction is ignored. Furthermore, we impose the axial symmetry and the time-reversal symmetry on the system, enabling the local densities to be expressed in the cylindrical coordinates (r, z) [33]. Notice that Ref. [6] also used a two-dimensional EDF for electron systems. With these simplifications, in principle, the EDF of the system should be able to be expressed solely with the nucleon number density $\rho(r, z)$, which can be considered as a monochromatic image.

We solve the Kohn-Sham equations for this EDF by introducing various external fields to obtain a set of ground-state

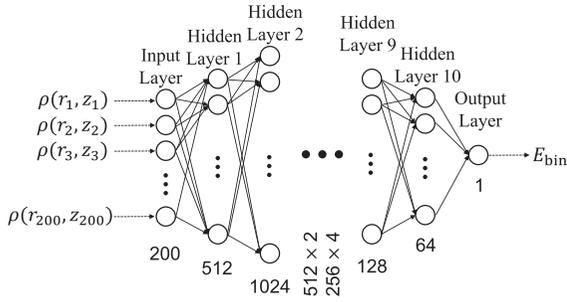


FIG. 1. A neural network employed in this work to learn the Skyrme-EDF, $E[\rho]$. It consists of 10 hidden layers, all of which are fully connected. Their activation functions are the ReLU, and the sigmoid activation function is employed for the output layer. The number of neurons in each layer is listed below the layers.

energies and nucleon number densities. The explicit forms of the external fields are specified in Sec. II C. We solve the Khon-Sham equations by discretizing the real space, with the mesh size of 0.8 fm in both the r and z directions, which is a standard choice in solving the Kohn-Sham-Bogoliubov equation in the real space, as in Ref. [34] and references cited therein. We take 10 grid points in the r direction and 20 points in the z direction, with which the density $\rho(r, z)$ can be considered as a 10×20 -dimensional vector in our calculations. We choose the box boundary condition and include the z component of angular momentum up to $9/2$. Without the external potentials, the ground state of ^{24}Mg becomes spherical in our setup with the binding energy of -222.59 MeV.¹

B. Neural network

In this paper, we carry out a regression analysis of $E = E[\rho]$ using a set of the particle number density and the energy $D = \{E^{(i)}, \rho^{(i)}\}_i$ generated by the KS-DFT. To this end, we utilize a neural network with fully connected layers for the fitting function. The fundamental structure of a neural network involves a repetition of linear and nonlinear transformations on the input vector; fully connected layers signify that all the neurons in the previous layer are connected to all the neurons in the next layer.

We mention that neural networks composed solely of fully connected layers may encounter an issue of an excessive number of parameters when the dimension of an input vector is large. To avoid this problem, a convolutional neural network (CNN) is often employed, which has demonstrated remarkable success in the field of computer vision [35,36]. In fact, in the previous application of deep learning to KS-DFT [6], the input size of a vector has as large as 256×256 dimension, and thus a CNN was employed. However, the dimension of our studies in this paper is much smaller, with a 10×20 dimension. Therefore, we do not need to introduce the CNN, and a simpler neural network consisting of the fully connected

layers, as depicted in Fig. 1, is employed in this study (see the caption for the details).

We use the Adam optimizer [37], which has several tunable parameters. Among these parameters, we set a learning rate to be 10^{-4} and the others to be default value of the Keras API [38].² The batch size is 128, namely, we divide training data into subsets, each of which contains 128 components. In each update of the fitting parameters, we do it only within each subset to minimize a loss function, for which we take a mean square loss function. To avoid the problem of overfitting, we adopt the early stopping strategy and stop the learning at the 500th epoch. We decrease the learning rate sequentially to 10^{-5} (at epoch = 101), 10^{-6} (at epoch = 201), 5.0×10^{-7} (at epoch = 301), and 10^{-7} (at epoch = 401).

C. External fields

For a given EDF, one can make a correspondence between the particle number density and the energy of the ground state for a specific external field. This property will be used to construct a data set to be trained for an OF-EDF. For this purpose, a diverse range of external fields is required. In this section, we introduce two methods to generate the external potentials used in this study. The basic idea of these methods is adapted from the previous studies [6,7] on two-dimensional systems, but we modify them for the axial-symmetric systems.

1. Simple harmonic oscillators

The first method is to use external fields based on a simple harmonic oscillator (SHO). As the name implies, this is a deformed harmonic-oscillator potential shifted in the z direction:

$$v_{\text{SHO}}^{(i)}(r, z) = \frac{1}{2}k_r^{(i)}r^2 + \frac{1}{2}k_z^{(i)}(z - z_0^{(i)})^2. \quad (10)$$

The parameters in the range of $0 \leq k_r, k_z \leq 1.1$ MeV/fm², and -1.6 fm $\leq z_0 \leq 1.6$ fm in the potential are generated from uniform random parameters.³

The SHO potentials would be able to encompass only a small portion of a domain of the external fields to be used in the Skyrme-EDF. However, for practical calculations, only a limited variety of external fields, such as a quadrupole moment, has frequently been utilized, if a constrained field is regarded as an external field in a broad sense. It is therefore still useful to examine the effectiveness of the learning process with the SHO potentials.

2. Random potentials

As the second method, we introduce a random potential (RND). In this regard, we mention that a primary purpose of introducing external fields in this study is to generate various types of density distributions ρ . It is highly nontrivial to generate external fields which efficiently lead to a multitude of densities with low binding energies, as one does not know

¹We have confirmed that a smaller pairing strength yields a prolate shape.

²We have also tested the Rectified Adam optimizer (RAdam) [39], but results were not improved.

³Unless otherwise noted, all the random numbers used in this paper are uniform random numbers.

a priori the relation between a distribution of ρ and that of external fields. While Gaussian processes (GPs) [40] are certainly one of the powerful tools for this purpose, in this paper we follow the idea of Refs. [6,7] to generate random potentials as follows.

These are highly random potentials with many random numbers:

$$v_{\text{RND}}^{(i)}(r, z) = m(r, z) \times \text{sr}^{(i)}(r, z), \quad (11)$$

where $m(r, z)$ and $\text{sr}^{(i)}(r, z)$ are defined as,

$$m(r, z) = \exp\left(-4.0 \max\{0, \sqrt{r^2 + z^2} - r_0\}^2 / r_0^2\right), \quad (12)$$

and

$$\text{sr}^{(i)}(r, z) = \sum_{r', z'} s^{(i)}(r, z; r', z') \text{rnd}^{(i)}(r', z'), \quad (13)$$

respectively, with

$$s^{(i)}(r, z; r', z') = \exp\left[-\{(r - r')^2 + (z - z')^2\} / \mu_2^{(i)}(r', z')\right]. \quad (14)$$

The meaning of $\text{rnd}^{(i)}(r, z)$ in Eq. (13) and $\mu_2^{(i)}(r, z)$ in Eq. (14) is as follows: First, for each grid point (r, z) , a random number within the range of $[v_{\min}, v_{\max}]$ is generated and labeled as $\text{rnd}^{(i)}(r, z)$. Since the potential with those random numbers is too irregular to be used as a potential, it is smoothed with a Gaussian filter, denoted as $s^{(i)}$, as in Eq. (13). At this stage, the square of the Gaussian width $\mu_2^{(i)}(r, z)$ in Eq. (14) is randomly generated within the range of $[\mu_{2\min}, \mu_{2\max}]$ to prevent the external field from acquiring scale information due to the standard deviation of the Gaussian. Finally, a mask defined by Eq. (12) is applied in Eq. (11) to circumvent a numerical instability caused by a reduction of the external field near the boundary. In this study, we take $r_0 = 1.4 \times 1.2A^{1/3}$ fm, $v_{\min} = -1.1$ MeV, $v_{\max} = 1.1$ MeV, $\mu_{2\min} = 0.8$ fm², and $\mu_{2\max} = 1.2$ fm², which are determined to obtain a large number data for ²⁴Mg with excitation energies below 5 MeV.

In Refs. [6,7], random $\{0, 1\}$ binary data were utilized for $\text{rnd}^{(i)}(r, z)$. For electronic systems, such a choice would be plausible because the potential primarily arises from the Coulomb potential due to a nucleus. On the other hand, in nuclear systems, it would be a highly nontrivial question to ask which potential is useful to describe static and dynamical properties of atomic nuclei. While many calculations employ a phenomenological deformed mean-field potential with, e.g., a quadrupole deformation to study deformed nuclei, it is not obvious whether such choice is optimal. Therefore, in this study, we use random real numbers for $\text{rnd}^{(i)}(r, z)$ to generate more diverse external fields than in the previous studies. Additionally, since the constraint on the center-of-mass position is included in the definition of the KS-EDF (1), a different mask function m from that in the previous studies is also introduced.

III. RESULTS

A. Generation of a dataset

Let us now apply the deep learning protocol discussed in the previous section to the Skyrme EDF. We first

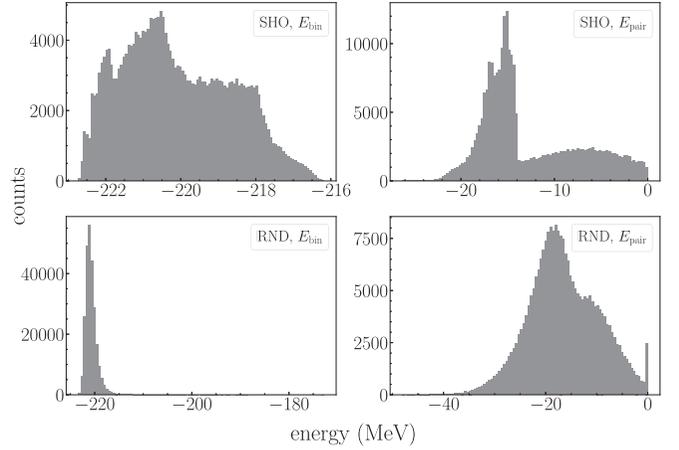


FIG. 2. Histograms for the results of the Skyrme EDF calculations with 250 000 different external fields based on the SHO (the first row) and the RND (the second row) fields. The left and right panels show the binding energy and the pairing energy in units MeV, respectively. It can be observed that the structure in the shape of the histograms is washed out to a large extent for the RND external fields, which are more random than the SHO cases.

prepare 250 000 data sets for each of the SHO and the RND external fields. For each calculation, the outputs are (i) the nucleon number density ρ , (ii) the kinetic energy E_{kin} , (iii) the interaction energy E_{int} , (iv) the pairing energy E_{pair} , and (v) the energy for the external field E_{ex} . The binding energy E_{bin} is also computed as a sum of E_{kin} , E_{int} , and E_{pair} , as $E_{\text{bin}} = E_{\text{kin}} + E_{\text{int}} + E_{\text{pair}}$. Figure 2 displays the distribution of E_{bin} and E_{pair} for each of the SHO and the RND external fields. The distributions of the other components of the energy are summarized in Appendix A (see Fig. 8). To use these data for deep learning, we reject those outside the regions given in Table I to remove outliers. Following Ref. [6], we randomly select 200 000 data for dataset from the remaining data. Out of those 200 000 data, we randomly adopt 90% of them for training data, while the rest for test data, which are not used for training.

TABLE I. The lower and the upper cutoff energies, in units MeV, for the two different types of the external fields, SHO and RND. The numbers in the parentheses denote the number of data after each cutoff is implemented for the original 250 000 data. For each learning, only the data within the intervals are employed. The value of cutoffs are determined so that approximately all the data shown in Figs. 2 and 8 can be included.

Type	SHO		RND	
	Lower	Upper	Lower	Upper
E_{bin}	$-\infty$	-217.5 (239 827)	$-\infty$	-217.5 (245 740)
E_{kin}	395.0	450.0 (249 491)	360.0	420.0 (249 310)
E_{int}	-650.0	-600.0 (248 782)	-630.0	-550.0 (249 398)
E_{pair}	-22.0	$+\infty$ (248 911)	-35.0	$+\infty$ (249 112)
E_{ex}	$-\infty$	120.0 (249 704)	-70.0	50.0 (249 468)

B. $\rho \rightarrow E[\rho]$

We first discuss the results for each energy as an objective variable with the nucleon number density as an explanatory variable. In other words, we construct the OF-EDF, which yields the energies from a density distribution as an input. In DFT, apart from an external field, there would be an ambiguity to divide the functional into components: E_{kin} , E_{int} , and E_{pair} themselves may not have strict physical meanings. Nevertheless, these components can be employed as indicators at least for qualitative discussions, and we thus follow Ref. [6] to examine the subparts of the EDF. In particular, it is interesting to investigate the pairing energy E_{pair} because it qualitatively verifies whether we can learn the effect of superfluidity, or the pair density, with deep learning. Notice that this was not addressed in the previous study in Ref. [6].

The panels in the first row in Fig. 3 show the difference between the results of the Kohn-Sham method and the neural network predictions for the test data with the RND external fields not used in learning. The results with the SHO external fields (see the panels in the third row) are found to be more accurate.⁴ Figure 3 shows only E_{bin} , E_{pair} , and E_{ex} , while the other components of the energy are displayed in Appendix A (see Fig. 9). One can see that our neural network can predict most of the data except for the energy of the external fields plotted in the rightmost figure.

We have found that the large error in E_{ex} was not improved by changing the learning method, such as a CNN model (see Appendix). This may be due to the fact that the particle number densities with different external fields tend to have a similar shape because of the saturation property, which results in an information loss in the process of compressing information on the external fields into the density distributions. Of course, according to the principle of DFT, ideally there should be no loss of information because there is a bijection between an appropriately defined density and an external field. However, in actual calculations, information on the detailed structure of external fields may be lost due to several numerical errors such as rounding errors, finite difference errors, and errors associated with a convergence criterion in self-consistent calculations. It is then natural that the prediction error becomes large when one attempts to recover the external field information from such a density distribution. The inaccuracy in predicting the energy of external fields was reported also in the previous study [6], but the inaccuracy seems more pronounced in atomic nuclei, which are systems with an attractive interaction. As we will show in the next section, this problem can be improved by using the external fields as explanatory variables.

To quantitatively evaluate the errors, we calculate the mean absolute errors (MAEs) for each learning, which are summarized in Table II. It is remarkable that the MAE for the binding energy is as small as 0.0051 MeV for the SHO external fields and 0.0433 MeV for the RND external fields, which are much more accurate than the accuracy required, e.g., for a fission

TABLE II. The mean absolute error (MAEs) for each learning with the SHO and the RND external fields. The units are MeV for $E[\rho]$ and $E[v]$, while the MAE for $\rho[v]$ is dimensionless [see Eq. (16)].

Type	SHO		RND	
	$E[\rho]$	$E[v]$	$E[\rho]$	$E[v]$
E_{bin}	0.0051	0.0054	0.0433	0.0237
E_{kin}	0.0165	0.0071	0.1131	0.0900
E_{int}	0.0105	0.0182	0.0431	0.1499
E_{pair}	0.0233	0.0261	0.1567	0.1411
E_{ex}	0.0318	0.0105	6.6973	0.1338
	$\rho[v]$		$\rho[v]$	
	0.1107		0.4101	

barrier of heavy nuclei as well as for nuclear masses. For instance, for the latter, the accuracy of 100 keV is required for the r-process studies [41]. The MAE for the pairing energy is 0.0233 MeV for the SHO and 0.1567 MeV for the RND. These values indicate that the particle number density predicts well the contribution of the pairing correlation, even though the error is slightly larger than that for the binding energy.

Finally, let us discuss a computational time. For the ^{24}Mg nucleus, it typically takes about a minute to solve the Skyrme-EDF with the Kohn-Sham method and obtain a single training data point. In marked contrast, the time to predict the energy with the neural network used in this paper from a given density is much shorter, about 0.1 ms. The difference in the computational speed will become larger for heavy nuclei. This makes a great advantage of using the deep learning method, e.g., in plotting a multidimensional potential-energy surface for nuclear fission studies of heavy nuclei.

C. $v \rightarrow E[v]$

While it is somewhat tangential to the topic of DFT, there is a certain demand in electronic systems for a functional that directly predicts the energy from a given external field. Because of this, in the previous study [6], an energy functional $E[v]$ was constructed following the same procedure as that to construct a functional $E[\rho]$. Even though it is unclear whether such a functional is useful in nuclear physics, it may be worth investigating whether a functional $E[v]$ can be constructed in connection to the discussion in Ref. [6]. We therefore carry out similar calculations using the same neural network and dataset as those in the previous section, but with the external fields as the explanatory variables.

The MAEs for $E[v]$ are summarized in Table II, which shows that the MAE for $E[v]$ tends to be decreased compared with that for $E[\rho]$. This is because the external field contains more information than the density distribution. This is particularly true for learning the energy from the external fields. On the other hand, the accuracy gets lowered for the binding energy with the SHO external fields. To investigate the origin for this, the panels in the second and the fourth rows in Fig. 3 show the differences between $E[v]$ from the Skyrme KS

⁴This is a natural outcome from the simplicity of the SHO external fields.

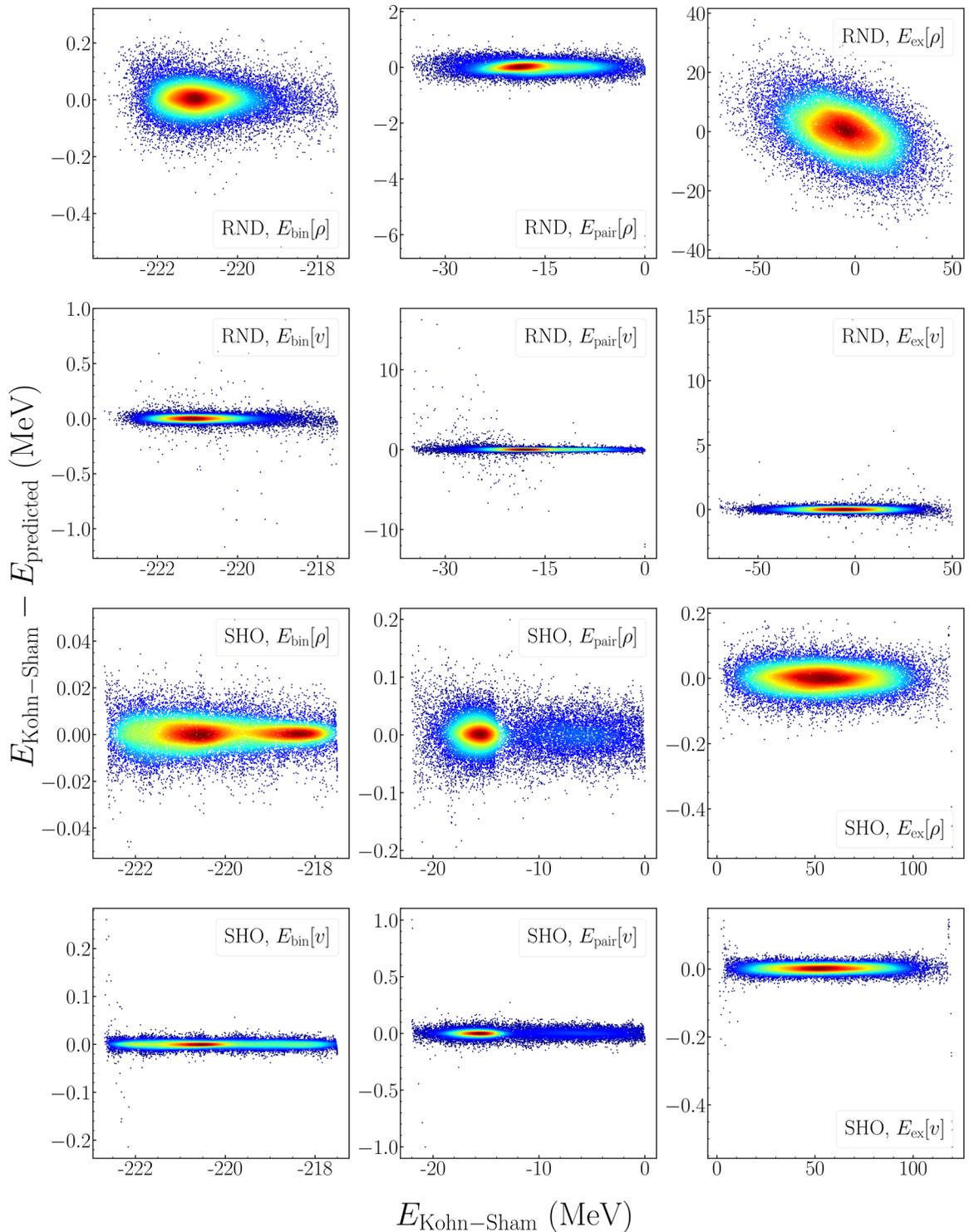


FIG. 3. The differences between the Kohn-Sham result and the predicted results from the neural network for $E[\rho]$ and $E[v]$ for the RND (the first and the second rows) and the SHO (the third and fourth rows) external fields. These are given as a function of the corresponding Kohn-Sham energies, all given in units of MeV. From the left to the right columns, the training results are shown for the binding energy, the pairing energy, and the energy of the external fields. The results for 20 000 test data points are plotted in each figure, in which densely populated (under populated) points are displayed in red (blue).

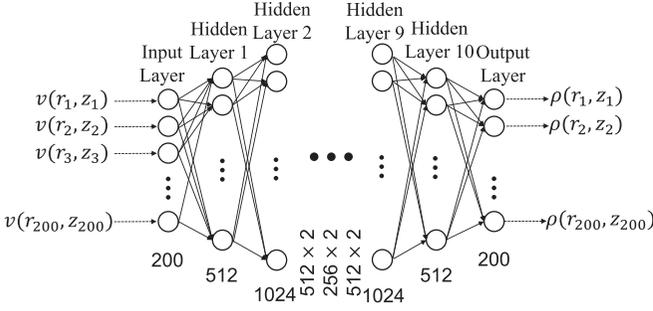


FIG. 4. A neural network with the encoder-decoder structure employed in this work for a mapping from an external v to a particle number density ρ . It consists of 10 hidden layers, all of which are fully connected. Their activation functions are ReLU, and the softmax activation function is employed for the output layer.

calculations and the result of the deep learning. We find that the points with large errors are due to external fields that have small amplitudes; that is, almost flat potentials. Since many SHO potentials used in the dataset have a large curvature, it is difficult to learn information about external fields with a small curvature. Such a problem is less likely to occur in fermionic systems when density distributions are used as the explanatory variables, leading to a somewhat better accuracy.

D. $v \rightarrow \rho[v]$

Observables are in general calculated in DFT with a particle number density, which is obtained with a given functional. That is, a functional has to be known in advance in obtaining a particle number density. As demonstrated in Ref. [6], if a neural network can directly predict the density for a given external field, the calculation speed will be significantly improved. We therefore carry out deep learning for the nuclear system with the external fields as the explanatory variables and density distributions as the objective variables. To this end, we have to take into account the fact that the densities are normalized to the particle number; that is, $\int d^3r \rho = A$. The

softmax function, which is commonly used in classification problems, enables one to require the normalization condition. We shall employ this approach in this study for the output layer. For the axial symmetric system, the following relationship exists with a discretized spatial mesh:

$$\frac{2\pi}{A} \iint r dr dz \rho(r, z) \simeq \sum_{i,j} \rho(r_i, z_j) \frac{2\pi r_i \Delta r \Delta z}{A} = 1, \quad (15)$$

where $\Delta r = \Delta z = 0.8$ fm are the mesh width.

Thus, by selecting $2\pi r_i \Delta r \Delta z \rho(r_i, z_j)/A$ as the objective variable, the normalization is automatically imposed. In this study, we use a neural network with an encoder-decoder structure for training, as is shown in Fig. 4. The MAE for $\rho[v]$ [42] is defined as

$$\text{MAE} = 2\pi \iint r dr dz |\rho_{\text{pred}}(r, z) - \rho_{\text{ans}}(r, z)|, \quad (16)$$

where $\rho_{\text{pred}}(r, z)$ and $\rho_{\text{ans}}(r, z)$ denote a predicted density and a Kohn-Sham result, respectively. Here, the bar symbol represents the average over the test data. We apply the same cutoff energies to the training data as those for the binding energy (see Table I).

Figure 5 shows the error for each test data point plotted as a function of the corresponding binding energy from the Kohn-Sham calculation. Their average corresponds to the MAE (16), which is 0.1107 for the SHO external fields and 0.4101 for the RND external fields. Figure 6 presents the images of the predicted densities for a few randomly selected data points for the RND external fields, in comparison with the corresponding Kohn-Sham densities. These examples clearly show that our neural networks successfully reproduce the Kohn-Sham densities.

We also evaluate the error obtained when the OF-EDF in Sec. III B is supplemented with the density obtained from the mapping $\rho[v]$ discussed in this subsection. For the test data of SHO and RND, the MAE of $E[\rho[v]]$ is found to be 0.0847 and 0.3059 MeV, respectively. These values are larger compared with those in Table II. This large error is referred to as the

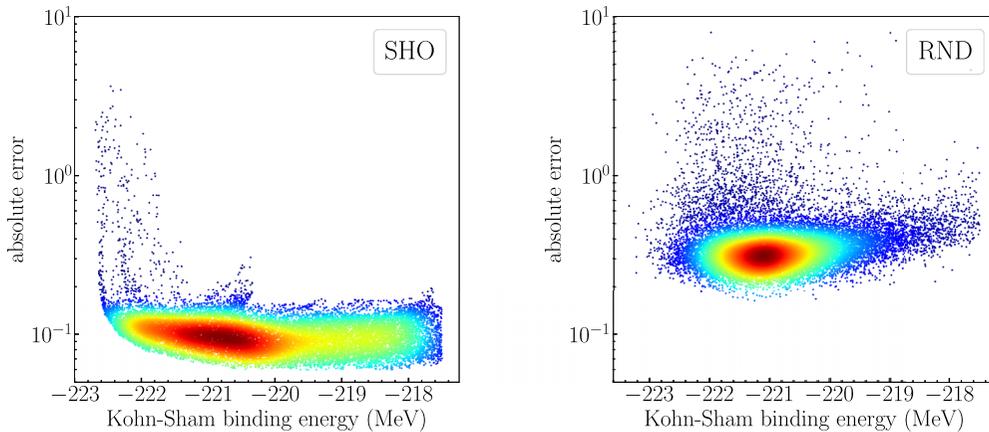


FIG. 5. The absolute error of the density distribution directly generated by a deep learning from a given external field v . It is plotted as a function of the binding energy from the corresponding Kohn-Sham calculation. The left and the right panels show the results with the SHO and the RND external fields, respectively. The densely populated points are displayed in red, while the underpopulated points are shown in blue.

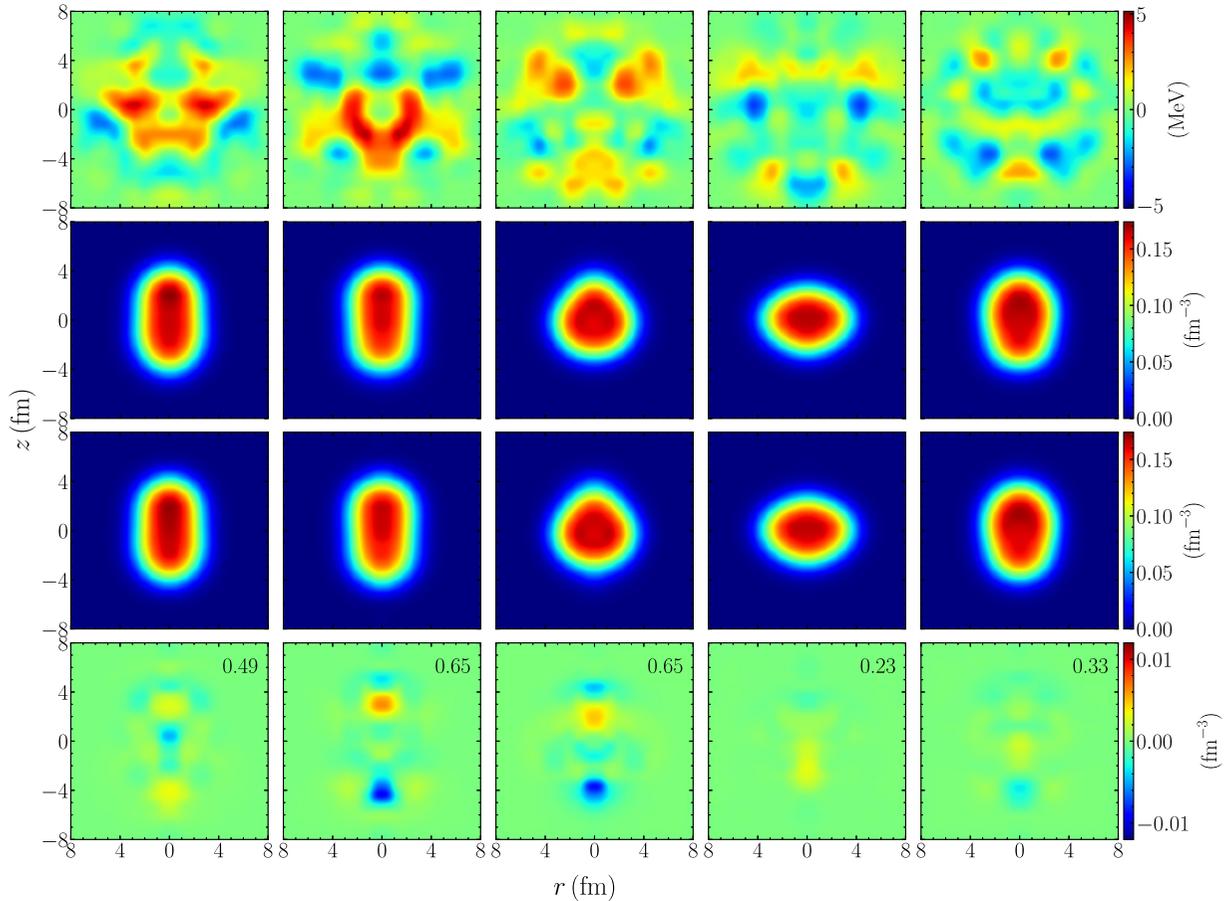


FIG. 6. Examples of the predicted densities (the panels in the third row) generated directly from the RND external potentials shown in the panels in the first row. For a comparison, the corresponding Kohn-Sham densities are also plotted in the second row. The panels in the fourth row show the difference between the predicted densities and the Kohn-Sham densities, on which the numbers denote the absolute error for each sample. The units of the color coordinate are MeV for the external potentials and fm^{-3} for the densities.

density-drive error (DDE) [6,43] and is considered to be due to the lack of information in the neural network $\rho[v]$ about the impact of the generated error on the energy during the learning process. Because of this defect, it would be more appropriate to use $E[v]$ directly when calculating the energy from a given external field.

E. Generalization performance

We have so far introduced the two types of external fields and constructed the two independent datasets. For each dataset, we have successfully provided predictions for the training data with sufficient accuracy; however, this does not guarantee performance for unknown data. For instance, a neural network trained with the RND data does not necessarily yield accurate predictions for the SHO data. This is because the RND and the SHO external fields yield density profiles in a different way to each other. In general, such generalization performance is a critical concern in applying a trained neural network to another dataset.

To investigate this issue in the context of nuclear physics, let us consider $E_{\text{SHO}}[\rho_{\text{RND}}]$ and $E_{\text{RND}}[\rho_{\text{SHO}}]$, where ρ_{SHO} and ρ_{RND} are the Kohn-Sham densities obtained with the

SHO and the RND external fields, respectively, and E_{SHO} and E_{RND} are the functionals trained with ρ_{SHO} and ρ_{RND} , respectively. In Sec. III B, we have investigated $E_{\text{SHO}}[\rho_{\text{SHO}}]$ and $E_{\text{RND}}[\rho_{\text{RND}}]$, but here we are interested in the performance of the functionals when the densities obtained with the other types of external fields are used as inputs. The left panel in Fig. 7 compares the binding energies obtained with the Kohn-Sham calculations with the RND external fields with $E_{\text{SHO}}[\rho_{\text{RND}}]$. The right panel shows similar quantities, but by inverting RND and SHO, that is a comparison between the Kohn-Sham calculations with the SHO external potentials and $E_{\text{RND}}[\rho_{\text{SHO}}]$. One can see that the performance of the neural network trained with the SHO external fields E_{SHO} is quite poor in reproducing the RND data with large randomness. On the other hand, the neural network trained with the RND external fields E_{RND} successfully predicts the SHO data, although the errors are larger than those for $E_{\text{RND}}[\rho_{\text{RND}}]$ shown in Fig. 3. The MAEs between Kohn-Sham results and predictions are 1.1523 MeV for $E_{\text{SHO}}[\rho_{\text{RND}}]$ and 0.122 MeV for $E_{\text{RND}}[\rho_{\text{SHO}}]$. A similar conclusion has been obtained also in Ref. [6]. Therefore, we can conclude that the RND potentials which we adopted are random enough for deep learning.

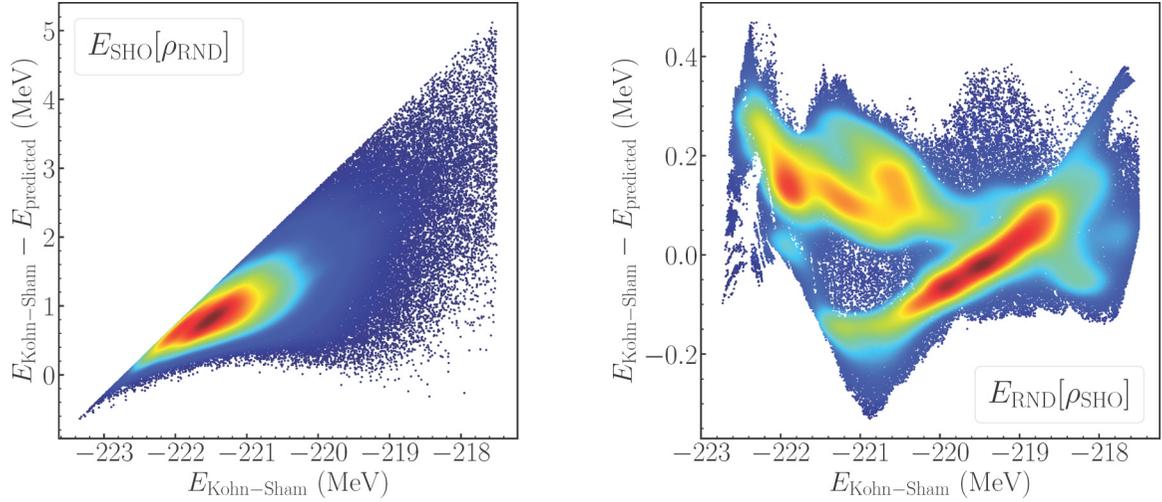


FIG. 7. A verification of generalization performance for the present deep learning. The left and right panels show the results with the RND and the SHO external fields, respectively. The horizontal axes denote the energies obtained with the Kohn-Sham calculations. On the other hand, the vertical axes denote the difference between the Kohn-Sham energy of RND $E_{\text{SHO}}[\rho_{\text{RND}}]$ and $E_{\text{RND}}[\rho_{\text{SHO}}]$, that is, the predictions of deep learning trained with the SHO (the left panel) and the RND (the right panel) external fields. Both the training and test data (200 000 data in total) are plotted in each panels because the RND (SHO) dataset are not used in training $E_{\text{SHO}}[\rho]$ ($E_{\text{RND}}[\rho]$).

IV. SUMMARY AND FUTURE PERSPECTIVES

Starting from a Skyrme functional, as a surrogate model, we have successfully constructed an energy density functional (EDF) which depends only on a particle number density. This functional does not require Kohn-Sham orbitals and thus can be regarded as an orbital-free EDF (OF-EDF). To this end, we have applied deep learning, in which the density distributions obtained with two types of random external fields (SHO and RND) were mapped on the energy with a neural network. For RND, we employed the external fields which are similar to those used in Refs. [6,7], even though one could carry out a similar study with Gaussian processes. The resultant EDF was found to predict various energies for the original Skyrme EDF with reasonable accuracy, except for the energy of the RND external fields, whose accuracy could however

be considerably improved when the energies were predicted with deep learning in which the external fields themselves were directly learned. The latter feature is more pronounced in systems with an attractive interaction than in electron systems. We have also found that deep learning with less random SHO external potentials has smaller errors as compared with that with the RND external fields.

In this paper, we have employed simple supervised learning. However, there are various methods of machine learning besides this. For example, generative models such as a generative adversarial network (GAN) [44,45] and a diffusion model [3,46] may provide efficient ways to generate the particle number densities, that is the input for deep learning used in this work to construct an OF-EDF. These methods may be useful alternatives for future application of the deep learning method discussed in this paper.

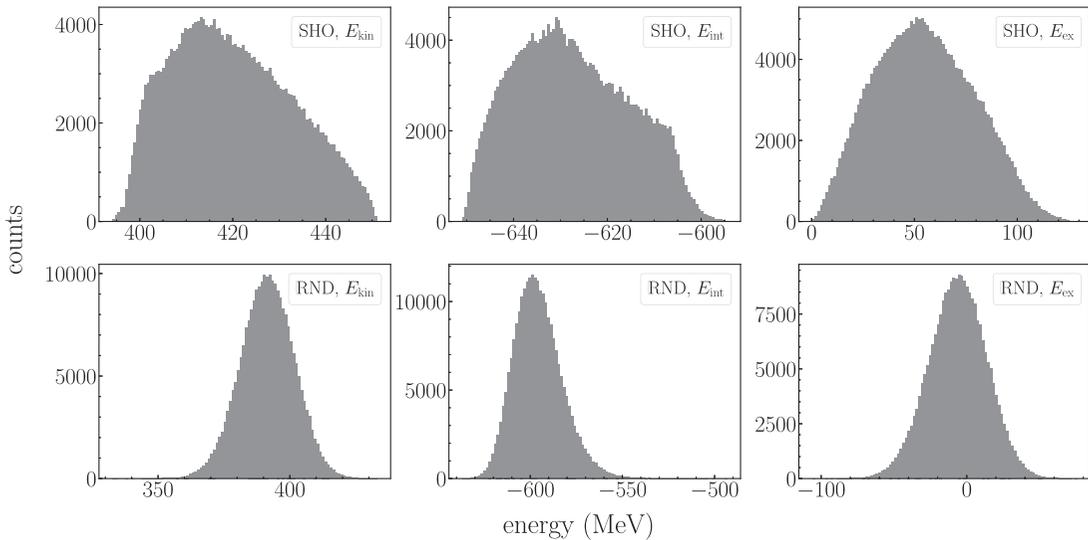


FIG. 8. Same as Fig. 2, but for the kinetic energy, the interaction energy, and the energy for the external field.

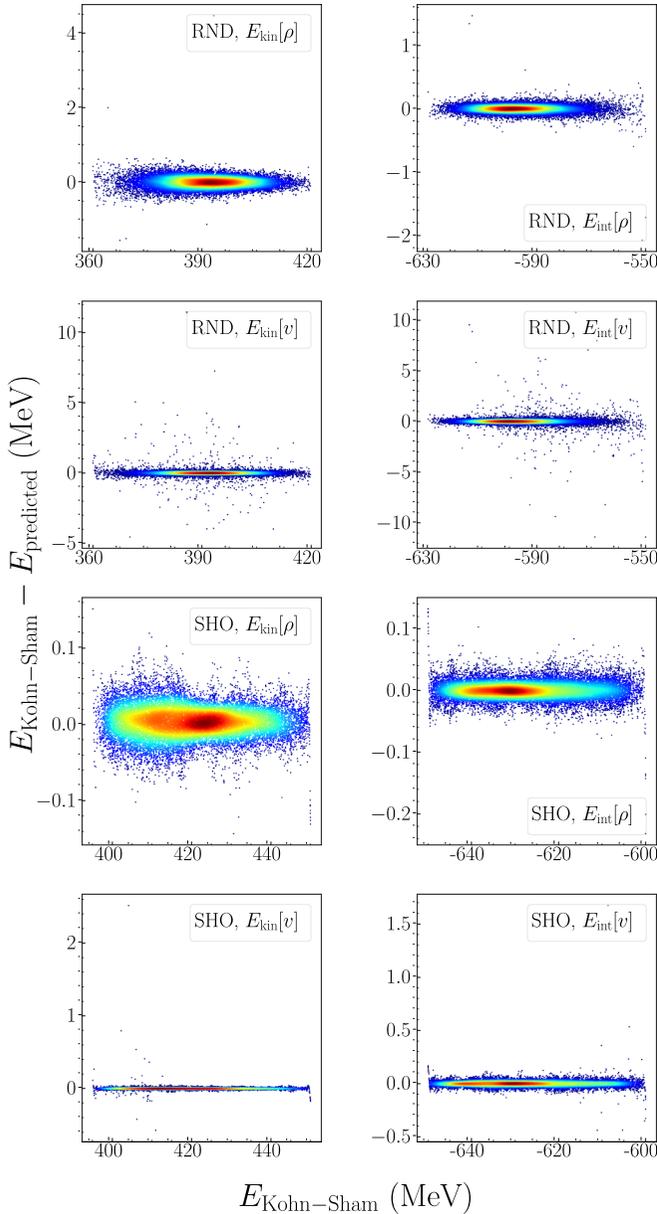


FIG. 9. Same as Fig. 3, but for the kinetic and the interaction energies.

In nuclear physics, a triaxial deformation often plays an important role, particularly in nuclear fission. In that occasion, one needs to deal with three-dimensional (3D) densities, accounting also for spin and isospin indices. We mention that traditional neural networks, comprising fully connected layers, tend not to perform efficiently with such three-dimensional data, primarily because the data size tends to become huge when the data are converted to one-dimensional data. On the other hand, CNNs have shown adaptability to data of general dimensions. With the Keras API [38], 3D CNNs can be conveniently implemented, making a straightforward extension of the present work to 3D cases. Furthermore, the vision transformer (ViT) [4], which has recently demonstrated success in image recognition tasks, can also be extended to three-dimensional data. With those

TABLE III. The mean absolute errors (MAEs) for each learning with the SHO and the RND external fields using the CNNs. The units are MeV for $E[\rho]$ and $E[v]$, while the MAE for $\rho[v]$ is dimensionless.

Type	SHO		RND	
	$E[\rho]$	$E[v]$	$E[\rho]$	$E[v]$
E_{bin}	0.0049	0.0055	0.0336	0.0245
E_{kin}	0.0151	0.0079	0.1010	0.1134
E_{int}	0.0119	0.0191	0.0484	0.1619
E_{pair}	0.0179	0.0250	0.1505	0.1852
E_{ex}	0.0220	0.0108	4.7059	0.0889
	$\rho[v]$		$\rho[v]$	
	0.1092		0.3484	

schemes, the dimensionality of the density itself is not a crucial issue in learning EDFs, without incurring additional costs for preparing training data.

One of the big advantages of using deep learning methods is that energies can be rapidly computed once test data are prepared and trained. With such low-cost calculations, numerical experiments will become much easier than before. We mention that, as objectives of research become more and more sophisticated, the number of DFT calculations required to publish a single research paper has in general been increased in these days. A typical example is a calculation for fission barriers in a multidimensional space. Even though computer performance continues to be improved, a computational cost of research has in general been increased, and it has been more complicated than before to test an idea with a numerical experiment. Fast computational methods like the one developed in this work, particularly when they are provided in a convenient format such as a Python library, can significantly shorten the time required to test and validate ideas. If numerical accuracy is an issue, one may revalidate ideas obtained with deep learning by using the traditional Kohn-Sham scheme. This could be interpreted as an application of the idea of materials informatics (MI) [47] to a theoretical research.

A potential problem in performing supervised learning is that one has to collect a large set of training data. In this work, we have chosen a relatively light nucleus ^{24}Mg and imposed axial symmetry, and thus we have treated a relatively low-cost system. However, heavy and superheavy nuclei, such as uranium isotopes, will be very costly in terms of data collection, especially when no symmetry is imposed, even though those nuclei have attracted lots of attention in nuclear physics, such as, e.g., a finding the optimal pathway in fission has still been a big theoretical challenge. In this regard, we would like to point out that a data collection needs not be performed individually; it could actually be done collaboratively by many researchers. A lot of good quality data, which are ready to be used in deep learning, may have already existed for some selected nuclei. Therefore, we believe that it is desirable to establish a framework in the nuclear theory community to collect numerical data and/or to carry out numerical calculations with unified hyperparameters such as a mesh size. Such a collaborative

TABLE IV. A CNN employed in this work to learn $E[\rho]$ and $E[v]$. The type of each layer is expressed with the language of Keras API [38], with which one can reproduce this neural network easily. In each layer, all arguments not mentioned are default values.

Layer	Type
Input	Input(shape = (10, 20, 1))
1	Conv2D(filters = 32, kernel_size = 3, activation = "relu")
2	Conv2D(filters = 64, kernel_size = 3, activation = "relu")
3–8	Conv2D(filters = 64, kernel_size = 4, padding = "same," activation = "relu")
9	Conv2D(filters = 128, kernel_size = 3, activation = "relu")
10–13	Conv2D(filters = 128, kernel_size = 3, padding = "same," activation = "relu")
14	Flatten()
15	Dense(units = 128, activation = "relu")
Output	Dense(units = 1, activation = "sigmoid")

approach will help advance research more efficiently and effectively, benefiting the whole nuclear physics community.

ACKNOWLEDGMENTS

We thank G. Colò for useful discussions. This work was supported by JSPS KAKENHI (Grants No. 21J22348, No. JP19K03824, No. JP19K03861, No. JP19K03872, and No. JP23K03414).

APPENDIX A: FIGURES FOR E_{kin} AND E_{int}

In this Appendix, we show Figs. 8 and 9 for E_{kin} and E_{int} which are not shown in Sec. III. We also show a figure for a histogram for E_{ex} . The conclusions remain the same as those for Figs. 2 and 3.

APPENDIX B: CONVOLUTIONAL NEURAL NETWORK

In a previous study [6], the convolutional neural network (CNN) was used for calculating an OF-EDF. An efficient structure in the CNN enables computers to recognize images,

which reduces the trainable parameters, and the CNN works well in learning with large-size images. Since the input image size is small enough in this study, we can use the neural network only consisting of the fully connected layer. We can in principle perform the training with CNNs as well, but we find that the results are not significantly different (see Table III). To this end, we use the CNN listed in Tables IV and V. The learning methods are the same as for the fully connected layer. Considering that the CNN is computationally more expensive than the fully connected layer, the benefit of using the CNN does not seem to be substantial, at least for the system studied in this paper.

Of course, there are many choices and hyperparameters in deep learning. Therefore, we do not mean to claim that there is no sufficient benefit from using CNNs. However, if one needs better and more accurate training results, skill and experience are required. In that occasion, one way to proceed is to ask professionals to submit their ideas in a competition on sites such as Kaggle [48], for example. Hopefully, such approach could help identify optimal machine-learning methods and hyperparameters. For example, the IceCube held a competition on Kaggle [49].

TABLE V. A CNN employed in this work to learn $\rho[v]$. The neural network has encoder-decoder structure. Layers 1–4 are reducing convolutional layers, and layers 5–7 are nonreducing convolutional layers, where the size of images are (2, 12). We use layers 8–11 as deconvolution layers to enlarge them up to (10, 20). In each layer, all arguments not mentioned are default values.

Layer	Type
Input	Input(shape = (10, 20, 1))
1	Conv2D(filters = 32, kernel_size = 3, activation = "relu")
2	Conv2D(filters = 64, kernel_size = 3, activation = "relu")
3	Conv2D(filters = 128, kernel_size = 3, activation = "relu")
4	Conv2D(filters = 256, kernel_size = 3, activation = "relu")
5–7	Conv2D(filters = 256, kernel_size = 4, padding = "same," activation = "relu")
8	Conv2DTranspose(filters = 256, kernel_size = 3, activation = "relu")
9	Conv2DTranspose(filters = 128, kernel_size = 3, activation = "relu")
10	Conv2DTranspose(filters = 64, kernel_size = 3, activation = "relu")
11	Conv2DTranspose(filters = 32, kernel_size = 3, activation = "relu")
12	Flatten()
13	Dense(units = 200, activation = "softmax")
output	Reshape[target_shape = (10, 20, 1)]

- [1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, edited by J. Burstein, C. Doran, and T. Solorio (Association for Computational Linguistics, 2019), pp. 4171–4186.
- [2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse *et al.*, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 1877–1901.
- [3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022).
- [4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) [cs.CV] (2021).
- [5] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, [arXiv:2005.08100](https://arxiv.org/abs/2005.08100).
- [6] K. Ryczko, D. A. Strubbe, and I. Tamblyn, *Phys. Rev. A* **100**, 022512 (2019).
- [7] K. Mills, M. Spanner, and I. Tamblyn, *Phys. Rev. A* **96**, 042113 (2017).
- [8] K. Shiina, H. Mori, Y. Okabe, and L. Hwee-Kuan, *Sci. Rep.* **10**, 2177 (2020).
- [9] V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi, *npj Comput. Mater.* **4**, 29 (2018).
- [10] R.-D. Lasserri, D. Regnier, J.-P. Ebran, and A. Penon, *Phys. Rev. Lett.* **124**, 162502 (2020).
- [11] G. Saxena, A. Jain, and P. Sharma, *Phys. Scr.* **96**, 125304 (2021).
- [12] Z.-A. Wang and J. Pei, *Phys. Rev. C* **104**, 064608 (2021).
- [13] X. Wu, L. Guo, and P. Zhao, *Phys. Lett. B* **819**, 136387 (2021).
- [14] M. R. Mumpower, T. M. Sprouse, A. E. Lovell, and A. T. Mohan, *Phys. Rev. C* **106**, L021301 (2022).
- [15] A. E. Lovell, A. T. Mohan, T. M. Sprouse, and M. R. Mumpower, *Phys. Rev. C* **106**, 014305 (2022).
- [16] A. Sarkar and D. Lee, *Phys. Rev. Res.* **4**, 023214 (2022).
- [17] O. M. Molchanov, K. D. Launey, A. Mercenne, G. H. Sargsyan, T. Dytrych, and J. P. Draayer, *Phys. Rev. C* **105**, 034306 (2022).
- [18] X. H. Wu, Z. X. Ren, and P. W. Zhao, *Phys. Rev. C* **105**, L031303 (2022).
- [19] M. Verriere, N. Schunck, I. Kim, P. Marević, K. Quinlan, M. N. Ngo, D. Regnier, and R. D. Lasserri, *Front. Phys.* **10**, 1028370 (2022).
- [20] M. Knöll, T. Wolfgruber, M. L. Agel, C. Wenz, and R. Roth, *Phys. Lett. B* **839**, 137781 (2023).
- [21] X. Zhang, W. Lin, J. M. Yao, C. F. Jiao, A. M. Romero, T. R. Rodríguez, and H. Hergert, *Phys. Rev. C* **107**, 024304 (2023).
- [22] G. P. A. Nobre, D. A. Brown, S. J. Hollick, S. Scoville, and P. Rodríguez, *Phys. Rev. C* **107**, 034612 (2023).
- [23] Y. L. Yang and P. W. Zhao, *Phys. Rev. C* **107**, 034320 (2023).
- [24] Z.-X. Yang, X.-H. Fan, Z.-P. Li, and H. Liang, *Phys. Lett. B* **840**, 137870 (2023).
- [25] K.-F. Pu, H.-L. Li, H.-L. Lü, and L.-G. Pang, *Chin. Phys. C* **47**, 054104 (2023).
- [26] A. Boehnlein, M. Diefenthaler, N. Sato, M. Schram, V. Ziegler, C. Fanelli, M. Hjorth-Jensen, T. Horn, M. P. Kuchera, D. Lee, W. Nazarewicz, P. Ostroumov, K. Orginos, A. Poon, X.-N. Wang, A. Scheinker, M. S. Smith, and L.-G. Pang, *Rev. Mod. Phys.* **94**, 031003 (2022).
- [27] M. Bender, P.-H. Heenen, and P.-G. Reinhard, *Rev. Mod. Phys.* **75**, 121 (2003).
- [28] T. Naito, S. Endo, K. Hagino, and Y. Tanimura, *J. Phys. B: At., Mol. Opt. Phys.* **54**, 165201 (2021).
- [29] P. Ring and P. Schuck, *The Nuclear Many-Body Problem* (Springer-Verlag, New York, 1980).
- [30] E. Chabnat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer, *Nucl. Phys. A* **635**, 231 (1998).
- [31] P.-G. Reinhard, D. J. Dean, W. Nazarewicz, J. Dobaczewski, J. A. Maruhn, and M. R. Strayer, *Phys. Rev. C* **60**, 014316 (1999).
- [32] A. Bohr and B. R. Mottelson, *Nuclear Structure* (Benjamin, New York, 1969), Vol. 1.
- [33] D. Vautherin, *Phys. Rev. C* **7**, 296 (1973).
- [34] H. Kasuya and K. Yoshida, *Prog. Theor. Exp. Phys.* **2021**, 013D01 (2021).
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Curran Associates, Inc., 2012), Vol. 25.
- [36] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, *J. Big Data* **8**, 53 (2021).
- [37] D. P. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG] (2017).
- [38] Keras official document, <https://keras.io>.
- [39] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, [arXiv:1908.03265](https://arxiv.org/abs/1908.03265) [cs.LG] (2021).
- [40] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).
- [41] M. R. Mumpower, R. Surman, G. C. McLaughlin, and A. Aprahamian, *Prog. Part. Nucl. Phys.* **86**, 86 (2016).
- [42] N. Dubray and D. Regnier, *Comput. Phys. Commun.* **183**, 2035 (2012).
- [43] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, *Nat. Commun.* **8**, 872 (2017).
- [44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Curran Associates, Inc., 2014), Vol. 27.
- [45] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, *IEEE Trans. Knowl. Data Eng.* **35**, 3313 (2023).
- [46] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, edited by F. Bach and D. Blei (PMLR, 2015), Vol. 37, pp. 2256–2265.
- [47] R. Ramprasad, R. Batra, G. P. Lian, A. Mannodi-Kanakithodi, and C. Kim, *npj Comput. Mater.* **3**, 54 (2017).
- [48] Kaggle, <https://www.kaggle.com>.
- [49] Winners of IceCube machine learning competition announced, <https://icecube.wisc.edu/news/outreach/2023/06/winners-of-icecube-machine-learning-competition-announced/>.