scientific reports

OPEN

Optimization of physical quantities in the autoencoder latent space

S. M. Park[1], H. G. Yoon[1], D. B. Lee[1], J. W. Choi[2], H. Y. Kwon[2✉] & C. Won[1✉]

We propose a strategy for optimizing physical quantities based on exploring in the latent space of a variational autoencoder (VAE). We train a VAE model using various spin configurations formed on a two-dimensional chiral magnetic system. Three optimization algorithms are used to explore the latent space of the trained VAE. The first algorithm, the single-code modification algorithm, is designed for improving the local energetic stability of spin configurations to generate physically plausible spin states. The other two algorithms, the genetic algorithm and the stochastic algorithm, aim to optimize the global physical quantities, such as topological index, magnetization, energy, and directional correlation. The advantage of our method is that various optimization algorithms can be applied in the latent space containing the abstracted representation constructed by the trained VAE model. Our method based on latent space exploration is utilized for efficient physical quantity optimization.

The optimization process is the search for the optimal solution among a large set of possible solutions under given constraints defining a system. It is a challenging problem in various research fields including science, and various numerical methods have been extensively utilized to solve optimization problems. In material science, optimal molecular structures have been successfully investigated based on density functional theory[1–3]. Simulated annealing (SA) is a conventional method to obtain an approximate global optimum[4–6], and a genetic algorithm has also been used for designing molecules with desired properties[7,8]. Recently, the machine learning technique has also exhibited great potential in solving optimization problems. For example, unsupervised machine learning algorithms were used to find the ground state of a two-dimensional magnetic system[9,10] and a generalized neural-network method was presented for constructing potential-energy surfaces based on density functional theory[11].

The optimization methods can be combined with deep generative models, e.g., generative adversarial networks (GAN)[12] or variational autoencoders (VAE)[13], for various purposes. A deep generative model denotes deep neural networks trained to approximate complicated high-dimensional probability distributions using given samples[14]. Once a deep generative model is successfully trained, it can convert randomly sampled instances into real space data using the deep generative model. In previous studies[15,16], evolutionary algorithms were adopted to explore the latent space of a trained generative model, where the latent space denotes a virtual space from which the random instances are located at to be converted into meaningful data. Since the generative model is constructed by differentiable neural networks, we can use gradient-based optimization methods[17]. Gomez-Bombarelli et al.[18] applied a gradient-based method in the latent space of trained VAE to optimize molecules via their properties. Similarly, if we combine various optimization methods with deep generative models trained on physical data, several advantages are expected in solving various optimization problems in physics. Specifically, a well-trained deep generative model obeying the constraints of the system will enable to search for the solutions of optimization problems under those constraints.

In this study, we train a VAE on our dataset composed of various spin configurations formed on a two-dimensional magnetic system. We explore the VAE latent space to obtain optimal physical states. To search for the optimal solution by exploring the latent space of the trained VAE network, three optimization algorithms are implemented: the single-code modification algorithm, genetic algorithm, and stochastic algorithm. Using these algorithms, we obtain either the local or global optimal states that are representative of the system but not included in the training dataset and propose that our methods have the potential to be applied to various optimization problems in scientific research fields.

Strategy

Dataset. The dataset should be chosen to contain rich features within a large variety of possible spin states, so that our VAE network can properly learn to abstract input spin configurations to latent codes and decode the abstracted information to spin configurations. In this study, the labyrinth spin configurations[19] which have various maze patterns constructed by magnetic spiral structures are selected as our dataset.

[1]Department of Physics, Kyung Hee University, Seoul 02447, South Korea. [2]Center for Spintronics, Korea Institute of Science and Technology, Seoul 02792, South Korea. ✉email: soky572@gmail.com; cywon@khu.ac.kr
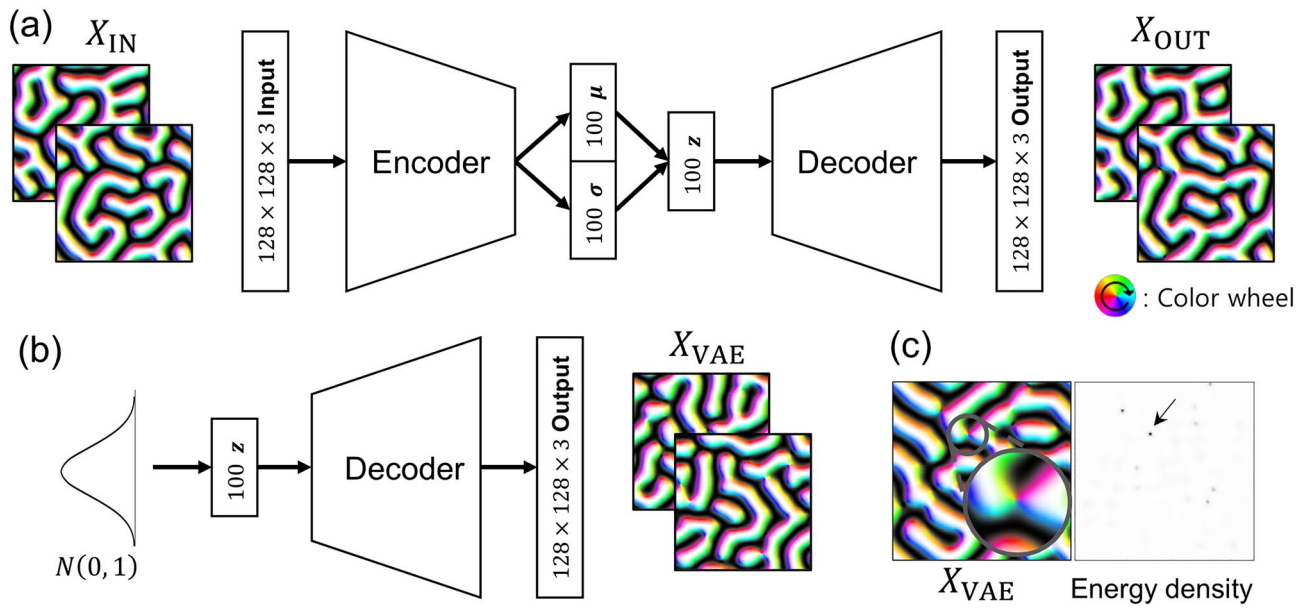
Figure 1. (**a**) Schematic diagram of the variational autoencoder and examples of X_{IN} and X_{OUT}. The color wheel and black/white contrast of the spin configuration images indicate the in-plane and out-of-plane magnetization directions, respectively. (**b**) A schematic illustration of the generation process of X_{VAE}. (**c**) A representative X_{VAE} image and its local energy density. The inset in the spin configuration image of (**c**) shows a magnified view of a nodal point. The arrow in the energy density image indicates the location of high energy density and it corresponds the position of the nodal point.

To generate the labyrinth spin configurations, we implement the two-dimensional magnetic system using a square lattice model composed of 128×128 grid sites and consider a simple Hamiltonian model, $\mathcal{H} = -J\sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j - \sum_{\langle i,j \rangle} \vec{D}_{ij} \cdot \left(\vec{S}_i \times \vec{S}_j \right)$, where J is the exchange interaction parameter, $\vec{D}_{ij}$ is the Dzyaloshinskii-Moriya (DM) interaction vector[20,21], and $\vec{S}_i$ is a Heisenberg spin on the i-th grid site. The exchange and DM interaction parameters are fixed to $J = 1$ and $\left| \vec{D}_{ij} \right| = 0.3$. The system was carefully studied, so that we can investigate and interpret the results from our algorithm by relying on those of previous studies[20–24]. The dataset is generated by an SA process, which has been used to simulate a temperature annealing process of magnetic systems in several previous studies[5,19,25]. Since the detailed magnetic textures are determined by the spontaneous symmetry breaking process, they can have countless different metastable states in a fixed condition. Using this process, we generate a training dataset consisting of 30,000 labyrinth spin configurations, which is called X_{IN} in this study, and some samples are shown in Fig. 1a.

Since our purpose is to find optimization solutions, it is important that the training data are sampled from entire data distribution. The SA process is performed by heat bath algorithm in Monte-Carlo simulations[26], a carefully studied method. This method can generate a variety of samples due to the contribution of random numbers of it.

Training VAE and generating new data. To extract the appropriate features implied in the labyrinth spin configurations, we construct the VAE network as shown in Fig. 1a. Specifically, the encoder of the VAE compresses X_{IN} into two groups composed of N numbers each ($N = 100$ in this study). These are used as the mean ($\boldsymbol{\mu}$) and standard deviation ($\boldsymbol{\sigma}$) values to construct N normal distributions, and one number is sampled from each of the normal distributions (N numbers are sampled in total). In general, the set of sampled numbers, $\boldsymbol{z}$, is called the latent code; $\boldsymbol{z}$ is considered an N-dimensional vector represented in the latent space. Through the decoder, $\boldsymbol{z}$ is decoded into an output spin configuration, X_{OUT}.

We train the VAE network using our spin configuration dataset. In other words, the network parameters are modified to minimize a loss function, L_{VAE}, shown in Eq. (1), where L_{RC} is the reconstruction loss term, L_{KL} is the Kullback–Leibler (KL) loss term, and β is the coefficient of L_{KL}[27]. The two loss terms, L_{RC} and L_{KL}, are expressed in Eq. (2), where $S_{IN,i,\alpha}$ and $S_{OUT,i,\alpha}$ indicate the α component of spins at the i-th grid site of X_{IN} and X_{OUT}, respectively. μ_n and σ_n are the n-th components of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively.

$$L_{VAE} = L_{RC} + \beta L_{KL}, \tag{1}$$

$$L_{RC} = \left(S_{IN,i,\alpha} - S_{OUT,i,\alpha} \right)^2_{i,\alpha(=x,y,z)}$$

$$L_{KL} = \frac{1}{2} \sum_{n=1}^{N} \left(\sigma_n^2 + \mu_n^2 - \ln \left(\sigma_n^2 \right) - 1 \right). \tag{2}$$

(a)
```
Procedure SingleCodeModification(Decoder, ObjectiveFunction, z)
    z* ← z                                          ▷ Initialize trainable variables
    For i In Iteration:                             ▷ Repeat for iteration
        X ← Decoder(z*)                             ▷ Decode the latent code
        q ← ObjectiveFunction(X)                    ▷ Calculate the energy of decoded spin configuration
        z* ← Adam(z*, q)                            ▷ Adam-Optimizer updates the latent code minimizing the Fitness
    Return z*
```

(b)
```
Procedure GeneticAlgorithm(Decoder, ObjectiveFunction, Population, MutationError)
    z* ← RandomSampling(Population, N(0,1))         ▷ Start with randomly initialized generation, which is the latent codes
    For i In Iteration:                             ▷ Repeat for iteration
        X ← Decoder(z*)                             ▷ Decode the generation
        Q ← ObjectiveFunction(X)                    ▷ Calculate the objective quantity of decoded spin configurations
        Probability ← Convert(Q)                    ▷ Calculate the probability the individual would be selected
        SelectedIndividuals ← Selection(z*, Probability)   ▷ Select individuals from generation
        z* ← Crossover(SelectedIndividuals)         ▷ Crossover the genes of the selected individuals
        z* ← Mutation(z*, MutationError)            ▷ Apply mutation operation to the new generation
    Return z*                                       ▷ Replace the generation with new generation
```

(c)
```
Procedure StochasticAlgorithm(Decoder, ObjectiveFunction, SampleSize)
    μ*, σ* ← {0}, {1}                               ▷ Initialize mean and standard deviation by 0s and 1s
    For i In Iteration:                             ▷ Repeat for iteration
        z* ← RandomSampling(SampleSize, N(μ*, σ*))  ▷ Sampling latent codes with the mean and standard deviation
        X ← Decoder(z*)                             ▷ Decode the latent codes
        Q ← ObjectiveFunction(X)                    ▷ Calculate the objective function of decoded spin configurations
        L_KL ← KLLossCalculation(z*)                ▷ Calculate Kullback-Leibler loss
        L ← TotalLossCalculation(Q, L_KL)           ▷ Calculate the total loss
        μ*, σ* ← AdamOptimizer([μ*, σ*], L)         ▷ Adam-Optimizer update the mean and standard deviation minimizing the total loss
    Return μ*, σ*
```
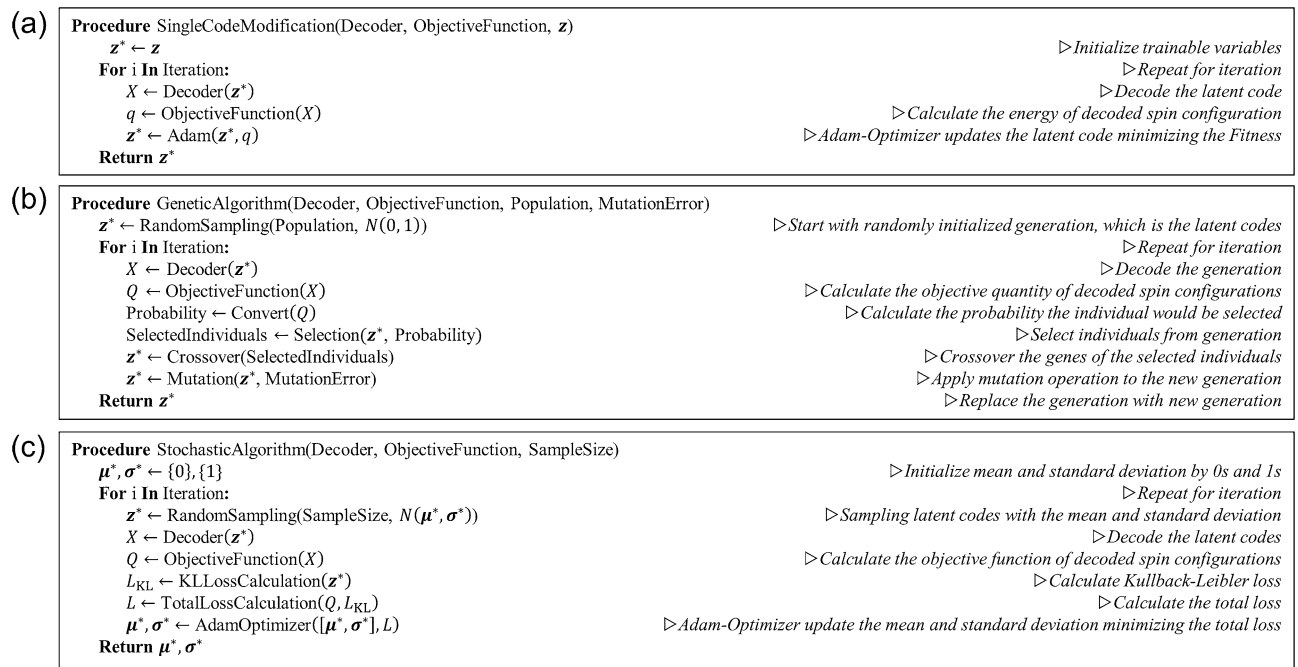
Figure 2. Pseudocode descriptions of the (**a**) single-code modification algorithm, (**b**) genetic algorithm, and (**c**) stochastic algorithm.

To minimize L_{RC}, the VAE model produces X_{OUT}, which is similar to X_{IN}. L_{KL} measures how a given probability distribution is different from a reference probability distribution. In this study, the purpose of L_{KL} is to make the probability distribution of z equal to the standard normal distribution; as L_{KL} decreases, the values of σ_n and μ_n approach 1 and 0, respectively. β is chosen to be 0.001, as described in the Methods section.

After the training process, we can generate various spin configurations by decoding the latent codes that are sampled from the standard normal distribution. The generation process is illustrated schematically in Fig. 1b, and the spin configurations generated using this process are called X_{VAE}s in this study. The X_{VAE}s have similar characteristics to the spin configurations in our training dataset. This indicates that the VAE network is well-trained to learn the magnetic characteristics implied in the training dataset, such as the spin profiles, length scale, and chirality.

There are several nodal points in the X_{VAE}s, which are visually unfeasible and not found in the training dataset. We can also quantitatively evaluate the feasibility by calculating the energy of X_{VAE} since the energy is the generating rule of the training dataset. The energy density of nodal points is higher than that of other plausible pixels, as indicated by the energy peak points in Fig. 1c. Unstable points are inevitable from the direct interpretation of the sampled z, since the latent space is continuous while the stable structures are topologically separated. This problem of directly applying VAE to a topologically discrete structure was reported in a previous study[10].

Latent space exploration algorithms. To generate physically plausible data or to obtain the optimized structure using the trained VAE decoder, we attempt three optimization algorithms exploring the latent space. The first is a simple optimization algorithm, which is called the single-code modification algorithm in this study. The algorithm pseudocode is shown in Fig. 2a. We prepare a latent code, $z*$, composed of N trainable variables and initialize it with $z* = z$, where z is randomly sampled numbers from the standard normal distribution. $z*$ is decoded into a spin configuration using the trained decoder and evaluated with its energy. The Adam optimizer[28] modifies $z*$ to minimize the spin configuration energy. This process is repeated until the energy stabilizes. The main idea of the single-code modification algorithm is to search the latent space to find a better solution near the position indicated by the selected initial latent code. Thus, the states generated by the single-code modification algorithm are locally optimized states in the latent space, and they are not uniquely determined.

If we want to find a state representing a global optimum state, the whole latent space should be searched. To utilize the idea for the general optimization problem, we apply a genetic algorithm instead of the single-code modification algorithm. The genetic algorithm is one of the conventional optimization methods imitating biological evolution. In the algorithm, there is a population composed of many individuals, and the population evolves to optimize an objective goal. Every individual consists of their genes. Individuals are evaluated with the objective function and selected to pass on their genes to the next generation with a probability calculated from the evaluation. Crossover and mutation operations are performed to combine the genes of the selected individuals and to maintain the genetic diversity, respectively. The pseudocode of the genetic algorithm used in this study is shown in Fig. 2b. We consider $z*$ as an individual and every number $\{z_i^*\}$ of $z*$ is considered its gene. Our population is composed of 100 individuals, $z*$s, which are randomly initialized. In every iteration of the genetic algorithm
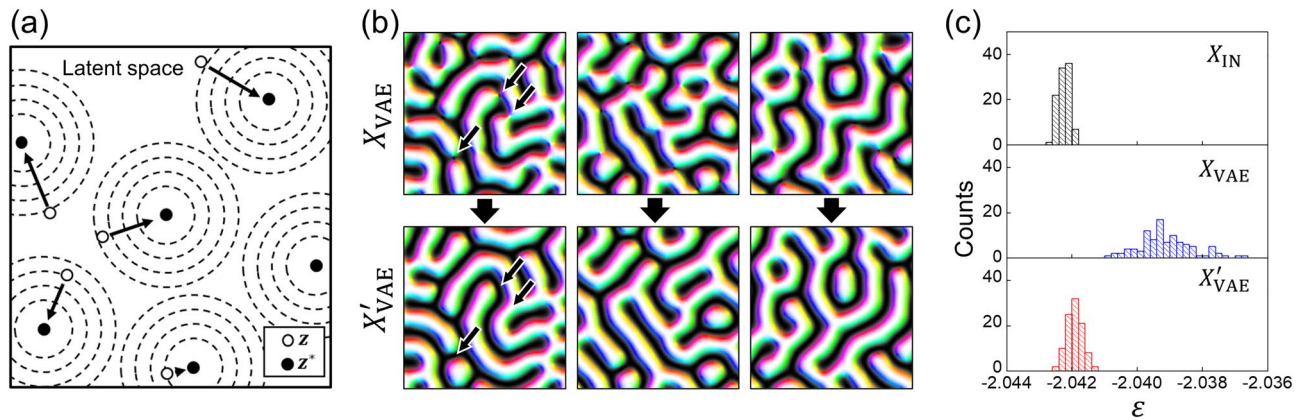
Figure 3. The results of the single-code modification algorithm. (**a**) A schematic illustration of the algorithm. The empty circles, filled circles, and dashed lines illustrate the initial latent code z, modified latent code z^*, and contour line of the energy, respectively. (**b**) Examples of the generated spin configuration X_{VAE} and the modified spin configuration X'_{VAE}. The arrows on the leftmost X_{VAE} image indicate nodal points, and the arrows on the leftmost X'_{VAE} image indicate the same regions, showing that the nodal points are removed. (**c**) The ε distributions of X_{IN}s, X_{VAE}s, and X'_{VAE}s. The distribution was gathered from 100 spin configurations of each case.

process, z^*s are decoded and evaluated with an objective function. In this study, we use several quantities that can be calculated from a spin configuration as the objective function, which is discussed in the Results section.

The third algorithm, which is the stochastic algorithm, is illustrated in Fig. 2c. The algorithm is similar to the single-code modification algorithm but is designed for global optimization. In this case, we update μ^* and σ^* instead of z^*, which denotes the mean and standard deviation values of normal distributions. μ^* and σ^* are each composed of N trainable variables, and they are initialized with 0 s and 1 s, respectively. In every training iteration, hundreds of z^* are sampled from N normal distributions constructed using μ^* and σ^* values, and the z^*s are decoded into output spin configurations using the trained decoder. They are evaluated with a given objective quantity, Q, similar to the case of the genetic algorithm. The Adam optimizer updates the μ^* and σ^* to minimize the function, $L = \langle Q \rangle + \gamma L^*_{\mathrm{KL}}$, where L^*_{KL} is a term similar to the KL loss term shown in Eq. (2) except that it is calculated using μ^* and σ^*, and γ is the coefficient of L^*_{KL}. L^*_{KL} is adopted to control the diversity of the z^*s, and it is expected that this diversity can prevent the distribution from being trapped in local extremum points in the latent space. As the iteration progresses, we gradually decrease γ so that the distribution slowly converges to the optimal solution in the latent space.

Results

Single-code modification algorithm. The purpose of this algorithm is to generate new structures that have similar characteristics to those in the training data, and Fig. 3a shows a schematic illustration of the algorithm. As described earlier, X_{VAE}s decoded from zs have many nodal points that are not included in the training data, which means that direct sampling from the standard normal distribution is not enough to generate physically plausible states. Hence, we perform the single-code modification algorithm to search the better positions near zs, which can be decoded into energetically plausible states. Figure 3b shows some examples of X_{VAE} and X'_{VAE}, where X'_{VAE} represents the modified spin configuration after applying this algorithm. As indicated by the black arrows, the spin structures around the nodal points are revised to stable structures; the nodal points are removed. Figure 3c shows the distributions of energy density, ε, of X_{IN}, X_{VAE}, and X'_{VAE}. The distribution of X'_{VAE} is similar to that of X_{IN}, but there is a clear difference for the case of X_{VAE}. This also indicates that we can generate spin configurations analogous to X_{IN} not only structurally but also physically and energetically using the single-code modification algorithm.

Of course, it is possible to incorporate the energy of X_{VAE}s in the loss function of VAE so that X_{VAE}s are forced to be energetically stable. A previous study[10] shows that this approach is available at generating feasible and energetically stable samples in a magnetic system.

Genetic algorithm. Genetic algorithms are known as efficient optimization methods, but their effectiveness is strongly dependent on how we define the individuals and genes in a given system. In this study, a latent code, which is the abstracted information of a spin configuration, is considered an individual, and the components of the latent code, which characterize the properties of the spin configuration, are considered the genes. By doing so, we perform genetic algorithm in a compressed space of $\mathbb{R}^{100}$ rather than the original space $\mathbb{R}^{128 \times 128 \times 3}$, reducing computational cost.

A schematic explanation of this algorithm is shown in Fig. 4a. We start from 100 individuals initialized with random numbers sampled from the standard normal distribution. To determine the probability that an individual will be selected, the spin configuration decoded from the individual is evaluated by a certain objective quantity. The mutated gene, an instance of an offspring, is sampled from the Gaussian distribution, so that the offspring keeps in the learned distribution by the VAE, following the general features of training spin configurations.
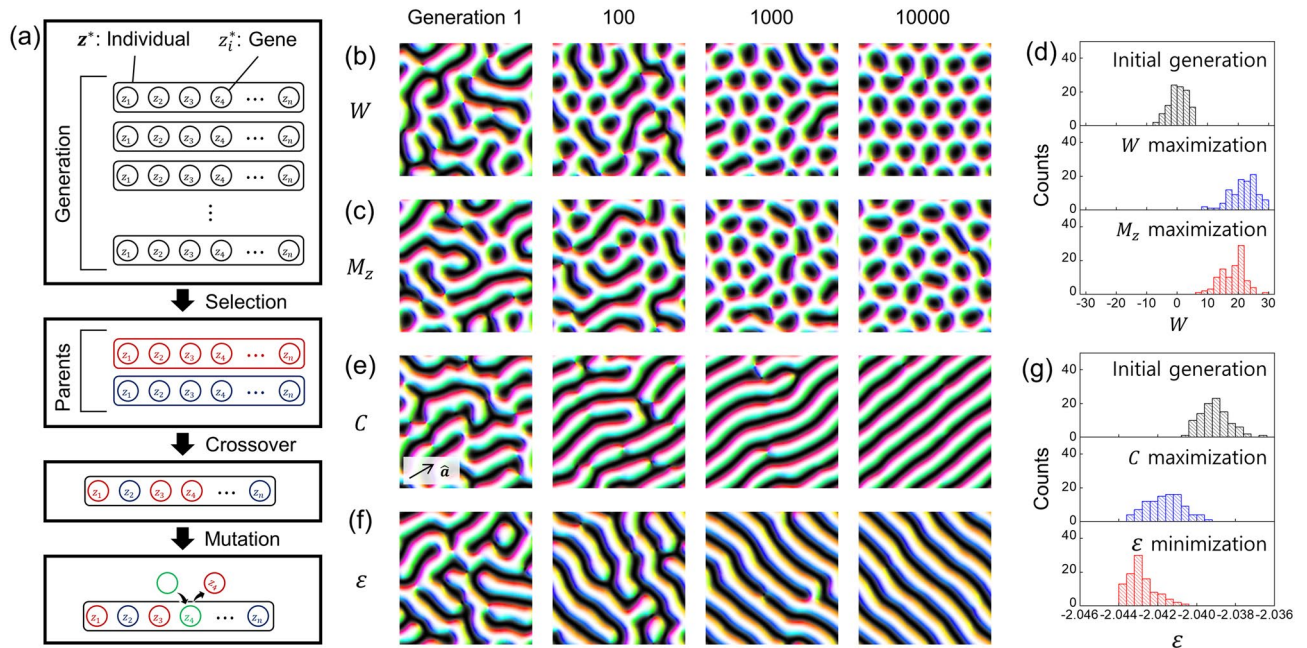
Figure 4. Results of the genetic algorithm. (**a**) A schematic illustration of the algorithm. The circles and rounded squares indicate the genes (components z_i^*) and individuals (latent code z^*), respectively. The blue/red colors indicate each selected parent, and the green circle indicates the mutated gene. (**b,c,e,f**) The best individuals in each generation optimizing (**b**)W, (**c**)M_z, (**e**)C, and (**f**)ε, respectively. The arrow in (**e**) indicates the direction of a. (**d**) W values of spin configurations evolved with W maximization and M_z maximization, (**g**) ε values of spin configurations evolved with C maximization and ε minimization.

We experiment with two structural quantities and two physical quantities for the evaluation. The structural quantities are the topological index, $W = \frac{1}{4\pi} \int M \cdot \left(\frac{\partial M}{\partial x} \times \frac{\partial M}{\partial y} \right) dx dy$, and the spin correlation along a certain direction pointed by a unit vector a, $C = \langle S(r) \cdot S(r + a) \rangle$. The physical quantities are the energy density, ε, indicated by the Hamiltonian of the system and the out-of-plane magnetization, M_z. Figure 4b,c,e,f show the best individuals of each generation evolved by the genetic algorithm to maximize W, maximize M_z, maximize C, and minimize ε, respectively.

As shown in Fig. 4b, the spin configuration is changed from the labyrinth structure to the skyrmion structure[29] by maximizing W. The spin configuration of the best individual in the last generation almost forms a regular lattice of skyrmions, and it is a noticeable result because the skyrmion lattice structure is not included in the training set of X_{IN}s; the skyrmion lattice structure is one of the possible metastable states of the system, but it is statistically impossible to generate it using an SA process due to the existence of numerous metastable states of the system. This means that the VAE model has learned the general features of the system, such as the chirality, the width of magnetic domains, and sinusoidal profiles of local spin structures, and a set of optimal features to maximize W is properly searched through the genetic algorithm in the latent space of the trained VAE.

Interestingly, a similar skyrmion structure is obtained by maximizing M_z, as shown in Fig. 4c. In numerous studies[24,30–32], it is commonly observed that the labyrinth structure becomes a magnetic skyrmion configuration when an external field is applied to the out-of-plane direction of the two-dimensional magnetic system. Since the out-of-plane external field usually increases M_z to reduce the Zeeman energy, it is reasonable to show similar structures as the result of maximizing M_z and W in our methods. Figure 4d shows the distributions of W values calculated from the spin configurations evolved by the genetic algorithm to maximize W and M_z for each. We find that the W values from both optimizations are increased compared with those from the initial generation.

It is also noticeable that the optimization is performed within the bounds of the system, producing feasible and realizable results as commonly observed in experimental research[24,31]. Since W counts the number of skyrmion structures in a spin configuration, breaking the spin structures into smaller structures is one way to maximize W, yet this is not compatible with the Hamiltonian of the system as well as not feasible in experimental observations. M_z can also be maximized by generating a single out-of-plane domain along the z direction, but this is also incompatible with the Hamiltonian. Although these are very intuitive methods for maximizing W and M_z, our method takes more suitable approaches to the Hamiltonian of the system during the optimization processes, as shown in Fig. 4b–d. This shows that our method is a physically plausible optimization method working within the constraints of a system.

By maximizing C, we obtain a well-aligned stripe pattern as shown in Fig. 4e; the stripes are aligned along the direction of a. A well-aligned stripe pattern is also generated by minimizing ε, as shown in Fig. 4f. Similar to the skyrmion cases discussed above, it is also noticeable that the results of both optimization processes, maximizing C and minimizing ε, are the well-aligned stripe patterns because there are no such spin configurations in our
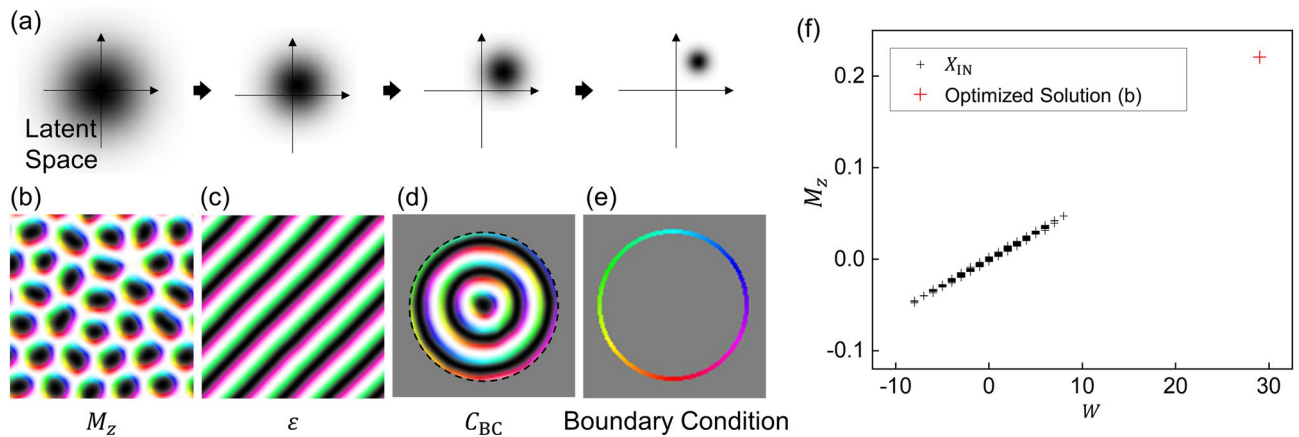
Figure 5. The results of the stochastic algorithm. (**a**) A schematic illustration of how the algorithm progresses. The blurs indicate the Gaussian distributions constructed with $\boldsymbol{\mu}^*$ (mean) and $\boldsymbol{\sigma}^*$ (standard deviation) in this algorithm. (**b–d**) Spin configuration obtained by (**b**) maximizing M_z, (**c**) minimizing ε, and (**d**) maximizing C_{BC}. The boundary condition is given along the dashed line. (**e**) Illustration of the boundary condition $\boldsymbol{S}_{BC}$. (**f**) W and M_z of 1000 X_{IN}s and the optimized solution shown in (**b**).

training dataset. The stripe pattern is the ground state of this system[19]. Nevertheless, generating the ground state through an SA process is significantly difficult due to the existence of countless labyrinth patterns that are the metastable states of the system. The ε of both results, obtained by maximizing C and minimizing ε, decrease, as shown in Fig. 4g. Considering our method can search out the well-aligned stripe patterns that are not given in the training process, we believe that our method can be utilized as a novel approach to finding uncharted ground states of various systems. As discussed in "Training VAE and generating new data" section, energy is a proper indicator to evaluate the feasibility of spin configurations. Therefore, it can be interpreted that we ranked the offspring based on the feasibility while optimizing the energy with the genetic algorithm.

Our results show that VAEs are advantageous for applying genetic algorithms to optimization problems. Results of the algorithm are optimal solutions following the general rules of the system. Various metrics, such as discriminator networks of GANs or pre-trained classifiers, can be adapted for those purposes.

Stochastic algorithm. In this algorithm, there is a Gaussian distribution from which the latent codes, $\boldsymbol{z}^*$, are sampled, and the mean, $\boldsymbol{\mu}^*$, and standard deviation, $\boldsymbol{\sigma}^*$, of the distribution are updated by the Adam optimizer to optimize an objective quantity. For each optimization iteration, we sampled 100 $\boldsymbol{z}^*$s from the distribution and decoded them to calculate an objective quantity similar to that of the genetic algorithm. As the algorithm progresses, the distribution gradually converges to a location in the latent space of the trained VAE, as schematically shown in Fig. 5a. We repeat this algorithm until the distribution converges sufficiently to a location and found that the latent codes sampled around the location can be decoded as the optimal solutions to given optimization problems.

Consequently, the skyrmion structures and perfectly aligned stripe pattern are obtained using this algorithm for the M_z maximization and ε minimization objectives, respectively, as shown in Fig. 5b,c. Compared with the results from the genetic algorithm, we find that the spin configurations obtained using this stochastic algorithm are more realistic in the visual inspection; in Fig. 5b,c, there are no nodal points and wiggles which are shown in the images for the last generations in Fig. 4b,c,e,f. Quantitatively, the maximum energy density of Fig. 5b is approximately -1.88, which is much lower than the energy peak (~ -0.11) in Fig. 1c. Note that we do not give any bias for reality or energetic stability in the optimization process.

Since VAEs model the given data, it is known that interpolation using VAEs works successfully providing reasonable results[33]. Our optimization results can also be understood as combinations of training data. For example, the skyrmion structures (see Figs. 4b,c, 5b) might be considered combinations of many training data that occasionally contain a few skyrmions. Nevertheless, as discussed in "Genetic algorithm" section, the skyrmion structure is outside the given data, not only in a visual inspection but also in physical quantities. Figure 5f shows the M_z and W of spin configurations. We found that the relation between M_z and W of X_{IN} is maintained in our optimized solution (b), even though it is an extreme situation. We confirm that it is possible to obtain desired optimal solutions using the stochastic algorithm, keeping the plausibility with the help of the VAE trained on samples.

We investigate whether this algorithm can be applied not only to a system with periodic boundary conditions but also to a geometrically confined system with a specific boundary shape. For this purpose, the objective quantity is set to be the correlation between a decoded spin configuration and the specific boundary condition map shown in Fig. 5d, $C_{BC} = \langle \boldsymbol{S}_{BC}(\boldsymbol{r}) \cdot \boldsymbol{S}(\boldsymbol{r}) \rangle$, where $\boldsymbol{S}_{BC}$ indicates the boundary condition. The boundary condition is illustrated in Fig. 5e. We perform an optimization process using this stochastic algorithm to maximize C_{BC}. Since the spins in the boundary condition map are located only at the round narrow strip, maximizing C_{BC} presents geometrical confinement on the decoded spin configurations as shown in Fig. 5d. We confirmed that the spin configuration is in an energetically stable state, which means that this stochastic algorithm is applicable to find physically plausible states in various systems regardless of boundary conditions.
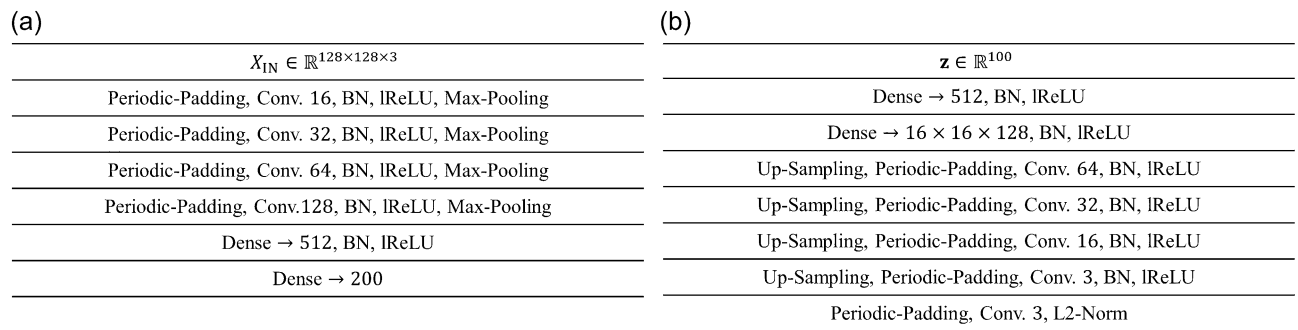
(a)

$X_{\text{IN}} \in \mathbb{R}^{128 \times 128 \times 3}$
Periodic-Padding, Conv. 16, BN, lReLU, Max-Pooling
Periodic-Padding, Conv. 32, BN, lReLU, Max-Pooling
Periodic-Padding, Conv. 64, BN, lReLU, Max-Pooling
Periodic-Padding, Conv.128, BN, lReLU, Max-Pooling
Dense → 512, BN, lReLU
Dense → 200

(b)

$\mathbf{z} \in \mathbb{R}^{100}$
Dense → 512, BN, lReLU
Dense → 16 × 16 × 128, BN, lReLU
Up-Sampling, Periodic-Padding, Conv. 64, BN, lReLU
Up-Sampling, Periodic-Padding, Conv. 32, BN, lReLU
Up-Sampling, Periodic-Padding, Conv. 16, BN, lReLU
Up-Sampling, Periodic-Padding, Conv. 3, BN, lReLU
Periodic-Padding, Conv. 3, L2-Norm

Figure 6. Detailed architectures of (**a**) the encoder and (**b**) decoder. The operations in a single block are written via its procedure. BN and lReLU represent batch normalization and leaky-ReLU activation layers, respectively.

The idea of exploring the latent space of a VAE for generating optimized samples of a distribution given cost functions is innovative and promising as our results demonstrate. Our methods, single-code modification, genetic, and stochastic algorithms, are generally applicable if you can train deep generative models that provide compressed representations of the real world. Given the complexity of the distribution of the problem, the number of instances may increase substantially, which needs to be considered. Also, it must be noted that the success of optimization with trained deep generative models can be applications dependent. For image generation and optimization, the success might be more expected because there has been much research and development. For other systems, it can be challenging to train reliable VAEs and obtain optimal solutions. If the trained deep generative model does not contain the desired optimal solution, to find the solution in the latent space is hardly expected. It is widely known that prescribed models such as VAEs have better data coverage than implicit models such as GANs[34]. We recommend choosing and training the deep generative model considering its performance in terms of data coverage.

Conclusion

In this study, we trained generative model, the variational autoencoder (VAE), to extract the features of a two-dimensional magnetic system. We implemented three algorithms, the single-code modification algorithm, genetic algorithm, and stochastic algorithm, to solve various optimizing problems by exploring the latent space of the trained VAE. The single-code modification algorithm obtains a metastable spin configuration by locally optimizing the energy minimization problem. The genetic algorithm and stochastic algorithm are applied to various global optimization problems, such as minimizing energy and maximizing topological numbers. We confirmed that, even though the optimal solutions are not given explicitly in the VAE model training process, the optimization algorithms successfully determine the optimal solutions in the latent space regardless of the boundary conditions of the system. We suggest that our method based on latent space exploration can be applied to various other systems to optimize their properties.

Methods

The network structure of VAE. The encoder and decoder of the VAE consist of fully connected neural networks (FCNNs) and convolutional neural networks (CNNs). Figure 6 shows the details of our architectures. The encoder is composed of four CNN layers with 16, 32, 64, and 128 filters and two FCNN layers with 512 and 200 neurons. After all CNN layers of the encoder, max-pooling layers with a 2 × 2 pooling size are adopted. The decoder is composed of two FCNN layers with 512 and 8,192 neurons, the reshape layer, CNN layers with 64, 32, 16, 3, 3 kernels, and the L2-normalization layer. Upsampling layers with 2 × 2 upsampling sizes are applied before all CNN layers of the decoder, except the last CNN layer. Batch normalization layers and leaky-ReLU activation layers are applied immediately after all FCNN and CNN layers, in both the encoder and the decoder, except for the last layer of each. Since the dataset is generated under the periodic boundary condition, periodic padding is applied immediately before all CNN layers.

Detailed VAE training process. We used 30,000 spin configurations as a training dataset of VAE and used 5000 spin configurations as a valid dataset during training. The VAE is fed a batch of data at a time, where the batch consists of 100 spin configurations. The Adam optimizer with the learning rate 10^{-3} modifies the network parameters to minimize L_{VAE} every time the batch goes through the VAE. The β_1 and β_2 of the Adam optimizer are fixed at 0.9 and 0.999, respectively. The training progresses for 100 epochs.

To select the appropriate coefficients of L_{KL}, β, we trained four VAE models with β values of 0.1, 0.01, 0.001, and 0.0001. VAEs with β larger than 0.001 are excluded because of the posterior collapse problem[35]. We need a sufficiently compressed latent space so that the standard normal distribution represents the whole latent space. Thus, we chose the VAE of β at 0.001, which represents a latent code closer to the standard normal distribution.

Single-code modification algorithm. The Adam optimizer with the learning rate 10^{-2} is used for the single-code modification. Other hyperparameters are identical to the Adam optimizer used for VAE training. The single-code modification algorithm is repeated for 10,000 iterations.

7

Genetic algorithm. The genetic algorithm starts with 100 latent codes sampled from a standard normal distribution. Roulette wheel selection, uniform crossover, and random reset mutation operations are sequentially applied to construct a single new individual. The roulette wheel selection is applied to choose two parents[36]. We give the selection pressure differently depending on the objective quantity, as the selection pressure is a hyperparameter of the genetic algorithm. The selection pressure is given by 8 to optimize W and M_z, and by 5 to optimize C and ε. Uniform crossover is applied, which means that each gene z_i^* is selected randomly from one of the corresponding genes of the parents[37]. Random reset mutation is applied so that each gene has a probability of being replaced by a new number generated from the standard normal distribution[38]. The mutation error, the probability that a gene will be mutated, is chosen by 0.01. The process is repeated for 10,000 iterations.

Stochastic algorithm. An Adam optimizer with the same hyperparameters as used in the single-code modification algorithm is used for the stochastic algorithm. γ starts at 1 and is gradually decreased to a tenth after every 10,000 iterations. The process continues until σ^* approaches zero so that the sampled latent codes are all the same.

Data availability

The dataset used in this work is available at https://data.mendeley.com/datasets/4833bhfjv3/1.

References

1. Fan, L. & Ziegler, T. Optimization of molecular structures by self-consistent and nonlocal density-functional theory. *J. Chem. Phys.* **95**, 7401 (1991).
2. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, 864 (1964).
3. Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **140**, 1133 (1965).
4. Pincus, M. Letter to the editor—A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Oper. Res.* **18**, 1225 (1970).
5. Kwon, H. Y. & Won, C. Effects of Dzyaloshinskii-Moriya interaction on magnetic stripe domains. *J. Magn. Magn. Mater.* **351**, 15 (2014).
6. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671 (1983).
7. Venkatasubramanian, V., Chan, K. & Caruthers, J. M. Evolutionary design of molecules with desired properties using the genetic algorithm. *J. Chem. Inf. Comput. Sci.* **35**, 188 (1995).
8. Deaven, D. M. & Ho, K. M. Molecular geometry optimization with a genetic algorithm. *Phys. Rev. Lett.* **75**, 288–291 (1995).
9. Kwon, H. Y., Kim, N. J., Lee, C. K. & Won, C. Searching magnetic states using an unsupervised machine learning algorithm with the Heisenberg model. *Phys. Rev. B.* https://doi.org/10.1103/PhysRevB.99.024423 (2019).
10. Kwon, H. Y. *et al.* Magnetic state generation using hamiltonian guided variational autoencoder with spin structure stabilization. *Adv. Sci.* **8**, 2004795 (2021).
11. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
12. Goodfellow, I. J. *et al.* Generative adversarial nets. In *Advances in Neural Information Processing Systems* Vol. 3 (eds Goodfellow, I. J. *et al.*) (Red Hook, 2014).
13. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014—Conference Track Proceedings* (2014).
14. Ruthotto, L. & Haber, E. An introduction to deep generative modeling. *GAMM-Mitteilungen* **44**, e202100008 (2021).
15. Fernandes, P., Correia, J. & Machado, P. Evolutionary latent space exploration of generative adversarial networks. In *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* Vol. 12104 (LNCS, 2020).
16. Gao, Z. *et al.* Large-dimensional seismic inversion using global optimization with autoencoder-based model dimensionality reduction. *IEEE Trans. Geosci. Remote Sens.* **59**, 1718 (2021).
17. Saha, S. *et al.* Quantifying the generative capabilities of variational autoencoders for 3D car point clouds. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020.* https://doi.org/10.1109/SSCI47803.2020.9308513 (2020).
18. Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Sci.* **4**, 268 (2018).
19. Kwon, H. Y., Bu, K. M., Wu, Y. Z. & Won, C. Effect of anisotropy and dipole interaction on long-range order magnetic structures generated by Dzyaloshinskii-Moriya interaction. *J. Magn. Magn. Mater.* **324**, 2171 (2012).
20. Dzyaloshinsky, I. A thermodynamic theory of "weak" ferromagnetism of antiferromagnetics. *J. Phys. Chem. Solids* **4**, 241–255 (1958).
21. Moriya, T. New mechanism of anisotropic superexchange interaction. *Phys. Rev. Lett.* **4**, 228 (1960).
22. Bogdanov, A. N. & Rößler, U. B. Chiral symmetry breaking in magnetic thin films and multilayers. *Phys. Rev. Lett.* https://doi.org/10.1103/PhysRevLett.87.037203 (2001).
23. Romming, N., Kubetzka, A., Hanneken, C., von Bergmann, K. & Wiesendanger, R. Field-dependent size and shape of single magnetic Skyrmions. *Phys. Rev. Lett.* https://doi.org/10.1103/PhysRevLett.114.177203 (2015).
24. Yu, X. Z. *et al.* Real-space observation of a two-dimensional skyrmion crystal. *Nature* **465**, 901 (2010).
25. Kwon, H. Y., Kang, S. P., Wu, Y. Z. & Won, C. Magnetic vortex generated by Dzyaloshinskii-Moriya interaction. *J. Appl. Phys.* **113**, 133911 (2013).
26. Miyatake, Y., Yamamoto, M., Kim, J. J., Toyonaga, M. & Nagai, O. On the implementation of the 'heat bath' algorithms for monte carlo simulations of classical heisenberg spin systems. *J. Phys. C Solid State Phys.* **19**, 2539 (1986).
27. Higgins, I. *et al.* B-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings* (2017).
28. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings* (2015).
29. Leonov, A. O., Bogdanov, A. N., Hirschberger, M., Hayami, S. & Tokura, Y. Nanometric skyrmion lattice from anisotropic exchange interactions in a centrosymmetric host. *New J. Phys.* **23**, 23039 (2021).
30. Bogdanov, A. & Hubert, A. Thermodynamically stable magnetic vortex states in magnetic crystals. *J. Magn. Magn. Mater.* **138**, 255–269 (1994).

31. Takagi, R. *et al.* Square and rhombic lattices of magnetic skyrmions in a centrosymmetric binary compound. *Nat. Commun.* https://doi.org/10.1038/s41467-022-29131-9 (2022).
32. Nagaosa, N. & Tokura, Y. Topological properties and dynamics of magnetic skyrmions. *Nat. Nanotechnol.* **8**, 899 (2013).
33. Yoon, H. G. *et al.* Interpolation and extrapolation between the magnetic chiral states using autoencoder. *Comput. Phys. Commun.* **272**, 108244 (2022).
34. Sami, M. & Mobin, I. *A Comparative Study on Variational Autoencoders and Generative Adversarial Networks: A Comparative Study on Variational Autoencoders and Generative Adversarial Networks* (2019).
35. Bowman, S. R. *et al. Generating Sentences from a Continuous Space.*
36. David, E. G. Genetic algorithms in search, optimization, and machine learning. *Choice Rev. Online* **27**, 0936 (1989).
37. Syswerda, G. Uniform crossover in genetic algorithms. In *Proc. the 3rd Intern. Conference on Genetic Algorithms* (1991).
38. de Falco, I., della Cioppa, A. & Tarantino, E. Mutation-based genetic algorithm: Performance evaluation. *Appl. Soft Comput.* **1**, 285 (2002).

Author contributions
S.M.P. devised algorithms, trained the VAE, and experimented with the algorithms. H.Y.K., D.B.L., and H.G.Y. generated the datasets and participated in the training process. J.W.C., H.Y.K., and C.W. contributed to the discussions of the main results. H.Y.K. and C.W. equally supervised the work progresses. All authors contributed to the final manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to H.Y.K. or C.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.